Socially-aware Management of 
New Overlay Application Traffic with 
Energy Efficiency in the Internet

European Seventh Framework Project FP7-2012-ICT- 317846-STREP

Deliverable D2.4 
Report on Final Specifications 
of Traffic ManagementMechanisms 
and Evaluation Results

The SmartenIT Consortium

Universität Zürich, UZH, Switzerland 
Athens University of Economics and Business - Research Center, AUEB, Greece 
Julius-Maximilians Universität Würzburg, UniWue, Germany 
Technische Universität Darmstadt, TUD, Germany 
Akademia Gorniczo-Hutnicza im. Stanisława Staszica w Krakowie, AGH, Poland 
Intracom SA Telecom Solutions, ICOM, Greece 
Alcatel Lucent Bell Labs, ALBLF, France 
Instytut Chemii Bioorganicznej PAN, PSNC, Poland 
Interoute S.P.A, IRT, Italy 
Telekom Deutschland GmbH, TDG, Germany

© Copyright 2014, the Members of the SmartenIT Consortium

For more information on this document or the SmartenIT project, please contact:

Prof. Dr. Burkhard Stiller 
Universität Zürich, CSG@IFI 
Binzmühlestrasse 14 
CH—8050 Zürich 
Switzerland

Phone: +41 44 635 4331 
Fax: +41 44 635 6809 
E-mail: info@smartenit.eu
Title: Report on Final Specifications of Traffic Management Mechanisms and Evaluation Results

Type: Public

Editor(s): Ioanna Papafili, George D. Stamoulis

E-mail: iopapafi@aueb.gr

Author(s): Valentin Burger, George Darzanos, Manos Dramitinos, Zbigniew Dulinski, Fabian Kaup, Ioanna Papafili, Patrick Poullie, Sabine Randriamasy, Grzegorz Rzym, Michael Seufert, George D. Stamoulis, Rafal Stankiewicz, Krzysztof Wajda, Matthias Wichtlhuber, Piotr Wydrych

Doc ID: D2.4-v1.7

AMENDMENT HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Description/Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>November 1, 2012</td>
<td>Burkhard Stiller</td>
<td>First version</td>
</tr>
<tr>
<td>V0.2</td>
<td>February 28, 2014</td>
<td>Ioanna Papafili</td>
<td>Update of deliverable template; initial TOC draft</td>
</tr>
<tr>
<td>V0.3</td>
<td>March 5, 2014</td>
<td>Ioanna Papafili</td>
<td>ToC update</td>
</tr>
<tr>
<td>V0.4</td>
<td>March 7, 2014</td>
<td>Ioanna Papafili</td>
<td>Responsibilities assignment</td>
</tr>
<tr>
<td>V0.5</td>
<td>May 12, 2014</td>
<td>Fabian Kaup</td>
<td>Input in theoretical models</td>
</tr>
<tr>
<td>V0.6</td>
<td>June 15, 2014</td>
<td>Valentin Burger, George Darzanos, Manos Dramitinos, Fabian Kaup, Patrick Poullie</td>
<td>Input in theoretical models and specification of individual mechanisms</td>
</tr>
<tr>
<td>V0.7</td>
<td>June 16, 2014</td>
<td>Ioanna Papafili, Matthias Wichtlhuber, Patrick Poullie, Michael Seufert, Manos Dramitinos, Fabian Kaup, Zbigniew Dulinski, Valentin Burger</td>
<td>Editing; comments on presentation style and content. Input in Chapter 3: Theoretical models, Chapter 4: OFS mechanisms, Chapter 5: EFS mechanisms – Specifications of individual and synergetic solutions and evaluation results – Detailed specification of individual mechanisms in appendices.</td>
</tr>
<tr>
<td>V0.8</td>
<td>August 6, 2014</td>
<td>Ioanna Papafili</td>
<td>Integration of provided material, editing, and comments.</td>
</tr>
<tr>
<td>V0.8.1</td>
<td>August 29, 2014</td>
<td>Valentin Burger, Manos Dramitinos, Fabian Kaup, Ioanna Papafili, Matthias Wichtlhuber</td>
<td>Input in Ch. 1, 2, 3, 4 and 5</td>
</tr>
<tr>
<td>V0.9</td>
<td>September 2, 2014</td>
<td>Ioanna Papafili</td>
<td>Integration of provided material, editing, and comments.</td>
</tr>
<tr>
<td>V0.9.1</td>
<td>September 5, 2014</td>
<td>Ioanna Papafili, Sabine Randriamasy</td>
<td>Revisions and updates</td>
</tr>
<tr>
<td>V1.0</td>
<td>September 12, 2014</td>
<td>Ioanna Papafili, George D. Stamoulis</td>
<td>Integration of provided material, editing, and comments.</td>
</tr>
<tr>
<td>V1.1</td>
<td>September 19, 2014</td>
<td>Valentin Burger, George Darzanos, Manos Dramitinos, Ioanna Papafili, George D. Stamoulis</td>
<td>Integration of provided material, editing, and comments.</td>
</tr>
<tr>
<td>v.1.2</td>
<td>September 22, 2014</td>
<td>Sabine Randriamasy, Patrick Poullie, Manos Dramitinos, Ioanna Papafili</td>
<td>Integration of provided material, editing, and comments. Prevention of version to be sent to the advisory board.</td>
</tr>
<tr>
<td>V1.3</td>
<td>October 7, 2014</td>
<td>Ioanna Papafili</td>
<td>Integrate comments from ABM; editing.</td>
</tr>
<tr>
<td>V1.4.1</td>
<td>October 16, 2014</td>
<td>Roman Lapacz, Sergios Soursos and George D. Stamoulis</td>
<td>Internal review of D2.4; provide feedback to editor and authors.</td>
</tr>
<tr>
<td>V1.5</td>
<td>October 23, 2014</td>
<td>Michael Seufert, Fabian Kaup, Manos Dramitinos, George Darzanos, Rafal</td>
<td>Address comments by internal reviewers.</td>
</tr>
</tbody>
</table>
Table of Contents

Amendment History..................................................................................................................2
Table of Contents......................................................................................................................4
List of Figures............................................................................................................................8
List of Tables.............................................................................................................................12
1 Executive Summary.................................................................................................................13
2 Introduction.............................................................................................................................16
  2.1 Summary of past work and TM selection process by SmartenIT.................................16
  2.2 Purpose of this Document..............................................................................................20
  2.3 Document Outline............................................................................................................21
3 Business modelling.................................................................................................................22
  3.1 Cloud Business Models Context ..................................................................................23
  3.2 Cloud Computing, Brokers and ISPs’ role....................................................................24
  3.3 Example Business Model Cases .....................................................................................30
    3.3.1 CloudItalia ...............................................................................................................30
    3.3.2 Huawei and China Mobile .......................................................................................30
    3.3.3 TM Forum ...............................................................................................................31
  3.4 ISP Managed Services ......................................................................................................34
    3.4.1 The Virtual Private Cloud Business Case .................................................................34
    3.4.2 The Federated Cloud business case ........................................................................35
    3.4.3 The Open Market Business Model .........................................................................36
  3.5 Cloud Business Models and SmartenIT Interplay ...........................................................37
4 Theoretical investigations.......................................................................................................41
  4.1 Investigation of incentives for cloud federation .............................................................41
    4.1.1 Cloud Providers as M/M/1 system ..........................................................................42
    4.1.2 Federation of Cloud providers ...............................................................................43
    4.1.3 Stakeholders and Pricing .......................................................................................47
  4.2 Investigation of content demand in OSNs ......................................................................49
    4.2.1 Content demand model with temporal dynamics ....................................................49
    4.2.2 Model for prediction of individual content requests based on social network traces 50
  4.3 Investigation of topological characteristics of the Internet..............................................53
    4.3.1 Model for global distribution of home routers on ASes .........................................53
    4.3.2 Model for geographic distribution of WiFi hotspots .............................................56
  4.4 Quantification of power consumption ...........................................................................59
    4.4.1 Power model of the Raspberry Pi .........................................................................59
    4.4.2 General Power Model of Wireless Data Transmissions ........................................64
  4.5 Investigation of VM migration ..........................................................................................66
  4.6 Virtual Node Model .........................................................................................................69
    4.6.1 Distributed Accounting Model .............................................................................70
    4.6.2 Communication Model within Virtual Nodes............................................................70
    4.6.3 Scenario Model .......................................................................................................71
5 Specification of Mechanisms for OFS................................................................................77
  5.1 Dynamic Traffic Management (DTM) ............................................................................77
    5.1.1 Intelligence .............................................................................................................78
    5.1.2 Parameters & metrics .............................................................................................82
5.1.3 Evaluation framework ........................................ 83
5.1.4 Evaluation results ........................................... 83
5.2 Inter-Cloud Communication (ICC) ............................ 85
5.2.1 Intelligence .................................................. 86
5.2.2 Parameters & metrics ....................................... 89
5.2.3 Evaluation framework ..................................... 90
5.2.4 Evaluation results .......................................... 91
5.3 Multi-Resource Allocation (MRA) ............................ 95
5.3.1 Intelligence .................................................. 95
5.3.2 Parameters & metrics ....................................... 99
5.4 DTM++ .......................................................... 100
5.4.1 Intelligence .................................................. 100
5.4.2 Evaluation framework and preliminary results ........ 103
6 Specification of Mechanisms for EFS ........................................ 106
6.1 RB-HORST ...................................................... 106
6.1.1 Intelligence .................................................. 108
6.1.2 Parameters & metrics ....................................... 109
6.1.3 Evaluation framework ..................................... 111
6.1.4 Evaluation results .......................................... 114
6.2 Socially-aware mechanism for Efficient Content Delivery (SEConD) ........................................ 120
6.2.1 Intelligence .................................................. 121
6.2.2 Parameters & metrics ....................................... 123
6.2.3 Evaluation framework ..................................... 125
6.2.4 Evaluation results .......................................... 126
6.3 The vINCENT extension for RB-HORST ...................... 128
6.3.1 Intelligence .................................................. 130
6.3.2 Parameters & Metrics ...................................... 131
6.3.3 Evaluation framework ..................................... 132
6.4 Mobile Network Assistant (MONA) .......................... 134
6.4.1 Intelligence .................................................. 134
6.4.2 Parameters & metrics ....................................... 135
6.4.3 Evaluation framework ..................................... 136
6.4.4 Evaluation results .......................................... 136
6.5 Multi-Criteria Application endPoint Selection (MUCAPS) ........................................ 137
6.5.1 Intelligence .................................................. 139
6.5.2 Parameters & metrics ....................................... 140
6.5.3 Evaluation framework ..................................... 142
6.5.4 Evaluation results .......................................... 144
6.6 RB-HORST++ .................................................... 146
6.6.1 Intelligence .................................................. 146
6.6.2 Parameters and metrics ................................... 148
7 Coverage of the SmartenIT objectives .............................. 150
7.1 Coverage by individual mechanisms and synergies .......... 150
8 Summary and Conclusions .......................................... 154
8.1 Key Outcomes and Lessons Learnt ............................ 154
8.2 Next Steps ...................................................... 159
9 Smart Objectives ................................................... 160
10 References ......................................................... 165
11 Abbreviations ....................................................... 172
12 Acknowledgements ................................................ 172

© Copyright 2014, the Members of the SmartenIT Consortium
13 Appendixes .................................................................................................................. 173

13.1 Detailed specification of DTM .................................................................................. 173
13.1.1 Inter cloud communication - multi cloud transfer to single cloud .......................... 173
13.1.2 Inter cloud communication - single cloud transfer to multi cloud ......................... 175
13.1.3 Traffic distribution procedures ............................................................................. 175
13.1.4 Physical link topology limitation and routing policy limitation .............................. 176
13.1.5 Innovation of DTM mechanism for inter-cloud communication ............................ 179
13.1.6 Allocation of the DTM functionalities to the SmartenIT components .................... 180

13.2 Simplified DTM for inter-cloud communication ....................................................... 186
13.2.1 Initial manual configuration in the ISP-A domain ................................................. 188
13.2.2 Initial communication between components in the ISP-A domain ......................... 189
13.2.3 Initial manual configuration in the ISP-B domain ................................................ 190
13.2.4 Initial communication between components in the ISP-B domain ........................ 190
13.2.5 Component communication .................................................................................. 190
13.2.6 Reference vector calculation algorithm for DTM Economic Analyzer subcomponent in the S-box-A: ................................................................. 195
13.2.7 Compensation vector calculation algorithm for DTM subcomponent in the S-box-A: 198
13.2.8 Traffic distribution algorithm for the Traffic Distributor in the S-box-B (in SDN controller or DTM subcomponent) ............................................................... 198
13.2.9 Test plan ............................................................................................................. 199

13.3 Detailed specification of ICC ..................................................................................... 202
13.3.1 Terminology – ICC layers and modes of operation ............................................... 202
13.3.2 Pseudocode – ICC Scheduling Time-shiftable Traffic (STraS) .............................. 206
13.3.3 Information – DB schema ..................................................................................... 208
13.3.4 API and Interfaces ............................................................................................... 209
13.3.5 Discussion .......................................................................................................... 210

13.4 Detailed specification of MRA .................................................................................. 211
13.4.1 Terminology ........................................................................................................ 212
13.4.2 Utilization Information ........................................................................................ 212
13.4.3 Pseudo code for the consumption collector ......................................................... 213
13.4.4 Internal methods ................................................................................................. 213

13.5 Detailed specification of RB-HORST ........................................................................ 215
13.5.1 Functionalities of RBH ........................................................................................ 216
13.5.2 SMT components involved in RBH ................................................................. 216
13.5.3 Component-based sequence diagram for each functionality ............................... 218

13.6 Detailed specification of SEConD ............................................................................. 226
13.6.1 Socially-aware messaging overlays (messaging clusters) ..................................... 226
13.6.2 Socially-aware Proxy Server (SPS) ...................................................................... 227
13.6.3 Local Content-based P2P overlays ....................................................................... 227
13.6.4 Caching .............................................................................................................. 227
13.6.5 Socially-aware pull based prefetching algorithm ................................................. 228
13.6.6 Stored information in different components .............................................. 228
13.6.7 Methods ............................................................................................................ 229
13.6.8 Communication ................................................................................................. 233

13.7 Detailed specification of vINCENT ............................................................................. 236
13.7.1 PING .................................................................................................................. 236
13.7.2 TUNNEL SETUP ............................................................................................... 236
13.7.3 TRUSTED SET MANAGEMENT ....................................................................... 237
13.7.4 Invoked Methods ........................................................................................................... 238
13.8 Detailed specification of MONA ...................................................................................... 244
13.8.1 Mechanism Specification ............................................................................................ 244
13.8.2 External Interfaces ........................................................................................................ 250
13.9 Detailed specification of MUCAPS .................................................................................. 253
13.9.1 Details on the MUCAPS related functional blocks ...................................................... 254
13.9.2 Network-based MUCAPS implementation and execution .......................................... 257
List of Figures

Figure 2-1: Prioritization of UCs to be considered. .................................................. 18
Figure 2-2: Prioritization of TM mechanisms to be considered. .............................. 19
Figure 3-1: Taxonomy of cloud business model according to Forrester Research Inc. ... 23
Figure 3-2: The spectrum of Cloud computing services. (Source: Gene Phifer, Managing
Vice President at Gartner) ................................................................................................ 25
Figure 3-3: The Cloud systems taxonomy and features [20]. ........................................ 26
Figure 3-4: Appirio focus on brokering and building differentiated services. .............. 27
Figure 3-5: Cloud brokers’ services and importance [32]. ............................................ 28
Figure 3-6: From low-value commodity services to high-value business solutions [29]. ... 29
Figure 3-7: Cloud business and technical evolution [33]. ............................................ 29
Figure 3-8: The Amazon repeatable business model [36]. ........................................... 32
Figure 3-9: The TMforum transformation of the Amazon BM for cloud services [36]. .... 33
Figure 3-10: The TM Forum Ecosystem Enablement Platform and sample service
instantiations on top that ISPs could provision – but also OTTs [36]. ......................... 33
Figure 3-11: The Arzuna Agility high-level approach [39]. ........................................... 35
Figure 3-12: The Arzuna Agility approach materialized via SLAs [39]. ......................... 36
Figure 3-13: The Deutsche Borse Cloud Exchange [44]. ........................................... 37
Figure 3-14: Mapping of SmartenIT mechanisms to business models. ......................... 39
Figure 3-15: Mapping of SmartenIT theoretical investigations (presented in Chapter 4) to
the business models that motivate them. ................................................................. 40
Figure 4-1: A single Cloud Provider as an M/M/1 queueing system .......................... 42
Figure 4-2: The two Cloud Providers operating separately. ...................................... 43
Figure 4-3: One way federation of two cloud providers ........................................... 44
Figure 4-4: Values of $\alpha^*$ in $d$, for $\lambda_1 = 2\lambda_2$ and $\mu_1 = \mu_2$ ....... 45
Figure 4-5: Full federation of two cloud providers .................................................... 46
Figure 4-6: Values of $\alpha$ and $\beta$ that minimize the average delay of total system, for $d = 0$, $\lambda_1 = 2$ $\lambda_2$ and $\mu_1 = \mu_2$ ................................................................. 47
Figure 4-7: Global optimal (blue line) vs. $P_1$ optimal (red line) ($\alpha^*$ in $d$, $\lambda_1 = 2\lambda_2$ and $\mu_1 = \mu_2$) ............................................................ 48
Figure 4-8: Function for the dimension age. ............................................................. 51
Figure 4-9: 16bit AS address room. ............................................................................ 54
Figure 4-10: 32bit AS address room. ......................................................................... 54
Figure 4-11: Zipf-law of the number of active IP-addresses in autonomous systems .... 55
Figure 4-12: Probability distribution of the number of active IP-addresses per AS .... 56
Figure 4-13: Hotspots and Intersections of London. ............................................... 56
Figure 4-14: Distribution of Hotspots with Respect to the Distance to the City Center .... 57
Figure 4-15: London Angular Distribution. ............................................................. 58
Figure 4-16: Measurement setup and its wiring for the Raspberry Pi. ....................... 60
Figure 4-17: Power Consumption vs. CPU utilization. ............................................ 61
Figure 4-18: Ethernet power consumption during download (left) and upload (right) vs.
used bandwidth ......................................................................................................... 62
Figure 4-19: WiFi power consumption during download (left) and upload (right) vs. used
bandwidth ................................................................................................................ 62
Figure 4-20: Virtual node concept. .......................................................................... 69
Figure 4-21: Different scheduling strategies. Lower numbers indicate pieces with higher
priority. The base layer is at the bottom. ................................................................. 71
Figure 4-22: Workload [87]. .................................................................................. 73
Figure 4-23: Main evaluation results (5 experiments, 95 % confidence intervals). .... 74
Figure 4-24: Evaluation of token trading cost..............................................................76
Figure 5-1: Example ISP and cloud/DC link topology. .............................................79
Figure 5-2: Network prefixes announced by BGP for DTM via selected inter-domain links. BGP routing policy selectively puts prefixes in the routing tables for the DTM purposes ..................................................................................................................80
Figure 5-3: Tunnels used by DTM. End-points of each tunnel are addressed with different network prefix this results in different inter-domain link traversing by each tunnel. 80
Figure 5-4: The network resources used by the DTM mechanism..................................81
Figure 5-5: Distribution of observed 95th percentile sample pairs, measured during the billing period; values of achieved traffic vector, reference vector, and related inter-domain traffic costs for the experiment without DTM (red) and with DTM (blue): (a) success rate 40%; (b) success rate 30%. ..........................................................84
Figure 5-6: Sorted 5-minute samples with (blue) and without (red) DTM compensation and corresponding 95th percentile thresholds, link 1 and link 2, success rate 40%......85
Figure 5-7: ICC system architecture.............................................................................87
Figure 5-8: The Best Effort traffic pattern and 95th percentile.......................................92
Figure 5-9: The ICC_FAP traffic pattern, the 95th percentile and the extra delays incurred.92
Figure 5-10: The ICC_FA traffic pattern, the 95th percentile and the extra delays incurred.93
Figure 5-11: Comparison of traffic patterns attained under Best Effort, ICC_FA, ICC_FAP.93
Figure 5-12: The Best Effort Internet traffic pattern (without ICC)...............................93
Figure 5-13: The ICC_FA traffic pattern, the 95th percentile and the extra delays incurred.93
Figure 5-14: Comparison of traffic patterns without ICC and with ICC_FAP .............94
Figure 5-15: Comparison of traffic patterns attained under ICC_FA and ICC_FAP. ........94
Figure 5-16: Reminder on DTM operation over the inter-DC bulk data transfer scenario.101
Figure 5-17: The rationale of the combined operation of DTM and ICC .....................102
Figure 5-18: Traffic shaping in Link 1 performed by ICC after DTM is run for a trace of 1 week duration.................................................................104
Figure 5-19: Traffic shaping in Link 1 performed by ICC after DTM is run for a trace of 1 week duration.................................................................104
Figure 5-20: Distribution of observed 95th percentile sample pairs, measured during the billing period of 7 days: experiment with DTM (blue) and with DTM++ (green)....105
Figure 6-1: Process-diagram of a WATCH-event .........................................................113
Figure 6-2: City area of Darmstadt with WiFi access point locations (red) and end-user locations (blue).................................................................114
Figure 6-3: Total ISP cache contribution and mean ISP cache hit rate dependent on home router sharing probability for different ISP cache capacities ................................116
Figure 6-4: Total amount of requests served intra-AS dependent on home router sharing probability for different ISP cache capacities ........................................117
Figure 6-5: Total ISP cache contribution and mean ISP cache hit rate dependent on home router sharing probability for different content popularity distributions with Zipf-exponent α ................................................117
Figure 6-6: Total amount of requests served intra-AS dependent on home router sharing probability for different content popularity distributions with Zipf-exponent α ......118
Figure 6-7: ISP cache hit rate for the five autonomous systems with most end-users dependent on home router sharing probability for different ISP cache capacities. 119
Figure 6-8: WiFi offloading potential for three WiFi access point seeding ranges dependent on the WiFi sharing probability ........................................120
Figure 6-9: The messaging overlay constructed for an interest category .....................122
Figure 6-10: An example of the prefetching algorithm – The source node shares a video hosted in a third-party owned server. (Sequence numbers arrows.) ................123
Figure 6-11: Total inter-AS traffic during a day. ................................................................. 127
Figure 6-12: HORST-vINCENT privacy layer and incentive layer. ................................. 129
Figure 6-13: Basic OMNET++ setup. .................................................................................. 132
Figure 6-14: Visualization of analyzed traces. .................................................................. 133
Figure 6-15: Schematic overview of the MoNA mechanism ............................................. 134
Figure 6-16: Power Consumption of the Nexus S splitting a fixed rate data stream to
multiple interfaces (3G, WiFi). ......................................................................................... 137
Figure 6-17: The MACAO Request Service: function, interface and client ...................... 138
Figure 6-18: Deployment of MUCAPS in the ISP network ............................................. 139
Figure 6-19: Steps involved in MUCAPS and details on the AMWS (Application Metrics
and Weights Selection) process. .................................................................................. 140
Figure 6-20: MUCAPS demonstration and functional evaluation setup ....................... 143
Figure 6-21: Chunk-based representation of a video. ..................................................... 148
Figure 7-1: Topological view of architecture with added scenario overlay (based on
component map taken from D3.1[7]) ........................................................................... 151
Figure 13-1: Inter cloud communication - multi cloud transfer to single cloud .......... 173
Figure 13-2: Inter cloud communication - single cloud transfer to multi cloud ............. 175
Figure 13-3: Available Internet path determined by BGP routing policy (example 1) .... 177
Figure 13-4: Available tunnels resulting from routing policies (refers to Figure 13-3) ...... 177
Figure 13-5: Available Internet path determined by BGP routing policy (example 2) ... 178
Figure 13-6: Available tunnels resulting from routing policies (refers to Figure 13-5) .... 178
Figure 13-7: Available Internet path determined by BGP routing policy (example 3) .... 179
Figure 13-8: Available tunnels resulting from routing policies (refers to Figure 13-7) .... 179
Figure 13-9: The network nodes taking part in the DTM communication and data transfer
between partner ISPs ...................................................................................................... 183
Figure 13-10: The simplified DTM for inter cloud communication - network topology,
network nodes and tunnels (system release 1.0). .......................................................... 186
Figure 13-11: Cost functions. ............................................................................................ 187
Figure 13-12: Initial communication between SmartenIT components in ISP-A domain. 189
Figure 13-13: Initial communication between SmartenIT components in ISP-B domain. 190
Figure 13-14: The DTM operation sequence diagram with only NetFlow link monitoring 193
Figure 13-15: The DTM operation sequence diagram with SNMP and NetFlow link
monitoring .................................................................................................................... 194
Figure 13-16: The areas used in the reference vector optimization procedure. Vectors S1
and S2 determine the range in which optimal solution may be found. ....................... 196
Figure 13-17: The environment for the DTM tests including connection points for traffic
generators and receivers. ............................................................................................. 201
Figure 13-18: The operation of the ICC mechanism when datacenters are non-federated 203
Figure 13-19: The ICC mechanism cloud layer when datacenters are federated .......... 205
Figure 13-20: SmartenIT System Architecture as shown in [9]. ..................................... 211
Figure 13-21: Sequence Diagramm of the MRA Mechanism. ......................................... 211
Figure 13-22: SMT components (in orange) involved in RBH. ....................................... 215
Figure 13-23: The Facebook OAuth process. .................................................................... 218
Figure 13-24: Process of creating or joining the overlay network. ................................... 219
Figure 13-25: Maintenance of overlay network executed in a fixed time interval t. ...... 220
Figure 13-26: The tracking process is executed periodically for each content shared. .... 221
Figure 13-27: The Overlay Based prediction is initiated by the CTM and executed in the
OM. .............................................................................................................................. 221
Figure 13-28: Facebook information retrieval. ................................................................. 222
Figure 13-29: Social prediction uses information stored in the Social Monitor.

Figure 13-30: To find the closest UNaDa a traceroute is requested from all providers.

Figure 13-31: Before the app can be used the user has to be authenticated with Facebook and the user ID will be stored.

Figure 13-32: The mobile App has the Facebook ID of the user stored and uses this to authenticate with the foreign UNaDa.

Figure 13-33: The messaging overlay constructed for an interest category.

Figure 13-34: Steps of the SEConD’s prefetching algorithm.

Figure 13-35: The sequence diagram demonstrates the communication between a user and the local SPS during initialization time.

Figure 13-36: Sequence diagram of the prefetching algorithm.

Figure 13-37: PING messages ensure that the FacebookAPP always has an updated view of the uNaDas online, their location, and their social relations.

Figure 13-38: Tunnel setup procedure sequence diagram.

Figure 13-39: Assumptions on regressors and regressands.

Figure 13-40: I(r,t) determines the height of the incentive when deviating from the average that can be expected from the regression model.

Figure 13-41: Mockup of page design.

Figure 13-42: Overall structure of the Mobile Network Assistant (MoNA) and external interfaces.

Figure 13-43: Details of the Network Optimizer.

Figure 13-44: Sequence diagram of the Network Optimizer.

Figure 13-45: Components of the Network Optimizer.

Figure 13-46: Basic Scheduling decision.

Figure 13-47: Execution of scheduled connections.

Figure 13-48: MACAO with MARC embedded in an AEPG located in the ISP network.

Figure 13-49: Details on the functions involved in MUCAPS and operations sequencing.

Figure 13-50: MARC integrated in a network located AEPG, here a DNS server.

Figure 13-51: Interaction between entities involved in the MACAO AEP ranking.
List of Tables

Table 2-1: Selection criteria for TM mechanisms to be implemented. .................................18
Table 2-2: Additional properties (functionalities) for the extension of the DTM and RB-HORST TM mechanisms leading to the formation of synergetic solutions. .................19
Table 4-1: Parameters of content demand model with temporal dynamics ..........................49
Table 4-2: Functions for the considered dimensions. .........................................................51
Table 4-3: Top five autonomous systems with most active IPs .......................................55
Table 4-4: Goodness of Exponential Fits of the Distance Distribution. ............................57
Table 4-5: The power models of the Raspberry Pi. Here, u is the CPU utilization in the range 0 to 1 and r the traffic rate in Mb/s. .........................................................63
Table 4-6: Bandwidth distribution of network model based on [94] (OECD) and [101] (VNI). 71
Table 4-7: SVC bandwidth per layer [87]. ........................................................................73
Table 4-8: Evaluated parameters; optimal settings are underlined. ..................................73
Table 5-1: Parameters and metrics associated to the DTM mechanism .............................83
Table 5-2: Key parameters for ICC. ....................................................................................89
Table 6-1: Caching strategies employed by RB-HORST. ..................................................110
Table 6-2: Demand models employed for the evaluation of RB-HORST. .........................111
Table 6-3: Default parameters of the content delivery simulation framework ....................114
Table 6-4: Proxy server and SPS contribution w.r.t. the AS size. ......................................127
Table 6-5: For each example AEP: utility vector of normalized relative performances and value for 2 utility functions and optimal AEP (= AEP1)..........................144
Table 6-6: Demonstration scenario with 3 cases. ...............................................................145
Table 7-1: Mapping of TM mechanism and single-scenario synergies to the layers and objectives of the SmartenIT field. .................................................................152
Table 9-1: Overall SmartenIT SMART objectives addressed [1] .....................................160
Table 9-2: Specific SmartenIT SMART objectives addressed; excerpt from the set of Tables of [1] with all SMART objectives of the project. ...............................................161
Table 13-1: Integration of the implemented functions of RB-HORST in the SmartenIT architecture components. ..........................................................216
1 Executive Summary

This document is the deliverable D2.4 “Report on Final Specifications of Traffic Management Mechanisms and Evaluation Results” of Work Package 2 “Theory and Modelling” within the ICT SmartenIT Project 317846. According to the Description of Work of the project, the main objectives of Deliverable D2.4 and the related achievements reported (covering both the Operator-focused and the End user-focused scenarios of the project) are as follows:

Objective 1: To provide a deeper understanding of the technical and economic dependencies between the various stakeholders.

A broad investigation of business models of cloud services has been carried out. Moreover, taking into account these business models, the applicability of SmartenIT mechanisms in the European market of cloud services has been studied. It was deduced that the SmartenIT mechanisms constitute attractive solutions a) for the European ISPs, who can employ SmartenIT Operator-focused solutions to cope with the increase of the traffic in their networks, to differentiate their service portfolio with sophisticated bundled cloud-network service offerings on top of either best-effort or premium differentiated network connections, and to deal with the energy efficiency challenges in data centers, and b) for software and hardware vendors, who can incorporate SmartenIT End user-focused capabilities at the end user devices, so as to promote energy efficiency in mobile devices and attain enhanced Quality of Experience.

Moreover, new theoretical studies have been undertaken to investigate aspects motivated by the identified business models. For instance, concerning the potential of federation formation among two Cloud Service Providers, it has been shown that, depending on the values of certain parameters, there can be high potential for optimization of both performance and providers’ aggregate revenues when the providers cooperate. To achieve this though, mutually beneficial agreements between the providers participating in the federation should be established on the way load and revenues are shared.

Additionally, a multitude of technical aspects have been studied theoretically: In particular, a content demand model is developed to generate requests with temporal dynamics that exhibits a Zipf popularity law, while another model is developed for the global distribution of home routers on ASes, which is used to evaluate the performance of WiFi-offloading approaches. Evaluation results obtained by the model showed a high optimization potential, since more than 60% of the connections can be offloaded by RB-HORST / MONA, if only 10% percent of WiFi access points in a city area are shared.

Furthermore, a preliminary model was introduced to study the complex resource interdependencies arising in the case multiple physical resource sharing within a cloud federation. Also, two power models, one for the consumption of specific equipment (such as the Raspberry Pi), one for the consumption of specific activities (such as mobile/WiFi data transmissions) have also been introduced. All these models will facilitate the evaluation of a wide variety of aspects of the traffic management mechanisms; such evaluation are currently in progress.

Objective 2: To provide the final specifications of the incentive-based cloud traffic management mechanisms that have been proposed in deliverable D2.2.

The traffic management mechanisms introduced in deliverable D2.2 have been specified in detail and assessed by means of both theoretical evaluations and simulations. Operator-focused traffic management solutions include: Dynamic Traffic Management (DTM), which
minimizes the inter-domain traffic cost in multi-homed AS by influencing the distribution of the traffic among links, Inter-Cloud Communication (ICC), which attains a reduced 95th percentile transit charge by controlling the rate of delay-tolerant traffic, marked a priori accordingly by the ISP’s business customer (e.g. cloud/datacenter), and shifting its transmission at off-peak intervals, and Multi-Resource Allocation (MRA), which aims to ensure a fair resource allocation among federated Cloud Service Providers.

Evaluation results show that DTM can: 1) find such a traffic distribution amongst links so that the total interconnection cost is lower, 2) dynamically influence the traffic distribution during the billing period by influencing the network path for manageable traffic. On the other hand, the evaluation of ICC reveals its efficiency in reducing the transit link charge and balancing the load and usage of the network, since non-critical traffic is delayed and transmitted when the network is underutilized. Also, critical traffic such as real-time or zero-tolerance cloud traffic under ICC will experience lower congestion and thus improved performance compared to current Best Effort Internet, thus benefiting the user, the Cloud Service Provider and the ISP. The evaluation of MRA is currently in progress.

Concerning the End user-Focused Scenario, the following traffic management solutions are specified: Replicating Balanced Tracker and Home Router Sharing based on Trust (RB-HORST), which provides WiFi access to trusted users to offload their mobile traffic from 3/4G to WiFi, while simultaneously RB-HORST enabled home routers of trusted users identified through a social network are organized in a content-centric overlay network to support content prefetching, Socially-aware Efficient Content Delivery (SEConD), which employs social information, AS-locality awareness, chunk-based P2P content delivery and prefetching, and a centralized node acting as cache and as P2P tracker and proxy to improve the Quality of Experience of video streaming for users of OSNs and reduce inter-AS traffic, Virtual Incentives (vINCENT), which aims to leverage unused wireless resources among users, while it ensures security by employing tunneling among trusted users, Mobile Network Assistant (MONA), which schedules wireless data transmissions to reduce the energy expenses on the air-interfaces, and Multi-Criteria Application Endpoint Selection (MUCAPS), which aims to improve users’ Quality of Experience by performing selection of communication endpoints, by employing basic ALTO functionality but also involving awareness on the underlying network topology.

The evaluation of RB-HORST shows that it has a high potential to save expensive inter-domain traffic and to take load of ISP caches. The amount of requests served locally increases with the home router sharing probability and reaches 60% of all requests if the ISP cache capacity holds 1% of the video catalogue size. SEConD appears also to be very effective in the cases evaluated, since it achieves: a) a high (~87%) reduction of inter-AS traffic in the potentially expensive transit links, both in total, and peak hours, b) high prefetching accuracy (~88%), and c) minimization of the contribution of the origin server (~13%) of total traffic, in the content delivery due to the P2P dissemination. Investigation of the impact of SEConD in multiple ASes of different size (i.e. number of users) reveals that the larger the AS, the lower the contribution of the centralized node (cache) to achieve similar performance. This provides evidence on the scalability of the mechanism. The evaluation of vINCENT, MONA and MUCAPS is in progress.

Following the detailed specification of individual traffic management mechanisms, synergetic solutions employing characteristics of multiple such mechanisms (yet addressing the same scenario) are investigated, namely: DTM++ employing features of DTM and ICC to perform scheduling of data flows in space (transit links) and time (5-min intervals) in order both to further improve the 95-th percentile inter-connection charge
(compared to the individual mechanisms) and to perform improved load balancing, and RBH++ employing features of RB-HORST, SEConD, vINCENT and MONA, in order to perform content prefetching and mobile to WiFi offloading in an energy efficient manner, while ultimately further improving end-users’ Quality of Experience. The evaluation of both synergetic solutions DTM++ and RBH++ is currently in progress.

Based on the outcome of the aforementioned investigations, we also assess the coverage of the SmartenIT playfield by the developed traffic management mechanisms. The main related conclusion is that there is high potential for a broader coverage of the SmartenIT objectives by integrating parts of developed solutions. This study will be concluded in D2.5 “Report on Definitions of Use-Cases and Parameters”.
2 Introduction

SmartenIT [1] targets an incentive-compatible cross-layer network management scheme for network and cloud operators, cloud service providers, and end-users. The project aims to address accordingly load and traffic patterns of inter-cloud communication and of special applications requiring Quality-of-Experience (QoE)-awareness. To accomplish this, one of the key approaches of SmartenIT is the exploitation of social awareness, i.e. in terms of users' social relationships and interests as an extra source of information to characterize the end-users of the cloud services, so as to predict future demand. Efficient content placement and pre-fetching, or service mobility, e.g. migration of workload and Virtual Machines (VMs), can be thus supported. Orthogonally intersecting the aforementioned objectives, one of the key targets of SmartenIT is energy efficiency both in the Operator-/Provider- and the End-User-sides.

2.1 Summary of past work and TM selection process by SmartenIT

In order to address these objectives and targets, SmartenIT designed a broad set of incentive-based cross-layer traffic management (TM) mechanisms that handle inter-cloud traffic (cost-)effectively, employ social- and QoE-awareness, while achieving energy efficiency, in the sense that energy consumption is aimed to be kept low for data centers, networks or in end-users’ mobile devices. The set of TM mechanisms was initially introduced by SmartenIT in D2.2 [5]. The mechanisms were categorized in TM solutions addressing the Operator-Focused Scenario (OFS), and TM mechanisms addressing the End user-Focused Scenario (EFS). Operator-focused solutions are as follows:

- **Dynamic Traffic Management (DTM)**, which minimizes of inter-domain traffic cost in multi-homed AS by influencing the distribution of the traffic among links,

- **Inter-Cloud Communication (ICC)**, which reduces the 95th percentile transit charge by controlling the rate of delay-tolerant traffic, which a priori marked accordingly by the ISP’s business customer (e.g. cloud/datacenter), and shifting its transmission at off-peak intervals, and

- **Multi-Resource Allocation (MRA)**, which aims to ensure fair resource allocation among (federated) Cloud Service.

On the other hand, End user-focused solutions include:

- **Replicating Balanced Tracker and Home Router Sharing based on Trust (RB-HORST)**, which provides WiFi access to trusted users to offload their mobile traffic from 3/4G to WiFi, while simultaneously RB-HORST enabled home routers of trusted users identified through a social network are organized in a content-centric overlay network to support content prefetching,

- **Socially-aware Efficient Content Delivery (SEConD, former SECD)**, which employs social information, AS-locality awareness, chunk-based P2P content delivery and prefetching, and a centralized node acting as cache and as P2P tracker and proxy to improve the Quality of Experience of video streaming for users of OSNs,

- **Virtual Incentives (vINCENT)**, which aims to leverage unused wireless resources among users, while it ensures security by employing tunneling among trusted users,
- **Mobile Network Assistant (MONA)**, which schedules wireless data transmissions to reduce the energy expenses on the air-interfaces, and
- **Multi-Criteria Application Endpoint Selection (MUCAPS)**, which aims to improve users’ Quality of Experience by performing selection of communication endpoints by employing basic ALTO functionality, but also involving awareness on the underlying network topology.

For each mechanism, a set of use-cases (UCs), where that mechanism is applicable, based on the scenarios definition in [3], was defined, an initial description of the entities, modules and algorithms employed by the mechanism was provided, key influence factors and performance metrics were identified and initial evaluation results for certain mechanisms were presented. Moreover, in order to give a more complete view of the positioning of each mechanism w.r.t. the SmartenIT architecture defined in D3.1 [7], and the overall SmartenIT playfield, as defined by the key objectives and targets set in [3], the mapping of each single mechanism to the components and interfaces of the architecture, as well as an example instantiation of its operation have been provided in [5].

Accompanying the set of TM mechanisms, a set of models was also introduced in [5], which comprised two subsets: theoretical/mathematical models, and simulation models. The subset of theoretical/mathematical models included (inter alia) models for measuring QoE of applications such as online storage systems (Dropbox), and HTTP-video streaming (YouTube), energy-models, and models for estimating transit cost or pre-fetching accuracy. Moreover, mathematical models for cloud federation and VM migration among federated parties were developed. On the other hand, the set of simulation models included three models whose purpose was the evaluation of specific TM mechanisms or their components: i) a model for the evaluation of video dissemination among the users of an online social network (OSN), ii) a simulation model for HTTP-video streaming, and iii) a simulation model for online storage. Note that the focus of SmartenIT was and remains on both types of applications based on the investigation performed in [3].

Furthermore, in [5], an initial investigation of the synergies among the various TM mechanisms introduced was performed w.r.t. to the complete coverage of the SmartenIT playfield. In particular, the adherence of the TM mechanisms to the major scenarios of interest to SmartenIT was considered and a categorization of these mechanisms based on their properties, e.g. whether they employ data offloading or time scheduling techniques, etc., was performed. Based on the results of this investigation, two major groups of TM mechanisms have been identified: one comprises solutions dealing with content placement or uNaDas (nano data centers), i.e. HORST and RB-Tracker, SEConD (former SECD), vINCENT, MONA (former QoEnA), while other one includes solutions addressing inter-cloud communication, i.e. DTM, ICC and MRA. Hence, the groups, apart from reflecting the End-user-Focused Scenario and Operator-Focused Scenario definition that took place in D1.2 [3], also provide the high-level synergies of the traffic management mechanisms.

Out of the two groups of mechanisms described in [5], an initial selection/prioritization of the mechanisms and UCs was performed per scenario, as well as a selection of which scenario to address first, w.r.t. to the objectives and targets of the project. To facilitate the selection, a set of WP2/WP3 criteria were defined by WP2 in collaboration with WP3, which reflect the goals of the consortium, e.g. innovative intelligence, broad coverage of the SmartenIT playfield, ISP-friendliness, consistency with the architecture, reusability and maturity at the time of consideration. The criteria are presented in Table 2-1.
Table 2-1: Selection criteria for TM mechanisms to be implemented.

<table>
<thead>
<tr>
<th>WP2</th>
<th>Architecture (components and interfaces)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novelty and innovation</td>
<td>The mechanism is novel and promises impact.</td>
</tr>
<tr>
<td>Coverage of addressed scenarios+UCs</td>
<td>The mechanism addresses major axes of the SmartenIT work as described in DoW. TM, cloud applications, mobility, inter-domain, energy efficiency, social awareness, cost efficiency.</td>
</tr>
<tr>
<td>Strategic ISP-friendliness</td>
<td>The mechanism is ISP-friendly or ISP-aware. Addresses the incentives of ISPs, e.g. interconnection charges.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WP3</th>
<th>Consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>The mechanism employs major arch components and IFs. It is consistent with the SmartenIT architecture.</td>
</tr>
<tr>
<td>Reusability</td>
<td>The mechanism employs logic/components which are re-usable by other TM mechanisms in the future.</td>
</tr>
<tr>
<td>Maturity</td>
<td>The mechanism is specified in an adequate level, i.e. is mature enough at current time.</td>
</tr>
</tbody>
</table>

The outcome of the prioritization based on the aforementioned criteria is reported in Figure 2-1 and Figure 2-2. As observed in Figure 2-1, the UCs of highest interest were “Content service placement” and “Locality in uNaDas”, which are both associated with the End-user-Focused Scenario (EFS). The “Video content placement” and “Bulk data transfer” that are associated with the Operator-Focused one (OFS) were nominated as the third option. Nonetheless, all the other UCs were to be further developed as well yet following the aforementioned ones. The detailed specification of all UCs was documented in [6].

Concerning the prioritization of TM mechanisms, as observed in Figure 2-2, a synergy among HORST and RB-Tracker to create RB-HORST was selected as the first choice to address the EFS; the second choice was SECD (currently named SEConD). On the other hand, DTM was selected as the first choice to address the OFS, with ICC as the second option. As for UCs, all remaining mechanisms would be further developed at a later stage.

![Figure 2-1: Prioritization of UCs to be considered.](image-url)
Figure 2-2: Prioritization of TM mechanisms to be considered.

Moreover, during the 2nd year of SmartenIT, focus was be given on combinations (synergies) of multiple TM mechanisms so as to cover the most of the SmartenIT targets and objectives. Hence, as seen in Table 2-2, for the OFS, the integration of DTM with selected modules of ICC was decided, while for the EFS, a synergistic solution comprising RB-HORST and selected modules of three mechanisms: SEConD, vINCENT and MONA has been pursued.

Table 2-2: Additional properties (functionalities) for the extension of the DTM and RB-HORST TM mechanisms leading to the formation of synergetic solutions.

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTM ++</td>
<td></td>
</tr>
<tr>
<td>ICC</td>
<td>1. Core rate control algorithm (use of DTM data model).</td>
</tr>
<tr>
<td></td>
<td>2. Algorithm for DC selection (incl. energy efficiency aspects, pricing with or without a federation, collaborative with socially-aware prediction of SEConD).</td>
</tr>
<tr>
<td>RB-HORST</td>
<td></td>
</tr>
<tr>
<td>vINCENT</td>
<td>1. Dynamic inclusion of untrusted hotspots.</td>
</tr>
<tr>
<td></td>
<td>2. Tunneling.</td>
</tr>
<tr>
<td>SEConD</td>
<td>1. QoE triggered policy for SPS involvement in respective overlay.</td>
</tr>
<tr>
<td></td>
<td>2. Socially-aware chunk-based content prefetching (use social predictions of RBH).</td>
</tr>
<tr>
<td>MONA</td>
<td>1. a) Energy Models generated offline by energy measurements &amp; b) decision making algorithm about when to offload connections (based on energy consumed).</td>
</tr>
<tr>
<td></td>
<td>2. a) the actual offloading of connections using the uNaDas b) Requesting content predictions from the uNaDa (using RB-HORST) and pre-fetching these items to the handset while connected to the uNaDa (whenever it makes sense from an energy point of view).</td>
</tr>
</tbody>
</table>
2.2 Purpose of this Document

The main goals of this deliverable are as follows:

- **Objective 1:** To provide a deeper understanding of the technical and economic dependencies between the various stakeholders.
- **Objective 2:** To provide the final specifications of the incentive-based cloud traffic management mechanisms that have been proposed in D2.2 [5].

In the present deliverable, we investigate appropriate business models involving Cloud Service Providers, Data Center Operators and ISPs based on the consideration of economic and business dependencies, which was initiated in [2], and continued in [5]. These business models are mapped to UCs of interest to SmartenIT and practically generate the definition of new TM mechanisms and their modules.

Additionally, a multitude of theoretical and simulation models are presented, being either updates of the models introduced in [5], or entirely new ones. In particular, by means of the development and study of theoretical models, we investigate a multitude of design aspects such as incentives for cloud federation, content demand in online social networks and its topological characteristics, power consumption of specific equipment (such as the Raspberry Pi) and activity (such a mobile data transmissions), the effectiveness and impact of VM migration, as well as the virtual node model employed for WiFi offloading. These models will play a key role in the evaluation of (certain aspects of) the traffic management mechanisms, or are employed by certain modules of the TM mechanisms, e.g. the virtual node model is employed by vINCENT.

Moreover, we provide a deeper understanding of the technical and economic dependencies among the various stakeholders in the context of SmartenIT. To this end, investigation of incentives for cloud federation and resource sharing, either in datacenters, or in network-edge nodes, is performed. Moreover, concerning the cloud federation, new economic models including pricing schemes and tariffs are under development.

In this deliverable, final specifications of all TM mechanisms and of the aforementioned synergetic solutions are also provided. In particular, for each mechanism or synergetic solution, the intelligence including major entities and basic operation is described, while detailed specification including sequence diagrams and algorithms in pseudo-code is given in an appendix. Moreover, the associated parameters and metrics, and the evaluation framework for each mechanism or synergetic solution are defined based on the scenarios and use-cases, which have been identified as interesting, while also considering the special characteristics of each mechanism.

Furthermore, in this deliverable, we present and discuss the evaluation results for several individual mechanisms and synergetic solutions obtained by means of: i) theoretical evaluations employing the models developed for this purpose, and ii) through simulations employing either generic-purpose simulation environments (such as NS3), or specific and custom-made simulation frameworks of the proposed traffic management mechanisms. Although video streaming and online storage are mainly considered in the evaluations (following the prioritization of cloud applications that took place in D1.2 [3]), the TM mechanisms designed by the SmartenIT consortium are more generically applicable to other categories of applications as well. Final evaluation results, if not already reported in this deliverable, will be presented in the forthcoming deliverable D2.5 “Report on Definition of Use-cases and Parameters” due April 2015.
Finally, we discuss the coverage of the SmartenIT playfield by the developed traffic management mechanisms and the synergetic solutions derived thereof, while we consider the potential integration of multiple mechanisms in a cross-scenario approach.

2.3 Document Outline

This document is organized as follows: Chapter 3 provides the outcome of the investigation of appropriate business models; Chapter 4 describes the theoretical investigations including the definition of theoretical models, simulation models and business models. Chapter 5 presents the TMs that address the Operator-Focused Scenario and their synergies, where applicable. On the other hand, in Chapter 6 the TMs that address the End-user Focused Scenario and their synergies are described. Chapter 7 discusses the coverage of the SmartenIT playfield by the designed mechanisms which will be concluded in D2.5. Chapter 8 summarizes the deliverable and draws major conclusions on the specified TM mechanisms and the theoretical investigations. Finally, Chapter 9 reports which SMART objectives, as described in SmartenIT Description of Work (DoW) [1], have been addressed by the work performed in WP2 and reported in D2.4.
3 Business modelling

This chapter provides an overview of cloud business models and investigates their relation and interaction with the ISPs. In particular, we investigate the cross-layer interactions among the cloud and network layer stakeholders, the respective business agreements and value chain organization issues, as well as incentives and market power issues. We present several interesting business cases and models and then discuss their relation to SmartenIT models and mechanisms at the final section of this chapter.

It is worth noting that the business modeling overview of this chapter is intentionally kept at a high level. This is mostly due to the fact that business models inevitably cannot be completely separated from the respective products/services offered, market quantification, risks and opportunities, as well as the market and resource allocation mechanisms employed which can affect the market power of the stakeholders involved in the end-to-end product composition and delivery to the customer. This implies that there is a plethora of different value network and value chain structures, as well as market organizations applicable in different contexts, leading to multiple business model alternatives. Furthermore, even under specific use cases and assumptions, different stakeholders typically have different interests and strategies, hence leading to different preferable business models, especially in terms of control over customer ownership and service control, thus there cannot be an unbiased “optimal” business model proposition.

For all these reasons, the business modeling overview of this section is not a complete detailed overview of all business models alternatives. Instead, it is a high level overview and more detailed illustration of selected business model alternatives that are interesting for both the ICT market stakeholders and the project, have market potential and highlight the role and importance of the Cloud Provider-ISP interface also from a business perspective. This is of particular practical importance to SmartenIT since a plethora of SmartenIT mechanisms target this interface and can serve as catalysts for the materialization of the respective business models either in their present specification or by means of incremental extensions that could comprise future work.

Moreover, major risks and opportunities regarding unmet market needs, business evolution stumbling blocks and business models/strategies inefficiencies are identified and serve as the motivation for the identification of new roles that can potentially be relevant and promising, and can be undertaken by existing stakeholders or new comers in the market. Furthermore, the study to follow will reveal that the business cases and models overviewed motivate the theoretical investigations of the SmartenIT project, and also provide valuable feedback loop for the assumptions, context and specification of the SmartenIT models of Chapter 4. In particular, it is documented that the SmartenIT OFS mechanisms can serve as differentiated - in terms of statistical guarantees and expected quality – multi-domain network transport services, optimizing network resource usage, supporting multi-domain managed services and federations of datacenters, mitigating operators’ network and energy costs.

Though the SmartenIT mechanisms are designed to operate on top of Best Effort Internet, there is the potential that in the medium or long term the SmartenIT mechanisms can also take advantage of differentiated/guaranteed-QoS multi-domain network connectivity, where and when provisioned. The SmartenIT EFS mechanisms comprise attractive enhancements to current product offerings of software and hardware vendors who can benefit from the SmartenIT EFS capabilities at
the end user devices so as to promote energy efficiency and quality of experience especially within the emerging “terminal+service” and “HaaS+Internet” business context (presented later in this chapter).

In order to facilitate the reader, the subsequent material is organized in a hierarchical fashion: we start by providing the high-level big picture and we then proceed with a categorization and discussion of interesting business models. We focus on providing the business modeling key aspects and most important key features, rather than applying an exhaustive business model template for obvious brevity reasons.

### 3.1 Cloud Business Models Context

Cloud is not just a technology, but a novel services paradigm. To this end, there are multiple business modeling aspects inherent in Cloud since it is about a different way of bundling Information Technology (IT) and Communication Technology (CT) services in order to offer the Cloud-based ICT services.

The generic landscape of Cloud Business Models (BMs) is depicted in Figure 3-1: The main axes of differentiation are the level of sharing and the business value (i.e. the Value Proposition in the SmartenIT business modeling template terms). IaaS, PaaS and SaaS solutions target different customers, namely Operators/IT, Developers and end users respectively. Business Processes (BP) comprise a dynamic segment of the market as well. Each cell of Figure 3-1 depicts the dominant product offering for each type of cloud (Public, Hosted, Private) ranging from public pure infrastructure as a service in the upper left up to SaaS applications and virtualization tools for business processes.

![Figure 3-1: Taxonomy of cloud business model according to Forrester Research Inc.](image)

This taxonomy of Cloud business models allows us to have a high level view of the possible ways of offering cloud services, without looking into the exact structure of the value network or the value chain configuration. It is clear that the business value of the cloud services increases with the complexity of the service offered, since low-level
commodity services do not suffice to cover completely the respective customer business needs which become increasingly sophisticated over time. For each cell in the table above, certain instantiations of business models are relevant, while others are not. The market (and also technological) maturity are of prominent importance, while customer ownership and control over resources (and service orchestration) are crucial for the business models to emerge. Indeed, control over the customer and the end-to-end bundled ICT service offering results in higher business value for the customer and thus can be monetized substantially, as opposed to low-level commodity services. Last but not least, the Business Processes (BP) comprises an interesting market segment of increasing value and market size.

It is worth emphasizing that the business model under which cloud services are to be delivered is not just a technology issue but rather it is mainly a business/strategy issue. For instance, an e-health cloud service utilizing network and cloud resources can, from a technological point of view, be orchestrated from either a network stakeholder (an ISP) or an Over-The-Top (OTT) stakeholder (for instance a Cloud Service Provider) via proper SLAs and be provisioned efficiently. However, from a business point of view the way the end-to-end market value is to be apportioned among the value chain stakeholders will not be the same under each of the two possible configurations.

The above picture should be complemented with the constantly increasing role of Cloud Brokers, whose main Value Proposition is Brokerage: Brokers – as discussed later in this chapter – undertake the complexities of resource aggregation, brokerage and service orchestration in order to create high level marketable services.

A key business modeling issue is that as Cloud is becoming a key aspect and incarnation of ICT, it must be specified what will be the preferred business model in terms of service orchestration and customer ownership: thus the question is whether will ISPs be the dominant stakeholder adding cloud services to their service offerings, or will they remain commodity communication service providers, thus allowing the IT stakeholders to “invade” the CT world by offering cloud services on top of their networks? There are legitimate business models for both alternatives, thus it is a power game.

Having presented the big picture and main issues of the business modeling aspects, we now proceed to elaborate on some interesting business models that highlight the importance of the ISP-Cloud Provider interface from a business perspective. In the subsequent subsection we begin by elaborating on the y-axis of the above Figure and try to identify the role of the ISP (if any) at the resulting business models.

### 3.2 Cloud Computing, Brokers and ISPs’ role

This section discusses the spectrum of Cloud services in terms of ownership and openness. This is also depicted in Figure 3-2, with the two axis extremes being the closed private cloud and the open public cloud. Very interesting cases comprise the managed private cloud and the virtual private cloud, which are standard offerings for a variety of cloud service providers and ISPs. The Virtual Private Cloud in particular is typically considered the equivalent of Virtual Private Network of the CT world.
Figure 3-2: The spectrum of Cloud computing services. (Source: Gene Phifer, Managing Vice President at Gartner)

It is worth noting that with the exception of the enterprise private cloud, the role and importance of the ISP increases with the openness of Cloud. The provisioning of network connectivity that can efficiently and reliably serve the cloud services is crucial for the success of the end-to-end cloud service and greatly impacts the customer's satisfaction and thus willingness to pay. ISPs can play significant role in managed virtual clouds, a new service proposition as evolution of VPNs; this can be an additional service offering in their portfolio targeting their existing customer base that has both IT and CT needs. Interconnecting the Cloud IT infrastructure requires the active involvement of ISPs in large scale due to well-known inefficiencies of Internet: for instance, by-pass inefficient BGP routes, provision resources so as to guarantee Quality of Service (QoS), prioritize traffic possibly also in inter-domain layer.

Gartner [24] points out the following risks associated with cloud: Data/Process Location & Isolation (including Security, privacy & ownership), Regulatory, Compliance & Policies, Portability between Providers (Lack of standards, vendor lock-in), Provider Trust Management (Transparency to provider operations, immature vendors and certifications), Uncertain Failure Remediation (SLA guarantees, redundancy), Integration and Process Integrity across the cloud (Technical & Support issues), Bandwidth & Latency (Accessing or integrating “clouds”), Licensing Issues and Uncertain Financial Models. Cloud giants are already tackling some of the issues, via aggressive cloud service offerings over the markets of the IaaS, PaaS and SaaS model.

For completeness reasons, we provide also the benefits and the big picture with respect to different stakeholders, perspectives, benefits, as described in EU Future Cloud Computing Expert Group Report [25].
It’s easy for the definitions of public, private and hybrid Cloud environments to blur. In particular, service orchestration/control needs some kind of brokering, i.e. searching, obtaining, aggregating/combining cloud resources and services. To this end, the Cloud broker role is taking on new meaning as the Cloud market matures. The main motivation is the technology complexity and the lack of competences and resources in order to build, operate, administer and manage Cloud services: Cloud customers can’t blend together disparate cloud services on their own. Therefore, the role of brokers, undertaken by existing stakeholder or new entrants in the Cloud services market is increasingly important.

There are three main cloud broker roles:

**Aggregation**: A Cloud broker can bundle many individual services together and present them as a unified service. For instance, by partnering with a Cloud broker, a provider can offer a unified billing service or unified Cloud provisioning. Cloud aggregators tend to be companies with large, existing customer bases - such as Telcos and large IT distributors.

**Integration**: Integrate multiple services, collectively providing new functionality. The Cloud broker can help move data into the cloud and integrate the customer's network with the provider's network.

**Customization**: Usually around the network edge because cloud services can only be changed by the Cloud provider. As highlighted in [26], "Cloud customizers might take something from Cisco WebEx, an offering from Amazon and Microsoft Office 365 and aggregate (them) on a single bill for customers. Customizers integrate all the services to work together and sell this new offering under their own wrapper -- like 'Collaboration as a Service for small businesses,'“. While it's the cloud customizer's unique bundle of services
along with customization and proper parametrization of the service bundle so as to meet the individual customer’s needs, the new offering is powered by several disparate cloud services.

**Benefits to Cloud Providers**

Brokers are essential partner to Cloud Providers. A Cloud broker can combine the buying power of multiple enterprises, facilitating customer demand and purchases. It can also increase demand and negotiate better prices for its clients while delivering more customers to the provider. This comprises an attractive business model since each stakeholder remains focused on its core competences while brokers undertake the task of the middleman between supply and demand.

Appirio, for example, serves as a broker for enterprise customers that use Cloud services and platforms from Google, Amazon, Salesforce and Workday [27]. This is an interesting case for a broker that provides both consultancy and brokering services and that uses its partnerships and also crowd sourcing as well in order to provide customer-specific cloud solutions. Similar cloud brokers are Dell Boomi [30] and GXS [31].

![Appirio Cloud Brokering](image)

**Figure 3-4**: Appirio focus on brokering and building differentiated services.

There is potential for ISPs here to combine IT and CT resources and act as brokers-integrators, also increasing the competitiveness of their service offerings by means of:

- custom network services
- exploiting existing business relationships with customers of their pure CT services (e.g. VPN clients)
- bundling cloud brokering offers with network connectivity services so as to provide complete solutions to Small and Medium Businesses (SMB) with aggressive pricing.
The notion of Cloud service brokers will become very important to business, involving also carriers and telcos as depicted in Figure 3-5. Target markets are mid-size manufacturers, mid-size manufacturing related service business and independent E&P companies with interest in the notion of Digital Energy. There are a number of areas that seem to be key drivers for brokering:

- Differentiating services,
- Value added service creation,
- Taking customers to the cloud quickly but with control,
- Continually expand and innovate product and services,
- Being part of a wider cloud ecosystem,
- Leveraging existing assets via the cloud.

Forrester Research principal analyst, Stefan Ried argues that cloud brokering will enable new Cloud business models. Telcos can become a SaaS broker and create significant opportunity with SMBs, while system Integrators & Cloud providers can get the most out of the Infrastructure broker model [28].

**Service differentiation** comprises a key aspect of a successful business model for Cloud service providers, telcos and brokers. This is a key element that allows the provider to meet specific customer needs and attain high margin from the market, as opposed to simple low-value commodity services that solely serve as building blocks for more complex services. A portfolio of different highly specialized and differentiated services allows the provider to increase the resource efficiency and the revenue attained by targeting specific market segments. A representation of how organizations can generate value through high margin, differentiated cloud broker services follows, along with the evolution/transformation path of ICT services and business models from siloed to open Cloud broker environment, as depicted in Figure 3-6.
Figure 3-6: From low-value commodity services to high-value business solutions [29].

The path towards building a portfolio of competitive high-revenue differentiated services is an evolution process towards both business and technical maturity, as depicted in Figure 3-7. Brokers can serve as catalysts in this process by undertaking some of the complexity of this process and enabling firms to speed up their convergence to the target of Cloud maturity. SmartenIT solutions can act as catalyst to service differentiation, thus increasing the competitiveness of Cloud service offerings.

CIOs face both business and technical challenges in reaching cloud maturity

June 2012 “Cloud Brokers Become Change Agents”

Figure 3-7: Cloud business and technical evolution [33].
### 3.3 Example Business Model Cases

#### 3.3.1 CloudItalia

The SaaS model is considered to result in high opportunities for Telcos focusing on the SMB market. A prominent example is CloudItalia, an EMEA telecommunications provider. As also stated by Matt Davies, senior director of product marketing at Cordys in [34]: “Telcos have a significant opportunity with SMBs by adopting the role of SaaS broker. The Telco can encourage adoption of cloud services, open up new revenue streams and launch value added services to their SMB customer base. One of our recent customer wins was CloudItalia, an EMEA telecommunications provider. As with many operators – they faced declining revenues from traditional network services and the way their IT was deployed couldn’t meet the desire for growth through the cloud. Their customers and partners were looking for more integrated end-to-end IT solutions and CloudItalia saw a new business opportunity here”.

The strategic decision in this case was to adopt the role of cloud broker so as to federate, combine and aggregate IaaS, VMWare, their voice and data network, their OSS/BSS and other cloud services. This allowed them to offer new products and services “as a Service” to SMBs and gave them considerable business benefit including:

- New high margin revenue streams,
- Increased up-sell and cross-sell opportunities,
- Speed to market,
- Reduced operational costs.

Aggregation and service differentiation serves as an efficient form of business differentiation and control over a market segment, thus empowering new business models where traditional CT stakeholders undertake new roles. This is also evident in Huawei, TMForum and ChinaMobile BM propositions, which are presented in the following subsections.

#### 3.3.2 Huawei and China Mobile

The high costs incurred in multiple supplier/buyer transactions forms the basis for the introduction of a high-value agent service. This type of agent-based model can lower costs for both buyers and sellers, including monetary, mental, and physical expenditure. It is also a model that pervades daily life; traditional retailers, banks, stores, and real estate agents not only operate under this new model, but many of the big names in e-commerce are also its successful proponents, including Amazon, eBay.

According to Huawei this new business model applicable to cloud computing can be categorized as Hardware as a Service (HaaS), and Software as a Service (SaaS).

**HaaS** is based on the observation that an important catalyst driving Apple's success is the connection of its (hardware) products with Apple's virtual supermarket, the iTunes store that sells both Apple hardware and digital content and applications. A similar business model that also falls in this category is Amazon's Kindle Store that also allows users to download digital content, such as E-books, newspapers, magazines, and blogs. Therefore the successful combination of services with hardware is the prominent model envisioned.

In addition to HaaS, the **SaaS** business model also falls under Cloud computing. SaaS provides software services through the Internet under which a vendor deploys application
software on servers. Users can rent the application software service they need online, and pay according to service quantity, type, and duration. Note that under this business model, the service provider, not the user, undertakes complete software maintenance and management.

The SaaS leasing feature negates the need for would-be users to invest heavily in hardware, software, maintenance, constant upgrades, and staff. Thus it is a profitable business model in terms of network applications. Both individuals and enterprises can access a large range of software services online, and only pay for what they use when they use it. Companies such as Salesforce, Google and Microsoft provide SaaS.

**HaaS and SaaS Interplay**

The core of a very interesting business model is "hardware+software". Hardware is the service carrier, while software forms the specific service. For example, a GPS device is the carrier, while the GPS software provides the location service. The core of SaaS is "software+Internet". Traditional software that requires licenses is transformed into online software that provides services through the Internet. Cloud computing is thus a service for end users; metaphorically, the hardware and software are obscured by a cloud that is neither reachable for users, nor is it their concern.

**Influence on Telcos**

While HaaS, SaaS, and cloud computing first emerged in IT, they influence CT. As stated by Huawei [35], “Telcos' traditional walled garden is being breached by uninvited guests”. They use commodity network services provided by telecom operators, and gain the most profits from their own value added services. This process is gradually marginalizing operators into becoming merely wireless broadband channel providers. Thus, it is apparent that service differentiation, customer ownership and service orchestration are key factors here as well.

China Mobile also positions themselves as Digital/Cloud Services Broker/Integrator: According to China Mobile president “In industrial integration, China Mobile is a digital content integrator, editor, and distributor. The boom of mobile multimedia services has encouraged us to become information integrators.” China Mobile is also very interested in the “terminal+service” model, thus utilizing the Telco customer base for providing OTT-like (including cloud) services, mitigating the potential of OTT competitors to be the providers of value added services [36].

These new trends and respective business models also affect the SmartenIT research agenda. The increasing interest on the “terminal+service” model leaves room for innovation at the end user devices in terms of energy efficiency, advanced services and resource management, increased agility and performance. These features clearly fall in the scope of the SmartenIT EFS mechanisms, which can be integrated and customized by the provider to the user terminals for improving the quality of the services delivered in terms of energy consumption, delay, responsiveness and QoE.

### 3.3.3 TM Forum

Not surprisingly, the TM Forum has a similar to Huawei and China Mobile positioning and has explicitly expressed interest in “copying” the Amazon Store business model for cloud services, also specifying a list of BMs - under a structured business model canvas - per service; the TM forum business model approach is presented below.
The TM Forum rationale is based on two key observations/statements: the first one is that Amazon established its architecture to leverage their assets for implementing a wide range of business models in a repeatable way for the retail industry. The second one is that the digital services industry is still missing the architecture to leverage assets of communication service providers in a repeatable manner. These observations are followed by a rather strong statement which highlights the big concern of ISPs regarding the way cloud-based digital services are to be offered: “e-Health, SmartGrid, Connected Home, will all become OTT projects unless Telcos find a way to offer an Amazon-type repeatable Digital Services Provider capabilities” model [36]. Clearly this is a biased statement based on the TM Forum participants, which also is indicative of the fight over cloud services market control.

Having presented the motivation behind their positioning and avoiding an exhaustive presentation of the various business model canvases provided in [36], we briefly comment on the high-level concepts of the business model that TM Forum proposed to be copied from Amazon for cloud-based digital services. We begin the presentation by reminding the reader the core of the Amazon business model: this is also a multi-tenant environment that sits on top of a common functionality that is not restrictive for the tenants (retailers) to shape their stores and service/produce offerings. The Amazon Retail common functionality comprises “Concept to Cash” shopping cart with revenue collection and tenants get independent positioning, product management and transaction capabilities, as depicted in Figure 3-8.

![Figure 3-8: The Amazon repeatable business model [36].](image)

TM Forum argues in favor of transferring this repeatable business model to cloud and digital in general services where the Telcos have the dominant position in service control, orchestration and offering, instead of being used in a commoditized way by OTTs (cf. Figure 3-9).
This basic business model structure can efficiently support multiple services and use cases, but due to the inherent complexity of cloud services there are additional requirements, such as: Easy to set up new tenants, Tenants need E2E product management, Tenant configuration of business rules, Open Standards, Legacy / Any Orchestration. We provide in Figure 3-10 a high level example of the common architecture and the instantiation of a specific service.
Once again, the differentiation of services and a rich portfolio of different services is the dominant strategy for the TM Forum members. It is worth noting that this differentiation can also be extended to the network layer. Though these services can be efficiently provisioned – to some extent – in an intra-domain context through proper network management, it is also possible to offer differentiated network connectivity services. Such services could range from service-aware routing of traffic using e.g. GRE or well-provisioned MPLS tunnels to bypass routing inefficiencies and enhance the customer experience per service, thus allowing a high monetization of the respective end-to-end Cloud service offered. The SmartenIT OFS solutions can be of high value here.

### 3.4 ISP Managed Services

A list of sample existing service offerings of high value proposition due to cost reductions/economies of use/low investments-pay-per-use model advantages follows:

- Virtual Private Cloud/Data Centre (offered by many ISPs e.g. AT&T, IRT, GEANT),
- Managed Storage and Backup,
- Independent Software Vendor (ISV)-SaaS,
- Managed Security Services,
- Managed Servers and Virtualization,
- ICT for SMEs.

This is an indicative non-exhaustive list and there are multiple variations across providers in terms of both naming and content. We restrict attention to actual business cases, and refrain from elaborating on all those services; instead we provide some indicative business cases and use them to elaborate on the business modeling aspects and the importance of the cloud-ISP interface for some of them in the subsections to follow.

#### 3.4.1 The Virtual Private Cloud Business Case

The Virtual Private Cloud comprises a standard offering that highlights the importance of the ISP-cloud interface. It is considered an evolution of the VPN CT service and multiple European and US-based Telcos offer this service as part of their portfolio, e.g. Interoute, AT&T. The key proposition/service offering is the combination of Virtual Data Centre (VDC) technology (AT&T uses VMware products) with VPN over its network to construct a "Virtual Private Cloud". This virtual cloud can be entirely hosted or the hybrid case is also possible with the customer combining its in-house datacenter resources with the Virtual Data Center resources, also possibly on demand so as to reduce costs and maximize the efficiency of the infrastructure. This service for AT&T is referred to as “AT&T Synaptic Compute as a Service” [37], while for Interoute is referred to as Virtual Data Center and accessed via Interoute Cloudstore [38].

The service allows VMware’s more than 350,000 customers to extend their private clouds across and into AT&T’s network-based cloud using AT&T’s VPN. The benefits of this “virtual private cloud” include the flexibility and cost efficiencies of using private and public cloud systems interchangeably and strategically. This is an excellent example of a business model where the Telco controls service aggregation and orchestration. It combines an in-house product (VPN) with a cloud product (VMware suite) to make an attractive service offering.
Thus this demonstrates the preferable model for a Telco business where customer ownership and service control resides in the Telco side that via an aggressive marketing strategy prevents OTT from invading its value added service offerings to its customers.

Note that in Europe where a multi-provider setting is the typical case for multi-national EU companies, it would be difficult for a single provider such as AT&T to provide such services on pan-European scale. To this end, SmartenIT solutions could be of high value in order to efficiently manage the network traffic across multiple domains (via OFS mechanisms) in order to support the communication among multiple VDCs. A "simpler" version of this service is the Virtual Data Centre use case (e.g. the IRT product offering) that is provisioned within the boundaries of an ISP on top of dedicated network core infrastructure.

### 3.4.2 The Federated Cloud business case

This is a business case that aims at Data Centre interconnection via a federated infrastructure. Arzuna has provided the Agility framework: this is the “glue” of the federated cloud [39]. The main value proposition is that it enables organizations to be connected together in order to form a federation via common policy and SLA semantics, as depicted in Figure 3-14 and Figure 3-15. Commonly accepted SLAs are important in order to provide accountability and ensure that resource control remains within the administrative entities. The same framework applies in the context e.g. of interconnecting DCs via setting up common policies for Condor schedulers [40]. Though this is an OTT service, clearly there are huge benefits if ISPs are involved in the process since they can efficiently provision the federated infrastructure with the required CT infrastructure. Once again, SmartenIT Traffic Management solutions could be of high value here in order to provide differentiated connectivity services, by means of using efficiently multiple tunnels in order to forward traffic and also delay if needed the portion of traffic that is non urgent so as to prioritize the more demanding in terms of performance and response times Cloud services.

![Figure 3-11: The Arzuna Agility high-level approach [39]]
This commercial service offering and business model is very similar to the FP7 EU-ICT CONTRAIL project [41] in a technical sense but the main difference is the awareness of the underlying administrative entities that they belong in a federation. Thus, there is room for common policies and more active “collaboration”. Clearly there is big vertical market integration potential for ISPs in this business case, thus further diversifying the value chain configurations and resulting business models.

3.4.3 The Open Market Business Model

A very interesting business model is that of the open market: this approach envisions a new role, that of the marketplace, which aggregates providers and sellers of cloud and network resources and services. The marketplace role is typically assigned to a third party entity that is not actively participating in the market transactions due to obvious trust reasons. Typical examples of the approach are the Deutche Boerse Cloud Exchange [42] and the UK-based CloudStore, which is the market for G-Cloud, the UK public sector cloud services [43]. In G-cloud, all types of services are supported, public, private and hybrid clouds and IaaS, SaaS, PaaS and Specialist Cloud Services (SPCS) models. The main motivation is a unified market where economies of scale are achieved. Indeed, there are £175,479,562.12 of reported G-Cloud sales up to the end of March 2014; 60% of total sales by value and 59% by volume, from all reported G-Cloud sales to date, have been awarded to SMEs.

Similar case is the Deutche Borse Cloud Exchange, depicted as Figure 3-16, where advanced economic mechanisms are envisioned, namely spot and futures markets. These market mechanisms are common for the trading of stocks and electricity and enable the trading of sophisticated service products where the dimension of time has been also considered, thus supporting multiple time scales and differentiated service offerings.
3.5 Cloud Business Models and SmartenIT Interplay

In the previous sections we have overviewed a variety of different cloud business models and highlighted the (potential for) involvement of the ISP in each of them. As already explained, the ISPs by definition control the network infrastructure, which is required for the Cloud Service Providers to provision their products to their customers. This suffices to ensure that the ISPs will be included in the respective market value chain for the provisioning of Cloud services, especially since the network performance can greatly impact the overall end-to-end service quality and thus the monetization of the respective service. We have also briefly mentioned business cases motivating smart mechanisms at the end user terminals side, which also are well in the scope of SmartenIT research.

The extent of the ISP involvement will mainly depend on which stakeholder controls the service orchestration and the respective market power of the stakeholders, i.e. mainly ISPs and cloud service providers. A crucial factor will be the stakeholders’ capability to efficiently construct, sell and provision end-to-end differentiated market solutions for the (business) customers.

This differentiation could also be built around the Cloud Operator-ISP interface in order to develop new business models that are both relevant and plausible (e.g. the Open Market or Arjuna Agility business models clearly depict the importance of the ISP involvement). Thus, the business models examined can both interplay with and benefit from the SmartenIT Traffic Management solutions.

We now briefly illustrate the various layers of service offerings and pricing involved and the potential for innovative SmartenIT mechanisms-service offerings.

The SmartenIT OFS traffic management solutions could be used in the network layer in order to efficiently use the network infrastructure and support federated clouds or dispersed (V)DCs on top of multiple ISP hostings over the Best Effort Internet. In particular, the SmartenIT mechanisms can serve as a differentiated network transport mechanism, optimizing the network and the usage of its resources, also taking into account energy efficiency aspects. This SmartenIT “toolbox” can mitigate operators’
network and energy costs, also improving the quality of the network and the Cloud services that the users’ experience, thus increasing both the competitiveness and the quality of the services offered. This is particularly important for Europe where:

a) intra-domain network solutions (that could e.g. be easily applicable in the US) in general do not suffice to meet the European industry and users’ needs due to the multiple ISPs involved in cloud service delivery across Europe, and

b) European Cloud Operators/Datacenters are of small and medium size but if efficiently interconnected, managed and shared or federated could comprise an attractive alternative to large US cloud service providers.

The impact of the SmartenIT TM mechanisms on the resource management, energy efficiency and Cloud-ISP interface traffic management fronts can allow efficient connectivity among the IT resources available in Europe, thus making them usable for pan-European aggregations and testbeds/commercial uses. This, combined with market institutions such as open markets where IT and CT services are traded, can promote competition and allow for the efficient roll out of new Cloud services on top of the European ISPs networks. Once again, any of the aforementioned business models could serve this purpose.

For instance, a hybrid Virtual Private Cloud could be supported: companies can bid for clusters and storage at open market exchanges and be prices accordingly. Then, network connectivity among the purchased resources can either be bought via similar institutions or the Best Effort Internet in an ISP-friendly way with mechanisms such as ICC and DTM. The network charge for the bulk data transfers of the VPC example can be significantly reduced for both the customer who interconnects its Cloud resources and the hosting ISPs due to the properties of the traffic management mechanisms that lower the 95th percentile charge. Thus there will be two distinct layers of charging: a wholesale network layer charge over slow time scales and a faster on-demand IT layer charge for the Cloud resources bought and consumed.

Furthermore, the analysis of this chapter indicates the increasing importance of predictable service quality, and thus the provision of some type of guarantees for the service. To this end, there is an increasing use of SLAs among operators and business customers, possibly invisible to the end user, that attempt to mitigate uncertainty over the performance of the ICT stakeholders and their services that are aggregated in order to provide the end-to-end service to the customer. This has motivated alliances and projects such as SLA@SOI [46], CONTRAIL or even companies [32] to develop their SLA frameworks. We consider this work to be one of the critical technical foundations upon which the business models studied in this chapter can be supported. However, the overview and analysis of the various SLA frameworks remains outside the scope of this deliverable.

Such SLA frameworks, along with novel differentiated network services frameworks and architectures such as the one proposed by the EU FP7 ETICS project [47] for QoS-aware interconnection can serve as catalysts also for business models and services that by nature are “zero-tolerance” cloud services or heavily rely on stringent QoS requirements (e.g. on-line gaming). This technology paradigm is closely related to SmartenIT mechanisms: though the SmartenIT mechanisms are designed to operate on top of Best Effort Internet, there is the potential that in the medium or long term the SmartenIT mechanisms can also take advantage of differentiated network services where and when provisioned: for instance the SmartenIT OFS mechanisms could be extended to perform traffic scheduling for more than the two current (time-shiftable and real-time) classes of
traffic also taking advantage of better tunnel connectivity for the optimal traffic management.

The new “terminal+service” and “HaaS+Internet” trends and respective business models also depict an increasing interest on innovation at the end user devices side. Thus, the SmartenIT EFS mechanisms that can deliver energy efficiency, advanced services and resource management, increased agility and performance can be useful means for new differentiated services and terminal operations. For example, the case of Fon is driven by the operator's business model to take advantage of the end-user terminal equipment so as to expand their coverage areas. In the same sense, apart from connectivity, other resources like computation and storage that are already present at the edges of an ISP’s domain (i.e. at customers’ premises) could be used. Another new and relevant business model includes the cloudlets business case [48]. Cloudlets are decentralized and widely-dispersed Internet infrastructure whose compute cycles and storage resources can be leveraged by nearby mobile computers, i.e. a “data center in a box”. The focus of SmartenIT EFS mechanisms, nano-datacenters and end user terminals is well positioned within this emerging business context.

The mapping of the SmartenIT EFS and OFS mechanisms to the business cases and models overviewed in this chapter is provided as Figure 3-14. This classification shows that the SmartenIT EFS mechanisms serve as a network connectivity services layer allowing new differentiated network services that could be provisioned under the Network as a Service paradigm; on top of these services, federation can be established among multiple clouds/datacenters, as well as managed services on a pan-European scale. This layer of TM mechanisms can also be the fabric on top of which infrastructure, platforms, or services can be efficiently provisioned, with differentiated performance. The EFS mechanisms are most suited to the smart terminal+service model plus end-user side ISP services under the Cloudlets paradigm. Note also that the business models of this section motivate and provide feedback to the SmartenIT theoretical investigations of Chapter 4. The respective mapping is provided a Figure 3-15.

![Figure 3-14: Mapping of SmartenIT mechanisms to business models.](image-url)
Concluding, the business modeling issues highlighted in this chapter indicate the potential of the SmartenIT to contribute in the emerging Cloud services landscape and support smart end user terminals and services as well as efficient ISP-friendly traffic management on top of Internet, thus reducing administrative overheads and improving services performance. The SmartenIT mechanisms comprise also an attractive product for a) the European ISPs who can use SmartenIT OFS solutions to cope with the energy efficiency challenges of the ICT infrastructure, the increase of the traffic in their networks and in order to differentiate their service portfolio with sophisticated bundled cloud-network service offerings on top of either best-effort or premium differentiated network connections in order to remain competitive in the market and b) software and hardware vendors who want to incorporate SmartenIT EFS capabilities at the end user devices so as to promote energy efficiency and service quality of experience.
4 Theoretical investigations

In this chapter, we investigate a multitude of technical issues such as the effectiveness of caching in the case of distributed cloud-based content delivery systems, content demand in online social networks and its topological characteristics, power consumption of either specific equipment (such as the Raspberry Pi) or activity (such a mobile data transmissions), effectiveness and impact of VM migration and cloud federation. These models will facilitate the evaluation of (certain aspects of) the traffic management mechanisms.

Moreover, investigation of incentives for cloud federation and resource sharing, either in datacenters, or in network-edge nodes, is performed. Especially, concerning the cloud federation, new economic models including pricing schemes and tariffs are under development. Furthermore, based on the consideration of economic and business dependencies, which was initiated in [2], and continued in [5], we define appropriate business models involving Cloud Service Providers, Data Center Operators and Network Operators (or ISPs).

4.1 Investigation of incentives for cloud federation

Cloud computing technology aims to provide ubiquitous, convenient and on demand network access to a shared pool of computing resources. These resources can be virtual machines, storage, applications or services that can be dynamically provisioned and managed. So, the users can access and deploy applications on the cloud, from anywhere in the world, on demand. Also, companies that offer services and companies that develop software can take advantage of the cloud by leasing flexible and scalable resources or services (IaaS or PaaS), avoiding to create their own cost intensive hardware and software infrastructures.

In order to support the service demand all around the world, the Cloud Service Providers (PaaS or IaaS) have established datacenters in multiple geographical locations to provide redundancy and reliability. Although Cloud Service Providers (or simply Cloud providers) promising flexible and scalable resources giving the impression to their customers that they have infinite resources, there is no single Provider providing dynamic scaling or able to establish datacenters in all possible locations around the world to satisfy the demand.

As a result, Cloud Service Providers have difficulties in meeting QoS or QoE expectations of their customers. Hence, they would like to make use of multiple Cloud providers in order to better serve their users’ needs. Consequently, the formation of Cloud providers’ federation is required, where a number of providers cooperating to provide seamless provisioning of services. In fact, there are several incentives for Cloud providers to form a federation, such as inter-cloud load balancing, end-user QoE improvement, geographic coverage expansion for small providers and energy saving through host's utilization. In fact, there are several incentives for both large and small Cloud providers to join a federation (see also [49]-[53]).

First of all, through the federation each provider can expand his geographic footprint and have really global coverage. Moreover, they do not need to over-dimension their datacenters, since dynamic inter-cloud load balancing can be achieved by outsourcing jobs to the federated providers, in order to handle unexpected events of peak demand. The jobs outsourcing within the federation, can be also used in order to achieve reduction in energy costs, through hosts’ utilization. Furthermore, end-users’ QoE can be improved,
since a user can be always served by the Cloud provider with the most appropriate infrastructure environment.

Finally, the enterprises can avoid both the technological and business “lock-in” in a single Cloud provider. The different Cloud federation architectures proposed in the literature can be classified into three different models as follows: 1) Cloud aggregation, where different providers cooperate to integrate their cloud infrastructures into a unique virtual facility. 2) Hybrid Clouds, which combines the existing on premise infrastructure (private cloud) with resources of one or more public clouds. 3) Brokering, where the Cloud federation connects the multiple international and local Cloud providers to a global marketplace that enables each participant to buy and sell capacity on demand.

In this section, we present a simple model for resource management in a federation of Cloud providers. Then, we are considering how we can relate our approach to our proposed traffic management mechanisms. Since our model investigate the resource sharing and multi-objective load migration within cloud federation, is related to ICC mechanism and specifically to the cloud layer. This model could be an additive part on the Cloud scheduler of ICC, operating as balancer or re-allocator of the incoming requests of all cloud providers. Finally, we consider the relation of our model with the existing business models. In the proposed model, multiple Cloud providers cooperate to serve their customers in more effective way. These Clouds are publicly accessible, thus the model is related to the public cloud section services. Moreover, our approach is related to the federated cloud business case (cf. Section 3.4.2), since SLA agreements between the federated providers should be established.

4.1.1 Cloud Providers as M/M/1 system

Taking advantage of queueing theory, we are able to model the IaaS that a Cloud provider offers to his customers. Therefore, in our model, each Cloud provider is taken as a single datacenter with only one queue of incoming requests and one server to serve them.

In particular, we assume each Cloud provider as a M/M/1 queueing system where the incoming requests arriving with a Poisson process of rate \( \lambda \) (jobs/sec). Therefore, the server should spend computational power (processor flops), in order to fulfil each of these requests (jobs). We assume that the number of processor flops required for each job to be completed, amounts to an exponentially distributed number \( L \). Since, the server is taken to have a computational capacity of \( C \) (flops/sec), the service rate \( \mu(C, \bar{L}) \) of the Cloud provider is a function of this capacity and of the average number of flops \( \bar{L} \) required for the execution of a job. Therefore, the service time of a job is exponentially distributed and equals to \( \mu \). Additionally, the connection between service time and \( C, \bar{L} \) is given by \( \frac{1}{\mu} = \frac{\bar{L}}{C} \). The representation of a Cloud provider as an M/M/1 queueing system is depicted in Figure 4-1.

![Figure 4-1: A single Cloud Provider as an M/M/1 queueing system](image)
Based on Little’s Law for M/M/1 queues, we can estimate the average delay an incoming request will have for the moment inserting in the queue until it is finally served:

\[ d = \frac{1}{\mu - \lambda} \]

In order to present our model, we consider two Cloud providers \( P_1 \) and \( P_2 \) that operating separately, i.e. each one of them serve only the requests coming from their own customers (see Figure 4-2). For \( P_1 \) in particular, the requests are taken to arrive with a rate of \( \lambda_1 \) and its server is assumed to complete them with a service rate \( \mu_1 \). For simplicity, we assume that the server has a fixed over time computational capacity \( C_1 \) (flops/sec). However, the service time \( \frac{1}{\mu_1} \) remains exponentially distributed because of the exponentially distributed number of flops of each jobs \( L_1 \). Respectively for \( P_2 \), we define the parameters \( \lambda_2, C_2, L_2 \) and \( \mu_2 \). Finally, we refer to the total arrival rate in the system of two provider as \( \lambda = \lambda_1 + \lambda_2 \).

![Figure 4-2: The two Cloud Providers operating separately.](image-url)

Using again the Law of Little for M/M/1 queues, we can estimate the average delay for the incoming requests of each provider separately, but also the average delay for all the incoming requests of the system of two providers. Considering the requests of the customers of \( P_1 \), they arrive with a rate \( \lambda_1 \) and served with a rate \( \mu_1 \). Respectively, the requests of the customers of \( P_2 \) arrive with a rate \( \lambda_2 \) and served with a rate \( \mu_2 \). Therefore, the average delay for the incoming requests of \( P_1 \) and \( P_2 \), \( d_1 \) and \( d_2 \) are:

\[ d_1 = \frac{1}{\mu_1 - \lambda_1}, d_2 = \frac{1}{\mu_2 - \lambda_2} \]

Also, if we apply the Little’s Law to the system as a whole, then the total system’s average delay is given by the formula below:

\[ d_{tot} = \frac{1}{\sum_{i=1}^{\infty} \frac{1}{\lambda_i}} \sum_{i=1}^{\infty} \frac{\lambda_i}{\lambda_i} d_i. \]

### 4.1.2 Federation of Cloud providers

In our model, we consider the federation of Cloud providers in the sense of computational resource aggregation. In the case where \( P_1 \) and \( P_2 \) form a federation, one or both of them can serve requests for the other if it is beneficial according to some criteria. As a first step, we study the federation of two providers in a static mode, where we assume that we have full information for the arrival rates of each provider's incoming requests and the service rate of their servers. In our approach, we aim to achieve a resource allocation that
minimizes the total average delay of all requests arriving to the federated Cloud providers. Moreover, we assume that all the incoming requests have the same QoS requirements and each request is just a part of the total incoming workload. In other words, we try to achieve an average and not request-specific optimization.

As an example, we assume the one-way federation scenario (see Figure 4-3), where $P_1$ is more heavily utilized than $P_2$. Thus, in order to achieve load balancing, provider $P_2$ should serve a part of the incoming requests of provider $P_1$. To this end, we define a parameter $\alpha$ (where, $0 \leq \alpha \leq 1$) such that the part of requests of $P_1$ that $P_2$ should serve has arrival rate $\alpha \lambda_1$. Since this outsource part of the requests of $P_1$ will be served by the M/M/1 system of $P_2$, the arrival rate of requests inserted into both Cloud providers’ queues will eventually change. Especially, the arrival rate of requests will be served by $P_1$ becomes $\lambda'_1 = (1 - \alpha) \lambda_1$ and the rate of requests will be served by $P_2$ changes to $\lambda'_2 = \lambda_2 + \alpha \lambda_1$. As a result, the new arrival rates will affect the average delay introduced by each Cloud provider’s M/M/1 queueing system.

Moreover, the requests of $P_1$ that are ultimately served in $P_2$ are assumed to be transferred over the Internet, and thus incur network delay. This extra delay for outsourcing the requests is not constant, since the network conditions are different over time. Thus, we introduce to our system a new M/M/$\infty$ queuing component $D$, in order to represent the extra delay will occur by outsourcing the requests to another provider. Note that, the outsourced rate $\alpha \lambda_1$ is only a small fraction of total arrival and service rate of $D$. Also, this input sub-process of $P_1$ outsourced requests arriving at the queue of $P_2$ is also Poisson, due to the modelling of the delay element $D$ as M/M/$\infty$. In the current state of our model, we assume that the component $D$ introduces an average delay $d$, which is equal to the average service time of the M/M/$\infty$ system and independent of $\alpha$.

![Figure 4-3: One way federation of two cloud providers](image)

Hence, the new average delay of the customers of each provider within the federation depends on the value of $\alpha$. Eventually, the value of $\alpha$ also affects the total average delay of the system of two Cloud providers. Using again the Little’s Law for M/M/1 systems, we can estimate the average delay of the customers of each provider separately, and then the
average delays for all the customers in this one-way federation of the two providers. Then, we present the average delays as function of $\alpha$:

**Average delay of $P_1$ customers**

$$d_1(a, d) = \text{Avg. delay of requests served by the } M/M/1 \text{ of } P_1$$

$$+ \text{Avg. delay of requests served by the } M/M/1 \text{ of } P_2$$

$$+ \text{Avg. network delay of outsourced requests}$$

**Average delay of $P_2$ customers**

$$d_2(a) = \text{Avg. delay of requests served by the } M/M/1 \text{ of } P_2$$

**Total average delay of customers in the one-way federation**

$$d_{\text{tot}}(a, d) = \frac{1}{\lambda} (\lambda_1 d_1(a, d) + \lambda_2 d_2(a))$$

Since we aim to minimize the average delay of all customers in the federation, we have to solve the following minimization problem:

$$\min_{\alpha} d_{\text{tot}}(a, d)$$

$$\text{s.t. } 0 \leq a \leq 1$$

$$0 \leq d$$

Solving this problem, we can find the optimal value of $\alpha$ that minimizes the average delay of all customers in the system. This optimal value $\alpha^*$, depends on the value of network delay $d$. In particular, $\alpha^*(d)$ is decreasing in $d$ and if $d$ exceeds a threshold $d_{\text{max}}$, then $\alpha^*(d)$ becomes zero. Figure 4-4 depicts the values of $\alpha^*$ as $d$ increasing, in a one-way federation example.

![Figure 4-4: Values of $\alpha^*$ in $d$, for $\lambda_1 = 2\lambda_2$ and $\mu_1 = \mu_2$.](image)

Previously, we described the case where only $P_1$ was able to outsource requests. The same steps we would follow in a similar scenario, where only provider $P_2$ would outsource requests to provider $P_1$. On the other hand, in the **full federation** scenario where both providers can outsource a part of their incoming requests, the optimization problem
becomes more complicated (see Figure 4-5). In this case, apart from $\alpha$ we have to define another parameter $\beta$ (where, $0 \leq \beta \leq 1$) which represents the part of the incoming requests of $P_2$ outsourced to $P_1$. Therefore, the values of $\lambda'_1$ and $\lambda'_2$ depends both on $\alpha$ and $\beta$, having $\lambda'_1 = \lambda_1 - \alpha \lambda_1 + \beta \lambda_2$ and $\lambda'_2 = \lambda_2 + \alpha \lambda_1 - \beta \lambda_2$. Similar to the one-way federation scenario, the M/M/$\infty$ queueing component $D$ introduces a fixed network average delay in both directions of providers’ communication $d_{P1P2}$ and $d_{P2P1}$. For simplicity, we assume same average delay in both directions and equal to $d$.

![Figure 4-5: Full federation of two cloud providers](image)

Similar to the one-way federation scenario, we aim to minimize the system's average delay by finding the optimal values of $\alpha$ and $\beta$. Therefore, we can again produce the full federation average delay functions, by taking again advantage of Little’s Law and by introducing the network delays due to outsourcing:

**Average delay of $P_1$ customers**

$$d_1(a, \beta, d) = \text{Avg. delay of requests served by the } M/M/1 \text{ of } P_1 + \text{Avg. delay of requests served by the } M/M/1 \text{ of } P_2 + \text{Avg. network delay of outsourced requests from } P_1 \text{ to } P_2$$

**Average delay of $P_2$ customers**

$$d_2(a, \beta, d) = \text{Avg. delay of requests served by the } M/M/1 \text{ of } P_2 + \text{Avg. delay of requests served by the } M/M/1 \text{ of } P_1 + \text{Avg. network delay of outsourced requests from } P_2 \text{ to } P_1$$

**Total average delay of customers in the one-way federation**

$$d_{tot}(a, \beta, d) = \frac{1}{\lambda} (\lambda_1 d_1(a, \beta, d) + \lambda_2 d_2(a, \beta, d))$$

Again, we aim to minimize the average delay of all customers whose providers participating in the full federation. Thus, we have to solve the following minimization problem:
\[
\min_{a, \beta} d_{\text{tot}}(a, \beta, d)
\]
\[
s.t. 0 \leq a \leq 1
\]
\[
0 \leq \beta \leq 1
\]
\[
0 \leq d
\]

Solving the above minimization problem for different values of network delay \(d\), we can find the optimal values of \(a\) and \(\beta\) for each value of \(d\). Same as in the one-way federation (Figure 4-3), these optimal values \(a\) and \(\beta\) are decreasing in \(d\) and both become zero if \(d\) exceeds a threshold \(d_{\text{max}}\). In the general case of the minimization problem, for a given value of non-zero network delay \(d\), there is only one optimal pair \((a^*, \beta^*)\). However, in the special case of zero network delay \((d=0)\) we don't have any extra delay for outsourcing, hence there are more than one optimal pair of \(a\) and \(\beta\) that minimize the average delay of the system. In particular, the solution is an optimal pair \((\alpha^*(\beta), \beta^*)\), where \(\alpha^*\) is an increasing function of \(\beta^*\) \((0 \leq \beta \leq 1)\). Figure 4-6 depicts the values of \(\alpha^*\) and \(\beta^*\) that minimize the average delay of total system, for known arrival/service rates and network delay \(d=0\). Finally, since we assume that all the requests have the same QoS requirements and we only treat them a part of the incoming workload, in the optimal solution at least one of \(a\) and \(\beta\) is equal to zero.

![Graph showing the values of \(\alpha\) and \(\beta\) that minimize the average delay of total system, for \(d=0\), \(\lambda_1=2\lambda_2\) and \(\mu_1=\mu_2\).](image)

Figure 4-6: Values of \(\alpha\) and \(\beta\) that minimize the average delay of total system, for \(d=0\), \(\lambda_1=2\lambda_2\) and \(\mu_1=\mu_2\).

### 4.1.3 Stakeholders and Pricing

In the previous sections, we assumed that we had perfect information for our system and we made the optimal decision for outsourcing acting as coordinators. In practice, each provider alone decides which part of his incoming requests to outsource. In a non-cooperative environment, each provider makes decisions aiming to maximize his own benefit, i.e. to minimize his own customers’ average delay.

Taking as example the simple case of **one-way federation**, it’s easy to realize that requests outsourcing brings benefits to \(P_1\) with respect to his customers’ average delay. On the other hand, this action is not beneficial to \(P_2\), since the average delay of his customers will be increased. Note that, in a non-cooperative environment, \(P_1\) will not solve the problem of total system’s average delay minimization, but he will only try to minimize
his own customers’ average delay. Next, we define the problem of \( P_1 \) individual-optimization in the one-way federation:

\[
\min_{a} d_1(a, d) \\
\text{s.t. } 0 \leq a \leq 1 \\
0 \leq d
\]

If we define its optimal value by \( a_1^*(d) \), then it is intuitive that \( a_1^*(d) \geq a^*(d) \) for every \( d \). This is also verified by Figure 4-7.

![Figure 4-7: Global optimal (blue line) vs. \( P_1 \) optimal (red line) \((a^* \text{ in } d, \lambda_1=2\lambda_2 \text{ and } \mu_1=\mu_2)\).](image)

Assuming that the value of \( \alpha \) is defined by \( P_1 \), the average delay of \( P_2 \)'s customers in one-way federation is given by the function \( d_2(\alpha) \). Consequently, the customers of \( P_2 \) have the lowest delay when \( P_1 \) does not outsource any request and is equal to \( d_2(0) \). On the other hand, the extra delay for \( P_2 \)'s customers created due to additional workload, is equal to \( d_1(\alpha^*) - d_2(0) > 0 \). Since there is no gain for \( P_2 \), \( P_1 \) has to pay compensation to \( P_2 \) for the extra delay adds to his own customers. Thus, we have to define a reward function in order to decide how much \( P_1 \) should pay to \( P_2 \) for serving a certain part of request.

So far, we have assumed that the metric to be optimized is the average delay offered either by a single provider or by the federation as a whole. In a more general modelling framework we can introduce functions for monetary earnings and costs. In particular, a reasonable way to model the revenue per time unit of \( P_i \) is \( \lambda_i \cdot p_i(d_i) \), where \( p_i(d_i) \) is the price per job charged by Provider \( P_i \), and it is to be taken as a decreasing function of \( d_i \) i.e. of the average QoS level offered by \( P_i \). Also the per time unit total costs of \( P_i \) (infrastructure, energy etc.) are modelled as \( c_i(\lambda_i) \) which is an increasing function of the rate of jobs \( \lambda_i \) actually served in the queue of \( P_i \). Therefore, for some \( \alpha > 0 \), we can calculate the benefit of \( P_1 \) for choosing this value of \( \alpha > 0 \), and compare it with the loss incurred to \( P_2 \). If the former is larger than the latter, then \( P_1 \) can compensate \( P_2 \) and still benefit by selecting this value of \( \alpha > 0 \). Under this modelling approach, the optimal choice of \( \alpha \) is the one maximizing the net profit of \( P_1 \) after the compensation of \( P_2 \).
The presented model is in progress and the investigation of incentives for cloud federation is ongoing; thus, numerical results for pricing are not available at this point of time. By means of this model, we aim to design a mechanism that will achieve load balancing between cloud providers for either performance and/or profit optimization. As next steps, we aim to define revenue and cost functions for the providers. To define these functions we will employ the payoff obtained from end-users for offering a service and the infrastructure costs for serving the requests. Also, we aim to investigate the formation of a game among cloud providers that would drive to an equilibrium point close to optimal resource allocation. Finally, we will consider a dynamic approach, where the decision for outsourcing will be based on the current size of queue.

4.2 Investigation of content demand in OSNs

The investigation of individual content demand is especially important in the context of the End-user-Focused Scenario. Traffic management solutions like RB-HORST and SEConD improve content delivery by employing caching and prefetching.

In order to evaluate the performance of the developed mechanisms, a content demand model which accurately reflects content requests of real users is required. Such a model, which also takes temporal dynamics into account, is presented in section 4.2.1.

For the prefetching of content in RB-HORST, an algorithm is needed which predicts future content requests of individual users. Thus, a socially-aware prefetching algorithm was developed which is described in Section 4.2.2.

4.2.1 Content demand model with temporal dynamics

The popularity distribution of content in the Internet according to requests is usually approximated with a Zipf distribution with exponents $\alpha$ from 0.6 to 0.99 [80]. Recent studies show, that content demand is non-stationary and shows temporal locality [82], and that the temporal dynamics in the content request patterns are crucial for the performance of common caching strategies. To accurately investigate the performance of mechanisms like RB-HORST appropriate content demand models with temporal dynamics are necessary. In literature are many studies that describe the dynamics of content demand based on measurements [83].

However, study [81] shows that the detailed temporal dynamics are not necessary. It is sufficient to specify the life span and the popularity of content items to assess the cache efficiency accurately. Thus, we develop a model with content life span and popularity as parameters. The correlation between content life span and its popularity can further be specified by a joint distribution of life span and popularity.

Hence, the goal of the content demand model with temporal dynamics is to generate requests with specified life span and popularity that exhibit a Zipf law. Moreover, the developed model is in place to generate flash crowd events so as to stress TM mechanisms during evaluations.

Table 4-1: Parameters of content demand model with temporal dynamics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Zipf-exponent for content popularity</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Content upload rate</td>
</tr>
<tr>
<td>$\tau_i$</td>
<td>Content life span</td>
</tr>
</tbody>
</table>
Table 4-1 shows the parameters of the content demand model with temporal dynamics. The Zipf-exponent $\alpha$ specifies the slope of the content popularity power-law. Hence, it also specifies the content request rate $\lambda_i$ of each content item $i$. The content upload rate $\Lambda$ specifies the rate of new content items being uploaded. As soon as a content item $i$ is uploaded it is active in the system and can be requested. After a certain time more recent content items get more interesting and the content item is not consumed anymore and leaves the system, because it is not active anymore. The time lag a content item $i$ is active, i.e. between its upload until the point it leaves the system, is the content life span $\tau_i$.

To generate requests we start by determining the number of requests $n_i$ of each content item $i$ according to the Zipf distribution with exponent $\alpha$, given a total number of requests $N$, such that $\sum_i n_i = N$.

For each content item $i$ the content life span $\tau_i$ is determined by a lognormal distribution. We choose a lognormal distribution, since it produces positive values and its mean value and standard deviation can be parameterized.

The content upload arrival process can be modeled by a Poisson process, hence the time between two subsequent content upload times $u_{i,j}$ of items $i$ and $j$ is determined by an negative exponential distribution with rate $\Lambda$.

The main outcome of [81] is that given life span $\tau_i$ and request rate $\lambda_i$ the exact request arrival process distribution within interval $\tau_i$ is not necessary, but can be modeled with a poisson process without affecting the cache performance. Hence, the request arrival process of an item $i$ within its life span $\tau_i$ can be modeled by a Poisson process with rate $\lambda_i$.

Finally we obtain the request times $t_i^k, 1 < k < n_i$ of item $i$ by adding $n_i$ negative exponential distributed inter-arrival times $\theta_i^a, 1 < k < n_i$ with rate $\lambda_i = \frac{\tau_i}{n_i}$ to the upload time $u_i$. Hence, the time $t_i^k$ of the $k$th request of item $i$ can be calculated by:

$$t_i^k = u_i + \sum_{a=1}^{k} \theta_i^a, \quad 1 < k < n_i,$$

where $u_i$ follow a negative exponential distribution with rate $\Lambda$, and $\theta_i^a$ follow a negative exponential distribution with rate $\lambda_i = \frac{\tau_i}{n_i}$.

Next steps are to determine proper parameters for the lognormal distribution of the life span and a model for the joint distribution of life span and popularity. This model also supports flash crowd scenarios, by specifying a high content request rate $\lambda_i$ paired with a low life span $\tau_i$.

### 4.2.2 Model for prediction of individual content requests based on social network traces

A socially-aware pre-fetching algorithm for individual end users was developed. This algorithm can be used in systems, which prefetch content to end user devices, like RB-HORST. The algorithm was developed and validated based on different Twitter datasets.

The first step was to download external datasets and, with the help of crawlers, data for own measurements were collected. The problem with these datasets is that they do not contain the information, whether a user requested and consumed a content item.
Therefore, the assumption was made, that a user has at least consumed the content, before he retweets it. That means the number of retweets of the content is a lower bound of the number of content requests. In this work, the content was limited to YouTube videos.

The second step was the detection of the sessions for each user in the datasets. Then, the news feed for each session for each user was reconstructed with the wall algorithm. Based on the news feeds, the collection algorithm collected features of the tweet (like the age of the content, how many followers have posted the tweet, or how often did the user watch the video), and investigated if the tweet was retweeted or not.

For predicting, the algorithm calculates a weighted score out of five important dimensions, age, distance, history, popularity, and social. Each dimension was modeled by a mathematical function, which was best fitted to the observed retweet decisions. Due to the high noise in the data, binning was applied before the fitting. However, still the goodness of fit does not reach high values (see Figure 4-8).

Support vector machines (SVMs) are used to assess the influence of a variable. In [10], it is suggested that the squared weight can be used as an influence ranking criterion. Table 4-2 shows the best fitted function, the goodness of the fit, and the rank of importance according to the SVM weights for each dimension.

Table 4-2: Functions for the considered dimensions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Function</th>
<th>R-squared</th>
<th>Importance (rank)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age of video (time α since first upload in weeks)</td>
<td>$f_{\text{age}} : [0, \infty] \rightarrow [0, 1]$, is defined by $f_{\text{age}}(\alpha) = 4.898 \cdot 10^{-4} \cdot e^{-2.951\alpha} - 2.961 \cdot 10^{-7} \cdot \alpha + 1.111 \cdot 10^{-4}$</td>
<td>0.24</td>
<td>3</td>
</tr>
</tbody>
</table>
After each dimension was modeled, the algorithm can be designed. For predicting whether a video will be retweeted, a score, which is built as weighted sum of the five functions presented in Table 4-2, is an indicator. The value of each weight and the intercept $\lambda_6$ is calculated by logistic regression. Therefore, the dataset was split into two subsets, a test set and a training set. The training set consisted of 2,000 retweets and 2,000 tweets, which were not retweeted. On the other hand, the test set consisted of all other tweets (about 1,500 retweets and about 12.8 million tweets, which were not retweeted). The resulting weights, after the logistic regression model was trained, were:

$$\text{score} = \lambda_1 \cdot f_{age} + \lambda_2 \cdot f_{distance} + \lambda_3 \cdot f_{history} + \lambda_4 \cdot f_{popularity} + \lambda_5 \cdot f_{social} + \lambda_6$$

With $\lambda = \begin{pmatrix} -3646.8405 \\ -348.238 \\ -174.6666 \\ 32.539 \\ -1.0468 \\ 1.8463 \end{pmatrix}$

Then, this score is transformed to a probability $p_{\text{retweet}}$ for a retweet. If this probability $\geq 0.5$, the video is predicted to be watched, otherwise it is not watched.

$$6\% \leq p_{\text{retweet}} = \frac{1}{1 + e^{\text{score}}} \leq 100\%$$

The test set was used to evaluate the model, which reached an accuracy of 98.34%. The precision of a retweet, i.e., the fraction of correctly classified retweets and classified retweets, is 0.005. That means, from all tweets, which were classified as a retweet, only 0.5% of them were really retweets. In contrast, the precision of tweets, which were not retweeted, is almost 1. That means, almost all tweets, which were not retweeted, are correctly classified. Furthermore, about 81% of all retweets were detected, as indicated by the recall of retweets, i.e., the ratio of correctly classified retweets and all retweets. About 98% of tweets, which were not retweeted, were detected.

In a last step, the algorithm was compared to other machine learning approaches. For this purpose, two decision trees were created. The best one with only three dimensions achieves an accuracy of 96.59%, but has less effort than the presented algorithm. Moreover,
logistic regression was used with all collected tweet features, which include category, length, language, etc. This more complex approach reached an accuracy of 98.54%.

The outcome of this work is that all investigated models achieve a high accuracy. That means, a prefetching system applied to the used dataset would prefetch only 268,000 videos out of 12.8 million possible videos, which is only 2% of video content. But the recall of retweet has a very bad value of 0.005. That means, from the 268,000 prefetched videos only 1,500 videos are really needed. Furthermore, the precision of a retweet is 0.8, which means that 20% of the consumed content is not prefetched by the algorithm. However, it must be noted that the model is based on retweets and not on views. If views were available as input to the model, the precision and recall measures would likely improve.

To sum up, the presented algorithm is very promising and shows the possibility to predict a single user's content requests. Although it achieves high prediction accuracy there are too many wrong predicted consumptions due to the retweet limitation of the dataset. In the future, the algorithm should nevertheless be applied to a dataset or a real-world system, in which it is possible to monitor the content consumptions of end users. The presented approach is very likely to deliver a good performance then.

### 4.3 Investigation of Topological Characteristics of the Internet

Topological characteristics of the Internet are of special interest for the RB-HORST traffic management solutions, which creates a content delivery overlay consisting of home routers. In order to quantify the savings in terms of inter-AS traffic and the ISP cache contribution, a model of the distribution of home routers on ASes is needed. Such a model is elaborated in Section 4.3.1.

Additionally, RB-HORST also offers WiFi offloading to visiting trusted users. Section 4.3.2 presents a model for the distribution of open WiFi hotspots in cities to which the RB-HORST approach can be compared.

#### 4.3.1 Model for Global Distribution of Home Routers on ASes

In order to evaluate the performance of an approach that utilizes home routers to save inter-domain traffic or to estimate the potential of home router sharing approaches as proposed in [11] for ISPs, the number of home routers available in an ISPs autonomous system (AS) is needed. We develop a model for the global distribution of home routers on ASes by evaluating the Internet Census dataset. The Internet Census dataset [12] was collected from June to October 2012. The complete IPv4-address space was scanned by a botnet with different methods, like ICMP-Ping, Reverse-DNS, NMAP, Service Probes and Traceroute. The botnet consisting of 420,000 nodes is called "Carna Botnet". The Internet Census dataset is forensically analyzed and validated in [13].

The ICMP-Ping scan of the Internet Census dataset is used to identify all assigned and active IP-addresses in the Internet. Thus 1.8TB of data with 52 Billion records are analysed on valid and alive IPv4-addresses. The resulting dataset contains all identified active IPv4 addresses. To obtain the distribution of IP-addresses on ASes the IP-addresses have to be mapped to AS-numbers. For this purpose the MAXMIND GeoLite ASN database is used, which is available at [14]. The GeoLite ASN database is updated every month and can be easily accessed by APIs provided in different programming languages.

Figure 4-9 shows the number of alive IPv4-addresses for each AS in the 16bit autonomous system address room. The IPv4-addresses alive in an AS reach from one address to up to almost 10^8 active IPv4-addresses. That means that almost all 16bit AS-
addresses are assigned and active. The majority of autonomous systems have less than 10000 active IP-addresses. Hence, there are only few very large autonomous systems, but many smaller autonomous systems. Since the 16bit AS-address room is almost completely assigned, the AS-addressing has been expanded to 32bit.

Figure 4-10 shows the number of active IPv4-addresses for each AS in the 32bit autonomous system address room. At the low AS numbers the highly assigned 16bit AS address block can be identified. There are a few further peaks in the 32bit AS address room. Most IP-addresses are active in the additional peak at 20000. These autonomous system numbers are managed by the RIPE NCC regional Internet registry which provides Internet resource allocations, registration services and coordination activities around Europe. If the autonomous systems are classified into small ASes with less than 10000 active IPs and large ASes with more than 10000 active IPs, 98.81% of the autonomous systems are small ASes. However, the small ASes only cover 13.01% of the active IPs, whereas the remaining 86.99% of active IPs are in large autonomous systems.

![Figure 4-9: 16bit AS address room.](image)

![Figure 4-10: 32bit AS address room.](image)

This suggests that the number of active IPs in autonomous systems follows a power-law. Figure 4-11 shows the Zipf-law of the number of active IP-addresses in autonomous systems. The number of active IP-addresses is plotted in descending order. The Zipf-rank can be approximated by a power-law with slope 1.5 at its body. Hence, there are few autonomous systems with many IP-addresses and many autonomous systems with few IP-addresses. As stated before the majority of active IP-addresses is in a small number of very large autonomous systems. Table 4-3 shows the top five autonomous systems with most active IPs. The two large autonomous systems in terms of active IPs are operated in China. This depends on the high population of China. Comcast is a large telecommunication company in the US which acquired AT&T Broadband in 2002.

![Image](image)
The probability that an autonomous system has a number of active IP-addresses is shown in Figure 4-12. The probability distribution follows a power-law with slope 0.9970. The probability distribution can be used to generate the number of active IPs in a synthetic autonomous topology.

The goal of this section is to develop the distribution of home routers on autonomous systems. To obtain the number of home routers in an autonomous system, the amount of active IPs which belongs to home routers has to be determined. The sum of all active IP-addresses identified in the ICMP Ping dataset is 598 million. The number of active web servers and printers can be estimated by analysing the Service Probe dataset. The Service Probe dataset yields 71 million IP-addresses that listen to port 80, i.e. web servers, and it yields a quarter million IP-addresses that listen to port 9100, i.e. printers. Assuming that other network functions do not respond to ICMP Ping requests, as firewall do by default, the number of active home routers can roughly be estimated by 598 million less 71 million equals 527 million active home router. Hence, this would mean, that about 88.1% of active IPs are home routers.
According to these results a model for the distribution of active IP-addresses and home routers on autonomous systems is defined. The model is used for performance evaluation of traffic management mechanisms that integrate user equipment and home router. In future work the autonomous systems could be further classified to identify interesting relations between the number of active IPs and the type of ISP.

### 4.3.2 Model for geographic distribution of WiFi hotspots

The database OpenWiFiSpots was used to gather the addresses of public WiFi hotspots of the sample cities, and the database OpenStreetMap was used to gain geographic information about the hotspot addresses and the cities’ road networks (cf. Figure 4-13).
The analysis framework was used to evaluate the public WiFi hotspots of ten example cities both in Europe and the United States. The analysis of the probability distributions led to a basic model with exponential distribution of the hotspots’ distance from the city center and uniformly distributed angles. With this basic model an approximate hotspot distribution to any city can be modeled.

As the Cartesian coordinates (latitude/longitude) have inherent complex dependencies, the hotspot distribution with respect to the city center is analyzed by using polar coordinates in the following. In this context, the hotspot distribution in term of the distance to the city center of the ten example cities is presented. The one-dimensional distributions in term of the distance are fitted by an exponential function:

\[ f(x) = a \cdot \exp(b \cdot x) \]

The ten fitted curves are shown in Figure 4-14. The x-axis shows the distance of the hotspots with respect to the city center. It is measured in kilometers. The y-axis shows the number of hotspots per square kilometer of the respective area of circle. The figure represents each of the cities’ hotspot distribution fitted by the exponential function.

Table 4-4 provides the corresponding R-squared values, indicating that the exponential curve fits the observed data very well. This also approves the assumption of a higher density of hotspots at the city center compared to the borderlands. It can be seen that most curves have a nearly similar slope with two exceptions. The curve representing Berlin shows a distinctly minor slope, whereas the curve representing San Francisco shows a greater slope than the remaining curves. By investigating the exponential functions parameter no dependencies could be deduced and such analysis is open to future work.

![Figure 4-14: Distribution of Hotspots with Respect to the Distance to the City Center.](image)

### Table 4-4: Goodness of Exponential Fits of the Distance Distribution.

<table>
<thead>
<tr>
<th>City</th>
<th>R-square</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>97.37%</td>
</tr>
<tr>
<td>Berlin</td>
<td>99.60%</td>
</tr>
<tr>
<td>Boston</td>
<td>99.33%</td>
</tr>
<tr>
<td>Brooklyn</td>
<td>99.34%</td>
</tr>
<tr>
<td>Houston</td>
<td>99.12%</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>98.77%</td>
</tr>
<tr>
<td>London</td>
<td>98.72%</td>
</tr>
</tbody>
</table>
Figure 4-14 presents the hotspot angular distribution of London. The x-axis shows the various angles from −180 to +180 degree in steps of one degree per angle. The y-axis represents the number of hotspots at the respective angle. Therefore, the hotspot angles are reprocessed using the four quadrants of the coordinate system.

The hotspots in the first quadrant of the coordinate system with respect to the city center have angles between zero and +90 degree, the hotspots in the second quadrant are related to the angles between +90 and +180 degree. The hotspots located in the third quadrant are represented by the angles between −180 and −90 degree, and the hotspots in the fourth quadrant are related to the angles between −90 and zero degree.

The angular distribution is similar for each of the ten cities and shows a high similarity to a uniform distribution with minor deviations due to geographic conditions like water areas or parks, which led to free space at the corresponding angles. Furthermore, no differences between the European and US cities can be recognized.

Now, a simple model of the hotspot distribution is established by using the probability distribution of the polar coordinates, which are comprised of the distance to the city center and angle. In order to create a hotspot distribution two random numbers \( u, v \) have to be drawn uniformly from the unit interval \((0,1)\). Moreover, the parameter \( b \) of the exponentially distributed distance is required as input, and an additional scaling factor \( c = a/b \). The random distance \( d \) to the center can be obtained by applying an inverse transformation method to the first random number \( u \). The distance, which is exponentially distributed with parameter \( \lambda \), can then be obtained by:

\[
\text{Random Distance} \quad d = c \cdot -\log(u)/\lambda,
\]

while the corresponding angle \( a \) can be computed from the second random number \( v \):

\[
\text{Random Angle} \quad a = 360 \cdot (v - 0.5).
\]

The generated hotspot distribution will then follow the characteristics observed in the ten example cities.
4.4 Quantification of power consumption

The Information and Communication Technology (ICT) currently uses the same amount of electrical energy as the electrical energy generated in Japan and Germany combined [84], which is likely to rise in the following years. Hence, one of the main objectives of SmartenIT is the analysis and improvement of the energy consumption of the ICT in different aspects.

To reduce the power draw of individual components, detailed knowledge of the operating point and the power draw over a wide range of operating modes is required. As also traffic management on the end-user premises is to be conducted by the traffic management mechanisms like RB-Horst and vINCENT, a home gateway with extended capabilities was required. The Raspberry Pi was selected, because it is cheap, widely used and demonstrates low power consumption. To allow optimizing the power consumption, a power model for the Raspberry Pi is measured and calibrated, which allows determining the optimal operating state and intelligent scheduling of tasks.

The subsequent section details the cost calculation of mobile data transmissions for mobile phones, which are ubiquitous, and are predicted to cause the majority of end-user generated traffic in the future (cf. traffic studies in [2]). This allows estimating the transmission cost depending on the available network quality before starting a connection. This is important, as the per-bit cost of wireless interfaces assimilate under certain conditions, and such the cellular interface might be more energy efficient. Knowing the network quality at different times and locations such allows intelligent scheduling of data transfers in an energy-optimal way. The information derived from the network state is used by the Mobile Network Assistant (MoNA, cf. Section 6.4).

4.4.1 Power model of the Raspberry Pi

The Raspberry Pi is to be used by RB-Horst and vINCENT within the project. To quantify the energy efficiency gains of this approach, a comparison of the power consumption of the mobile content delivery with and without the use of uNaDas is to be performed. As physical power measurements are laborious, and difficult to execute for multiple devices at different locations, a power model for the Raspberry Pi is developed.

The following sections, we present PowerPi, a power model focusing on the power consumption of the Raspberry Pi to derive possible power saving strategies. The hypotheses of our work are:

- An accurate estimation of the Raspberry Pi’s power consumption can be obtained using the system utilization only.
- Knowing the power model, it is possible to reduce the power consumption by optimizing routing and caching decisions.

PowerPi is based on hardware measurements of the Raspberry Pi. After the description of the measurement setup and accuracy, the measurements are described. Then, the resulting power model is derived, and an upper bound of the error to be expected when applying the power model is given.

The main focus of related work currently is on analyzing and improving the energy efficiency of either high-performance networking, or mobile devices or network interfaces only. To the best of our knowledge, no power model for the Raspberry Pi is available, neither is there one for one of the other low-cost low-power ARM boards.
The power consumption of low power high efficiency processors (ARM, Intel ATOM) is compared to conventional high performance processors in [18]. Results of different benchmarks normalized by the power consumption of the processors are compared, concluding that, depending on the benchmark, the energy efficient variants of conventional desktop (i7) or server processors (Xeon E7) outperform ARM processors for most benchmarks in the number of achieved points per watt. As currently servers are usually operated in a CPU range of 10% to 50% [15], the power consumption while idle should also be included in the benchmarks.

In the following sections, the measurement and modeling of the power consumption of the Raspberry Pi is described.

**Measurement Setup**

The power consumption of the Raspberry Pi is measured using an external power meter. Simultaneously, scripts and custom tools on the platform generate load and monitor the device state. The following sections describe the measurement setup in detail.

![Diagram of measurement setup](image)

**Power Measurement**

The Raspberry Pi is a low-power device, which supports being powered via USB. Its power consumption is measured by interrupting the power lines of the USB connection and inserting a measurement shunt in the 5 V line. The wiring of the setup, as shown in Figure 4-16a, is detailed in Figure 4-16b. The current flowing through R1 causes a voltage U1, proportional to the current drawn by the Raspberry Pi.

Custom built software based on Measurement Computing’s FlexDAQ API constantly measures the voltage drop U1 over this shunt and the voltage U2 of the 5 V line. The power consumption of the Raspberry Pi is calculated with

\[ P_{Pi} = \frac{U_1 \cdot U_2}{R_1} \]

and writes the measurements together with a time-stamp to a local file.

The final error of the power measurement depends on the accuracy of the two voltage measurements, which depend on the accuracy of the A/D conversion and the accuracy of the measurement shunt. Mathematically, the maximum error is defined as

\[ \max \left( \frac{\Delta P}{P} \right) = \max \left( \sqrt{\left( \frac{\Delta I}{I} \right)^2 + \left( \frac{\Delta U}{U} \right)^2} \right) \]
Filling in parameters of the A/D converter and the measurement setup, the error sums up to 2.47%. The detailed measurement setup and system state monitoring, and load generation are detailed in [19].

**Measurements**

The influence of the system and network load on the power consumption of the platform is evaluated by generating CPU and network load, while simultaneously measuring the power consumption. For this, the system monitoring scripts described in [19] have been used. The resulting measurements and power models are given in the next paragraphs.

The collected measurements are plotted in heat maps to allow a visualization of the density of the measurements. This is advantageous compared to scatter plots, as the high number of measurements reduces the visibility of the individual data points. The heat map is logarithmically weighted to visualize the full range of measurements. On top of the heat map, the models derived below are plotted:

**CPU:** Figure 4-17 shows the result of the CPU measurements with CPU utilization in the range of 10% to 100% in 10% steps and a fitted linear function generated by Matlab’s robustfit function. The horizontal extent of the data spots reflects the accuracy of the CPU-limiting script.

![Figure 4-17: Power Consumption vs. CPU utilization.](image)

**Ethernet:** Figure 4-18 shows in the left graph the power measurements of the download experiments over the link utilization as a heat-map. From the raw measurements, the power consumption generated by the idle state and CPU is subtracted. Remaining is the power of the network transmission only. This is possible, as during the measurements, the network load, and the CPU utilization was monitored by the respective scripts. The same configuration, but uploads from the platform are measured in the right plot. The maximum increase over the idle power is lower than 40 mW for a fully utilized Ethernet interface while downloading is 96 mW at a rate between 30 Mb/s and 40 Mb/s. The lowest power consumption while uploading data is 21 mW lower than the power consumption of the idle interface, which is also visible in other measurements of consumer grade network equipment [16].
Figure 4-18: Ethernet power consumption during download (left) and upload (right) vs. used bandwidth

WiFi: Figure 4-19 (left) shows the power consumption when receiving data over WiFi. Contrary to the Ethernet measurements, a distinct increase in power consumption based over the downlink bandwidth is visible. For higher data rates (>85 Mb/s) two distinct power states are visible. The power consumption of the WiFi interface during upload is plotted in the right graph. For higher rates (>40 Mb/s) the measurements begin to spread. This is thought to be caused by the higher computational effort required to create the frames and coordinate the connection. This is also reflected in the power consumption, which is more than 700 mW higher at the upper end.

Figure 4-19: WiFi power consumption during download (left) and upload (right) vs. used bandwidth.

Model Generation

The power model for the different measurements is generated by fitting a linear function to the measured data, minimizing the remaining Root Mean Square Error (RMSE). For this purpose, Matlab’s robustfit function is used. This function is based on an iterative process fitting the linear function to the data, minimizing the RMSE. The detailed process is described in [17]. The underlying data is weighted with a bi-square function to reduce the effect of outliers on the final fit.
The measurements of the CPU utilization in Figure 4-17 show a clear linear dependency. This observation is confirmed, by calculating the 1st order regression. The resulting function for the platform including CPU utilization is

$$P_{\text{Pi, CPU}}(u) = 1.5778W + 0.181 \cdot u \cdot W.$$  

The power models with the best fit are plotted in Figure 4-18 and Figure 4-19. A detailed interpretation of the measurement results can be found in [19].

Table 4-5 shows the approximations of the power consumption of the Raspberry Pi for different utilization. The first column is the symbol used in the text to refer to this function. The second column indicates the approximation order, while the third column gives the RMSE, which is the mean error to be expected when using the model. The last column lists the formula describing the dependency between utilization and power consumption.

From this an additive model can be derived, where the absolute power consumption of the Raspberry Pi can be modeled on a per-component basis. The model can be expressed as

$$P_{\text{Pi}} = P_{\text{idle}} + P_{\text{CPU}}(u) + \sum_{i} (P_{\text{if, idle}} + P_{\text{if, up}}(r) + P_{\text{if, down}}(r)),$$

where the constants $P_{\text{idle}}$ and $P_{\text{if, idle}}$ and the approximations $P_{\text{CPU}}(u)$ and $P_{\text{if, up}}(r)$ are defined in Table 4-5. Here, the interface if is either WiFi or Ethernet. The RMSEs of the built in components are generally quite low (<18 mW). Only the WiFi measurement shows a larger error. This might be caused by a higher variance of the power consumption of the USB dongle, but might also be caused by the generally higher power consumption compared to the Raspberry Pi itself. A power consumption of 2 W for the WiFi connection is double the platform’s power consumption, and close to the maximum allowed power draw of a USB 2.0 device of 2.5W.

Table 4-5: The power models of the Raspberry Pi. Here, $u$ is the CPU utilization in the range 0 to 1 and $r$ the traffic rate in Mb/s.

<table>
<thead>
<tr>
<th>Function</th>
<th>Ord.</th>
<th>RMSE</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\text{idle}}(u)$</td>
<td>0</td>
<td>15 mW</td>
<td>1.5778 W</td>
</tr>
<tr>
<td>$P_{\text{Eth, idle}}(r)$</td>
<td>0</td>
<td>9 mW</td>
<td>0.294 W</td>
</tr>
<tr>
<td>$P_{\text{wifi, idle}}(r)$</td>
<td>0</td>
<td>8 mW</td>
<td>0.942 W</td>
</tr>
<tr>
<td>$P_{\text{CPU}}(u)$</td>
<td>1</td>
<td>18 mW</td>
<td>0.181W · u</td>
</tr>
<tr>
<td>$P_{\text{Eth, up}}(u)$</td>
<td>1</td>
<td>16 mW</td>
<td>(0.006W + 1.606e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}})</td>
</tr>
<tr>
<td>$P_{\text{Eth, down}}(r)$</td>
<td>2</td>
<td>17 mW</td>
<td>(0.003W + 1.634e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}} - 6.531e^{-6} \cdot r^2 \cdot \frac{W}{\text{Mb/s}^2})</td>
</tr>
<tr>
<td>$P_{\text{wifi, up}}(u)$</td>
<td>3</td>
<td>16 mW</td>
<td>(-0.002W + 2.702e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}} - 3.838e^{-6} \cdot r^2 - 2.331e^{-7} \cdot r^3 \cdot \frac{W}{\text{Mb/s}^2})</td>
</tr>
<tr>
<td>$P_{\text{wifi, down}}(r)$</td>
<td>4</td>
<td>14 mW</td>
<td>(-0.008W + 4.702e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}} - 0.164e^{-6} \cdot r^2 - 2.509e^{-6} \cdot r^3 - 12.408e^{-9} \cdot r^4 \cdot \frac{W}{\text{Mb/s}^4})</td>
</tr>
<tr>
<td>$P_{\text{WiFi, up}}(r)$</td>
<td>1</td>
<td>11 mW</td>
<td>(0.000W + 2.327e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}})</td>
</tr>
<tr>
<td>$P_{\text{WiFi, down}}(r)$</td>
<td>2</td>
<td>8 mW</td>
<td>(-0.002W + 5.542e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}} - 45.850e^{-6} \cdot r^2 \cdot \frac{W}{\text{Mb/s}^2})</td>
</tr>
<tr>
<td>$P_{\text{WiFi, up}}(r)$</td>
<td>1</td>
<td>59 mW</td>
<td>(0.037W + 4.813e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}})</td>
</tr>
<tr>
<td>$P_{\text{WiFi, down}}(r)$</td>
<td>2</td>
<td>26 mW</td>
<td>(0.010W + 11.003e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}} - 71.988e^{-6} \cdot r^2 \cdot \frac{W}{\text{Mb/s}^2})</td>
</tr>
<tr>
<td>$P_{\text{WiFi, up}}(r)$</td>
<td>1</td>
<td>83 mW</td>
<td>(0.064W + 4.813e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}})</td>
</tr>
<tr>
<td>$P_{\text{WiFi, down}}(r)$</td>
<td>2</td>
<td>71 mW</td>
<td>(0.020W + 24.387e^{-3} \cdot r \cdot \frac{W}{\text{Mb/s}} - 112e^{-6} \cdot r^2 \cdot \frac{W}{\text{Mb/s}^2})</td>
</tr>
</tbody>
</table>

Summarizing, in this section, we have presented PowerPi, a power model for the Raspberry Pi, which includes the CPU and Ethernet power consumption as well as the
power consumption of an external USB WiFi dongle. The power model is modular to incorporate all measured components.

The error of the model has been evaluated based on the measurement accuracy and the error introduced by the model. The resulting errors for the built-in components are in the order of tens of mW only. PowerPi can be used to improve the energy footprint of software running on the Raspberry Pi, using system traces only. Similar power models can easily be generated for other devices by repeating the same measurements. These power models are to be used within the project to assess the energy efficiency of the implemented approaches, and to improve the performance based on a feedback loop.

Using the PowerPi model, and similar models of other devices connected to the network like smartphones [20], routers [21], and servers [22], it becomes possible to estimate the power consumption of the full network for a given load. Hence, energy efficiency improvements of the full network infrastructure are possible based on the power models and the system utilization only. This eliminates the need for dedicated power measurement hardware, but allows deriving the power consumption with an accuracy of lower than 3.3%.

Energy efficiency gains can be achieved by exploiting nonlinear behavior of the individual components, and adjusting the workload to the optimal operating point, by e.g. scheduling traffic and/or switching off components.

For a networked system specialized on traffic forwarding, the traffic statistics alone are sufficient to generate accurate predictions of the network state and power consumption. This allows optimizing the traffic flows to use the most energy-efficient paths, or redirect computations to the most energy-efficient location based on the current utilization. Thus, both hypotheses are supported by this section.

### 4.4.2 General Power Model of Wireless Data Transmissions

The power consumption of a wireless data transmission depends on the power consumption of the active interface, the available data rate, and the status of the interface. The simplest case is the power consumption of a data transfer of a single connection, as described in the next section. This model is then extended to estimate the power consumption of multiple, parallel transfers, allowing to assign an energy cost to each request.

**Power Consumption of a Single Connection**

The power for each connection depends on the available networks and their parameters like idle power, timeouts, and the cost per byte sent/received. Generally, the power for a connection is calculated by

\[ E_c = E_t + E_i, \]

where \( E_t \) is the transmission energy and \( E_i \) is the initial energy required to open the connection, defined as

\[ E_i = E_r + E_e. \]

Here, \( E_r \) is the ramp energy required to enter the transmission state and \( E_e \) is the tail energy, used in the high power state after the last packet of the connection was transmitted. \( E_r \) and \( E_e \) are connection dependent. Influencing factors are the interface type (i.e. WiFi/cell), the used hardware and timing parameters set by the network operator.
The energy required for the transmission of the flow $E_t$ is the integral over the power consumption $P_c(t)$ of the interface over time:

$$E_t = \int_{t_0}^{t_0+T_t} P_c(t) \, dt.$$  

The transmission duration $T_t$ depends on the available data rate $R_i(t)$ of the interface at the time $t$. Using the (estimated) size of the connection in upload $S_{c,u}$ and download direction $S_{c,d}$, the transmission duration $T_t$ is defined as

$$T_t = \max \left( \frac{S_{c,u}}{R_u(t)} , \frac{S_{c,d}}{R_d(t)} \right).$$

This calculation is to be executed for upload and download directions. The power consumption of the interface itself consists of 3 distinct parts:

$$P_c(t) = P_a + P_u(R_u(t)) + P_d(R_d(t)).$$

These are the cost of the interface being active $P_a$, the upload power $P_u(R_u(t))$, and the download power $P_d(R_d(t))$. The instantaneous power consumption of an interface can be derived from models published in literature, like [20] or measured directly. Integrating this over the available data rate at the time $t$ results in the overall power consumption of the transmission part. Adding the energies required for opening and closing the connection results in the full cost of a single transmission on a single interface.

**Power Consumption of Simultaneous Connections**

Considering real traffic conditions, multiple connections may use the interface simultaneously, in particular if connections are scheduled for later or concurrent transmission (e.g. a connection of a background service is deferred until the connection is opened by a foreground request). Hence, the overall power consumption for $N$ simultaneous connections of the interface, excluding the ramp and tail energies is:

$$P(N, t) = \sum_{i=0}^{N} P_c(i, t, N), \quad s.t.: P_c < P_{max},$$

where $i$ is the index denoting the connection, and $P_c(i, t, N)$ is defined similarly to above as

$$P_c(i, t, N) = \frac{P_d}{N} + P_{i,u} \left( R_{i,u}(t) \right) + P_{i,d} \left( R_{i,d}(t) \right).$$

Here, $P_{i,u}$ is the power consumption of the connection $i$ in upload, $P_{i,d}$ the power consumption in download direction, and the data rate for each direction $R_{i,u}(t)$ and $R_{i,d}(t)$ are defined as the minimum of the requested data rate and the maximum available data rate $R_{max,i}$ and $R_{max, d}$, shared by the number of active connections $N_{act,u}(t)$ and $N_{act,d}(t)$

$$R_{i,u}(t) = \min \left( \frac{R_{max,u}}{N_{act,u}(t)}, R_{i,u}(t) \right), R_{i,d}(t) = \min \left( \frac{R_{max,d}}{N_{act,d}(t)}, R_{i,d}(t) \right).$$

To calculate this, it is necessary to know the number of active transmissions at all times. Similarly, the cost of the connection establishment and tear down is split between the active connections:

$$E_i = \frac{(E_r + E_t)}{N}$$

Considering fair bandwidth allocation, the power per connection can be simplified as
where $\bar{R}_{i,u}$ and $\bar{R}_{d,u}$ are the average data rates for all connections on the particular interface. Modelling this is only possible, if all connections are established at exactly the same time.

As in the case of a traffic scheduler on the mobile device, connections may have different priorities. Considering the power model for the individual connections derived above, and combining it with priorities, a number of changes must be made. As only a single request opens the connection, it needs to burden the full cost of the connection establishment and tear down. This, in contrast, relieves the scheduled connections of this cost, if they are started as long as the interface is active. Considering this in the power model leads to the energy cost for a single connection of

$$E_{c}(i, N) = E_i \delta_{i,1} + \int_{t_0}^{t_0+T_e} P_{c}(N) dt,$$

where $\delta_{i,1}$ is the Kronecker delta, returning one only if the arguments match. These calculations are interface-dependent.

For each transmission, the cost defined as energy per bit can be calculated, to make data transfers comparable. For this, the size of the transfer $S_t$ must be considered for normalization. Hence, the cost of each transfer is defined as

$$C_t = \frac{E_{c}(i, N)}{S_t}.$$

This metric can then be used in scheduling calculations to decide about the energy cost of a transmission on a specific interface, allowing the estimation of a connections cost before starting the transmission.

These energy estimates are to be used by MoNA to decide, when to allow or schedule transmissions on the mobile interfaces. In the case of scheduled connections, the assumption of synchronous connections is fulfilled. The energy estimate derived is then to be used to estimate the QoE of each scheduling option, to derive the most favorable decision for the user. For this, a deterministic QoE estimator similar to the ones described in [78] are to be used.

Knowledge of the power models of the devices involved in data transmissions, and computing in general, allows optimizing the operating point with respect to the power consumption of the individual device. If global knowledge of the power models and the system load is available, it becomes also possible to minimize the energy of the full system. If a limited accuracy is sufficient, energy models for each device class are required. For a more detailed power estimation, power models for each device type are required.

Future steps are the development of scheduling algorithms for network traffic and in the case of the Raspberry Pi also the caching, storage and forwarding.

### 4.5 Investigation of VM migration

One focal point of SmartenIT is the efficient management of traffic between clouds of a federation. Since clouds process workloads by virtual machines, this focal point includes...
the migration of VMs between data centers, which results in inter-cloud and often inter-AS traffic. To optimize this traffic not only with respect to routing but also with respect to transported data, this section presents preliminary modeling efforts to access the need of VM migration and the quality of SmartenIT’s VM migration mechanisms, to ensure that the right VMs are migrated. The innovation of this model is that it incorporates heterogeneous resources (CPU, RAM, disk space, bandwidth). Since the need to migrate a VM depends on the resource load on its hosting VM and the resources it is therefore allocated, the model is based on the consumption of multiple resources (CPU, RAM, disk space, and bandwidth).

Furthermore, the effectiveness of conducted live migration can only be measured by VM performance and accordingly customer satisfaction. Thus, a realistic model on VM multi-resource consumption is developed as a first step. Based on this model the time to process a workload in simulated scenarios will be evaluated and drawn directly to the satisfaction of the customer. However, also cases where VMs are live migrated between different data centers can be simulated, which is a scenario highly relevant for SmartenIT. These data centers may belong to different providers, i.e., the live migration of VMs in a cloud federation can be considered as well. Since multiple resources are considered in the model, the allocation (and migration) mechanism, as described in Section 5.3 and to be tested by the model, also takes multiple resources into account and ensures fairness based on these.

The model on VM utility consumptions will describe how many resources (CPU, RAM, disk I/O, bandwidth) different VM types require over time to process a certain workload. More precisely, for each VM type (VMs are created from flavors, which limits the range of VM configurations to be investigated) and workload it is specified how many resources are utilized over time by a VM of that type when processing this workload under perfect conditions, i.e., without resource overload. This data will be compiled by executing each workload on a VM of each type, where the VM's host sees no competing workload. The utilization of CPU, RAM, disk space, and bandwidth is measured during these runs and stored as a table, i.e., the table lists for every resource how much is consumed of it at periodic points in time. The workloads to be analyzed and to be fed to the simulator initially include a file synchronization service, video streaming, and a database server. For these workloads different functions (e.g., file upload, download, comparison; streaming, decoding; lookups, updates) will be analyzed and combined to realistic usage patterns.

The simulation will focus on VMs’ competition for resources. If a VM does receives less then the optimal amount of resources (due to strong contention), the model defines how the progression of the workload is decelerated. In particular, resources, which would be utilized during time \( t \) under optimal conditions (described by the tables), but are not allocated to the VM due to contention, will have to be utilized at \( t+1 \), additionally to the resources that are utilized at time \( t+1 \). However, since the lack of resources also forces the VM to process the workload at time \( t \) slower, the workload processing after time \( t \) is deferred. In particular, if the lack of resources causes only 70% of the tasks that are normally, i.e., under optimal conditions, processed at \( t \) to be processed, the remainder of these tasks needs to be processed at time \( t+1 \). Therefore, the tasks processed at time \( t+1 \) under optimal conditions will only start being processed at time \( t+1.3 \) (the resources required at \( t+1 \) therefore change to

\[
0.3 \times \text{“resources needed at } t \text{ under optimal conditions”} + \\
0.7 \times \text{“resources needed at } t+1 \text{ under optimal conditions”}
\]
wherefore 30% of $t+1$’s workload will have to be processed at time $t+2$. This again moves 30% of the workload normally processed at $t+2$ to $t+3$, and so forth. Additional lacks of resources defer resource utilization of subsequent points in time accordingly.

Formally this can be described as follows: let $d^t \geq 0$ be the delay a VM has experienced until time $t$ and $u^t$ the resources utilized under perfect conditions by the VM, i.e., the $t$th row of the table compiled by real world measurements. The resources it will then require at time $t$ are given by

$$res(t, d^t) := (1 - d^t + \lfloor d^t \rfloor) \cdot u^t + (d^t - \lfloor d^t \rfloor) \cdot u^{t-d^t}.$$ 

For example, we have $res(t, 0) = u^t$, $res(t, 0.3) = 0.7u^t + 0.3*u^{t-1}$. Also, if $x$ is an integer, we have $res(t, x) = u^{t-x}$.

Next it is defined how much delay a VM experiences in dependency on the required and the allocated resources: If the resource amounts required under optimal conditions are $r_1, r_2, \ldots, r_n$ at time $t$ (given by the table) and the resources allocated at time $t$ are $a_1, a_2, \ldots, a_n$, then the delay experienced during the $t$th

$$1 - \min \left\{ \frac{a_1}{r_1}, \frac{a_2}{r_2}, \ldots, \frac{a_n}{r_n} \right\},$$

i.e., the VM progresses as much as it receives of the resource it is allocated least of.

The model as defined now assumes that resources strongly depend on each other and once delay is caused it can never be compensated. The latter is justified, because if delay could be compensated at a later point in time, this would imply that during this time processing happens faster than under optimal conditions, which again would imply, that under optimal conditions not the VM’s full potential to process a workload is used. Thus, if a VM in the simulation cannot process a workload as fast as under optimal conditions, it cannot make up for this delay later. However, while this line of arguments certainly applies to the processing of workload it does not apply to the exporting of results, which is determined by the VM’s allocated bandwidth.

In particular, if a VM under optimal conditions exports results at some point via the network interface but does not receive this bandwidth during the simulation, the results can be buffered and exported when more bandwidth (compared to what would be used then under optimal condition) is available to the VM at a later point in time. This allows to make up for a temporary lack in bandwidth and thus, the resource of outgoing bandwidth is at least partially decoupled from the usage of other resources. However, if a VM needs outgoing bandwidth to exchange messages to receive input data, the temporary lack of bandwidth cannot be compensated. Therefore, whether outgoing bandwidth can be considered (partially) independent of the other resources or not, has to be defined according to the workload. For example, when the VM analyses scientific data and returns the results to an external desktop, the delay caused by a temporary lack of bandwidth can be compensated later, while this is not possible if the VM runs a streaming server, where data has to be exported in due time.

Live migrations of VMs will be triggered based on the delay of VMs, i.e., if the delay of a VM exceeds a certain threshold a live migration will be triggered for that VM to a less strained physical host. Furthermore, VMs may also be triggered without experiencing delay to unload physical hosts.

The utility function of VMs will be defined as the factor that it takes a VM longer to process a workload compared to optimal conditions, i.e., a utility of 1 is the optimum, while a utility
of 2 implies that the VM takes twice as long to process a workload then it would under optimal conditions. The satisfaction of a customer, i.e., his utility function, is then the average utility of his VMs. To arrive at conclusions about the efficiency of SmartenIT VM migration techniques, the increase in customer utility will be measured and compared to the customer utility achieved in simulations where SmartenIT mechanisms are not in place. Furthermore, also the fairness in the cloud according to the greediness metric (cf. Section 5.3.1) will serve as a performance indicator.

It can be concluded, that even simple resource dependencies are not straightforward to model, but the extra effort allows for more significant simulation results. The proposed dependencies will be further refined by measuring multi-resource consumptions of VMs that execute exemplary workloads. Furthermore, resources allocated to these VMs will be constrained and the effect of utilization of other resources monitored, to get an even clearer picture of resource dependencies.

### 4.6 Virtual Node Model

vINCENT is an incentive scheme for content distribution among uNaDas. In its current version, the model is extended to support the use case of mobile offloading (see Section 6.3). The scheme is based on classical bilateral Tit-for-Tat, which, as in BitTorrent’s unchoking algorithm, requires a balance of contributions between real nodes. vINCENT’s virtual node model extends this notion of balanced links between nodes to balanced virtual links between virtual nodes. A virtual node is defined as a cluster of trusted nodes. The notion of trust refers to the assumption, that members of the same virtual node do not behave fraudulent to other members of the same virtual node, e.g. since they belong to the same user.

As indicated in Figure 4-20 members of virtual nodes can have one of two roles: the role of a helper instance (HI) and the role of a data sink (DS). Helper instances exist to create credit in the network for other members of the virtual node, while data sinks are only consuming bandwidth from the network, depending on helper instances to support them. Besides these two roles, a node can have both roles at the same time (physical peer (PP)). Physical peers are bartering in a conventional way, i.e. they upload and download content at the same time. The set of connections maintained by the members of a virtual node to a specific other virtual node is called virtual link. Just as normal links in bilateral trade, virtual links are supposed to be balanced, i.e. ideally each piece sent from one virtual node to another virtual node is rewarded with a piece of data in exchange.

![Figure 4-20: Virtual node concept.](image)
However, a piece may be reciprocated by a different node than the consuming node, thus allowing for a split of contribution and consumption to different devices.

Figure 4-20 shows three virtual nodes A, B, and C where virtual node A and B each consist of a user’s uNaDa connected via fixed access technology and a smartphone connected via cellular access. The virtual link between A and C is balanced, when the helper instance in virtual node A uploads as much data to physical peer C as the data sink in virtual node A receives in return from C.

In the following, we describe the simulation model used to evaluate the mechanism of balancing virtual links.

### 4.6.1 Distributed Accounting Model

For enabling distributed accounting in the presence of virtual nodes, pull tokens are used as presented in [92]. A pull token is a temporarily valid identifier and represents a debt of a peer towards a specific third party. If a peer uploads a piece of content, the receiving peer issues a pull token, which will grant any other node possessing this token a piece to download. The data structure of a token contains the IP address/port pair of the issuing node, a time stamp, and a nonce large enough to prevent guessing by third parties.

Whenever a node sends pieces of data, it can be rewarded in two ways. Either, the receiver sends back a piece of data (Tit-for-Tat), or it sends back a pull-token. Helper instances will usually serve pieces in exchange of pull tokens only and forward the received pull tokens to a supported data sink, which can then redeem the token for a piece of data at the issuing node.

In order to detect and punish misbehaviour, helper instances rely on the feedback of data sinks regarding token redemption. Whenever a token cannot be redeemed by a sink, a message is sent to the helper instance, which in turn reduces the service for the misbehaving node, thus providing an incentive for cooperation. Further details on the token-trading approach are addressed in [92].

### 4.6.2 Communication Model within Virtual Nodes

vINCENT’s incentive scheme requires a protocol for intra-node communication providing a number of functionalities: support of authentication for node memberships and encryption to prevent eavesdropping of tokens, support of role descriptions to determine the roles of nodes within a virtual node, and support of heartbeat messages and rate control messages. The whole mechanism has been integrated into the TRANSIT P2P streaming system [93]. TRANSIT was designed to serve layered video content, e.g. H.264/SVC [95] and can thus vary the quality of the delivered video, allowing for the modelling of a wide range of scheduling modes.

### Scheduling Strategy Models

Using the same scheduling strategy for data sinks and helper instances is wasting optimization potential, as helper instances do not play back the stream. Instead, helper instances can download a fraction of the stream only and distribute the downloaded pieces widely in the network to create credit. How many and which pieces should be downloaded is determined by a scheduling strategy. In order to choose the best strategy, several scheduling strategies have been designed and evaluated as presented in the following:
Base Layer Only (BSO): This scheduling strategy downloads the SVC base layer only. All nodes need this layer to guarantee smooth playback. Thus, the base layer promises a high trading value to peers replicating this layer.

Highest Layer Only (HSO): The HSO strategy intends to download the highest layer only. Nodes possessing this layer also have to download all lower layers. Therefore, they are assumed to be good trading partners. Moreover, nodes possessing the highest layer are likely to be nodes with a high upstream bandwidth.

Random Scheduling (RAND): Random scheduling downloads random layers with no preference except for the most recent piece in reach. In [86], random scheduling was shown to be beneficial for conventional scheduling.

Every Other Frame (EALL): In this strategy, every second frame is skipped, while downloading every first frame with all quality layers, prioritizing the lower layers. This should effectively halve the traffic to be downloaded at the helper instance compared to a full retrieval of the stream.

Reverse Zig-Zag (RZZ): This strategy starts with the highest quality in the farthest available future, going down one quality layer and then back one in time following a zig-zag line. Reverse Zig-Zag was shown to be beneficial for SVC-based streaming systems in [85].

![Figure 4-21: Different scheduling strategies. Lower numbers indicate pieces with higher priority. The base layer is at the bottom.](image)

RAND, EALL and RZZ are further depicted as priority maps in Figure 4-21. The numbers illustrate the priorities given to a certain frame, where all strategies prioritize the most recent pieces as these pieces offer the highest trading value. Priority maps for each scheduling strategy are compared to the received buffer maps from the neighbouring nodes. Afterwards, the available piece with the highest priority is scheduled first.

### 4.6.3 Scenario Model

In the following, the viNCENT incentive scheme is evaluated on top of the TRANSIT P2P streaming system. All evaluations are conducted by means of experiments using the PeerfactSim.KOM framework [90]. Measurements were repeated 5 times using differing random seeds. All results are reported with 95% confidence intervals, if not stated otherwise.

Table 4-6: Bandwidth distribution of network model based on [89] (OECD) and [96] (VNI).
Evaluation Scenario Model

The mapping of the virtual node concept to the evaluation scenario is shown in Figure 4-20: Virtual node concept. above. The individual entities participating actively and passively in distributing the stream are mapped to entities in the incentive scheme as follows.

Physical peers are mapped to fixed access peers. As defined in the virtual node model above, these devices act as helper instances and data sinks at the same time, constituting a virtual node consisting of just one member. Thus, they upload as much as they download from the network. Helper instances are mapped to home gateways connected via fixed access networks. Home gateways create credit for mobile peers (see next paragraph) by performing the upload, while not playing back the stream themselves. Data sinks are mapped to mobile peers connected to the P2P network via a cellular connection (3G). They depend on a home gateway to provide credit. Mobile peers do not upload data to the network.

Moreover, the behavioural aspects of peers in the scope of the incentive scheme are mapped as follows: good peers try to achieve good playback performance through reciprocation, i.e. a good peer will always contribute as much bandwidth to the system as needed to view the desired video quality. As opposed to good peers, bad peers do not contribute bandwidth to the network, but only consume bandwidth from other peers (free riding). As an extension, they behave more aggressively in terms of hood selection, i.e., they try to maximize the probability to be unchoked optimistically by frequently changing the hood.

Network Model

In order to approximate the real world network situation, the bandwidth distribution of peers was chosen as measured by the Organisation for Economic Co-operation and Development (OECD) [89]. As [89] covers fixed network access only, figures for a mobile node’s average 3G access bandwidth was obtained from Cisco’s Visual Networking Index [96]. Based on these figures, nodes are divided into four bandwidth classes as depicted in Table 4-6. For simulating transmission delay, a normally distributed delay of 100 ms with a variance of 50 ms was used.

Churn Model

The peer churn model used in this work is based on measurement work of the PPLive network [91]. The authors of [91] showed that the probability p of a peer being online after x minutes follows the exponential distribution p = a*exp(b*<x), where a = 1.079 and b = −0.09594. In order to be able to control the size of the scenario, peers draw a session length following and join the system until a threshold of 300 nodes is reached, after which peers online are substituted only to keep the number of nodes constant.
Thus, the model reflects user’s viewing habits, i.e. the fluctuation of the peers (churn rate) is equivalent to the average churn rate of the PPLive system. The incorporation of a realistic churn model is important, as churn has major impact on system performance by determining the number of connection failures [94].

**Video Model**

The video model used for evaluation is derived from real SVC-encoded video files [85]. The resulting bit rates are depicted in Table 4-7 and are used throughout the experiments.

![Figure 4-22: Workload](image)

**Table 4-7: SVC bandwidth per layer [85].**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Bit rate (kbit/s)</th>
<th>Frame rate (f/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>303.8</td>
<td>3.75</td>
</tr>
<tr>
<td>1</td>
<td>503.0</td>
<td>7.5</td>
</tr>
<tr>
<td>2</td>
<td>705.3</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>905.8</td>
<td>30</td>
</tr>
</tbody>
</table>

**Parameter Setting**

Prior to the evaluation, several basic parameters of TRANSIT and vINCENT were chosen carefully as listed in Table 4-8. These parameters were varied independently of each other in separate simulation runs. The optimal setting (underlined in Table III) has been used for all consecutive experiments.

While the first four parameters from Table 4-8 are self-explanatory, the Neighbourhood Shuffling Interval determines the frequency with which low performing peers are exchanged for new Neighbours, the Sliding Window Size is the size of the window over which contributions are averaged, and the Unchoking Window Size determines the size of the grace period for new peers.

**Evaluation Metrics**

Suitable metrics to evaluate performance need to measure the following aspects of the proposed incentive scheme: (1) fairness and resilience as well as (2) QoS and QoE for the end user. For this purpose, three metrics are defined and applied:

Table 4-8: Evaluated parameters; optimal settings are underlined.
**Bandwidth spent/received**: This metric reflects the average bandwidth capacity spent and received over the entire streaming session by a node in the system.

**Playback Duration (PD)**: Playback duration describes, how long the stream is played back without stalling due to missing pieces. It is defined as the number of uninterrupted seconds of playback in the desired quality in a sliding window of $\Delta t = 60$ seconds. The metric is in the range of $[0; 60]$ by definition, with a perfect score being 60 seconds of playback and the worst score being 0 seconds of playback.

**Objective Quality of Experience (QoE)**: Playback duration alone does not represent the quality a user receives, as only the sum of seconds stalled is taken into consideration, but not the frequency and length of stalls. Therefore, the perceived experience is estimated using the QoE mapping by Hossfeld et al. (cf. [87], [88]), which was validated in user studies. The metric maps frequency (N) and average length (L) of stalling events to an estimated Mean Opinion Score (MOS) value ranging from 1 to 5, where 1 represents unacceptable quality and 5 represents excellent quality.
The metric is continuously calculated over an interval of $\Delta t = 30$s. Altogether, the defined metrics are suitable to judge incentive resilience, as bandwidth spent/received can quantify costs on a QoS base, while playback duration and QoE can measure the expected quality of a peer.

### Scheduling Strategy Evaluation

As a first step, the effect of scheduling strategies on the QoE of physical peers is evaluated in order to prevent side effects. Figure 4-23(a) shows this type of node to be nearly unaffected by the type of scheduling strategy applied by helper instances. All strategies offer a good stability as indicated by the small 95% confidence intervals. However, the scheduler implementation severely affects the QoE of a sink supported by a helper instance. More precisely, the scheduling strategies BSO and HSO offer lower performance than expected. This is caused by their nature to request only a certain layer of the video. However, if the sink is in need of a higher rate of token generation, there is no possibility to request higher layers to fill the gap. As opposed to that, EALL, RAND, and RZZ offer the desired quality for the sink.

Figure 4-23b depicts the average bandwidth spent/received of virtual nodes, illustrating the upload and download bandwidth consumed at the helper instance and the download traffic of the sink. For EALL and RAND, the up-/download of the helper instance are loosely coupled to each other with a 300 Kbps offset, whereas RZZ manages to keep the upload of the helper instance lower at a comparable QoE. BSO and HSO minimize the up-/download traffic. However, Figure 4-23(a) unveils that by doing so, they severely decrease QoE of data sinks.

Depending on the optimization goal (bandwidth efficiency at the helper instance or QoE on the sink), EALL, RAND, and RZZ provide a good trade-off between bandwidth usage and performance. Due to RZZ’s better bandwidth efficiency, this scheduling method is used for all further evaluations.

### Incentive Resilience

In order to evaluate the resilience of vINCENT, the scheme’s performance is measured under an increasing fraction of bad peers in the system as depicted in Figure 4-23(c) (for playback duration) and Figure 4-23(d) (for QoE). Both figures show the clear trend, that bad peers are separated from good peers, regardless whether the content is consumed by a physical peer or a sink in a virtual node. The gap in service between good and bad peers is more obvious for the playback duration, due to the QoE metric’s exponential decrease of quality (cf. [87], [88]), punishing even small deviations from perfect playback much harder than playback duration does. Nevertheless, it has to be considered that a MOS of 4 is still perceived as good playback quality, while a MOS of three or less is between annoying and slightly annoying.

Moreover, data sinks in a virtual node receive a slightly worse playback duration than physical peers caused by the additional overhead of token exchange. However, even though the playback performance of a data sink within a virtual node is not as good as that of a physical peer (probably due to signalling overhead), its performance does not depend on the amount of bad peers in the system. This is an important result, as bad peers as well as data sinks do not serve any requests.

In order to compare the vINCENT scheme to the classic Tit-for-Tat mechanism, we compare the performance of data sinks being supported and data sinks being unsupported. As discussed in the System Design (Section II), the latter case is equivalent to applying a Tit-for-Tat mechanism for resource poor nodes. Figure 4-23(e) shows a
comparison of the playback duration of both configurations. As expected, a large drop in performance for the sink can be observed for the Tit-for-Tat case, which is caused by the missing support of a helper instance. Thus, the sink has to use its own upload capacity to create credit in the system, which is not sufficient to reach a playback performance comparable to the case of the peer being supported by a helper instance.

![Graph showing message count and volume](image)

(a) Message count of helper instance.  
(b) Message size of helper instance.

Figure 4-24: Evaluation of token trading cost.

**Cost Analysis**

In the experiments above, the sink did not upload a single piece of video data. Thus, the overall goal of relieving the sink from uploading data can be achieved with the proposed approach. However, this comes at the expense of communication overhead induced on the sink, which is quantified in the following. To enable the token exchange, control messages and tokens need to be exchanged. The effects of piggybacking are reflected by the overhead message count of the helper instance, which is depicted in Figure 4-24(a). As the sink only receives tokens, which are piggybacked to requests for pieces of data to other peers in the network, the incoming message count is higher than the outgoing message count. However, considering the data volumes of control traffic transferred by the sink, incoming and outgoing traffic are nearly equal in volume (Figure 4-24(b)). Nevertheless, the amount of control traffic on uplink and downlink is as low as 100 Bytes per second, which corresponds to less than 1% when compared to the transferred video data.
5 Specification of Mechanisms for OFS

In this chapter, we present the SmartenIT mechanisms that perform the management of traffic exchanged among cloud, network providers and their business customers. The operation of these mechanisms pertains to the aggregate wholesale agreements among operators and the respective interconnection links. Overall, the solutions presented in the remainder of this section are specified to optimize in terms of cost and resource consumption and balance in terms of load the network. This implies that by adopting the SmartenIT mechanisms presented in this section, the costs of the operators are reduced, the efficiency of the network is increased and the quality of service experienced by the users is enhanced. Finally, the mechanisms presented in this section assume the Best Effort Internet as we know it: This is a conscious design choice so as to increase the mechanisms’ importance, applicability, potential for adoption and last but not least their impact on user experience from network and cloud services.

Below, three basic mechanisms and one synergetic solution of the Operator-Focused Scenario (OFS) are specified. Parameters and metrics for the performance evaluation of the mechanisms are provided, as well as the respective evaluation frameworks and some preliminary results where available. The results indicate that the SmartenIT OFS mechanisms comprise an elegant and powerful toolbox that can be applied in Internet without protocol modifications, benefiting all Internet stakeholders.

5.1 Dynamic Traffic Management (DTM)

DTM mechanism is dedicated for management of traffic related to inter-cloud or inter-Data Center (DC) communication. DTM aims at minimization of transit cost. It is assumed that traffic transfer on all inter-domain links managed by DTM is subject to some cost. The cost of traffic transfer on a given link depends on the tariff used between respective operators and is calculated using some cost function. The mechanism is designed to optimize ISP costs for two types of tariffs: total volume based and 95th percentile based.

First version of DTM mechanism (dedicated to overlay application traffic management) has been introduced in [5] and also described in [54] and [55]. The first version of DTM was designed for management of traffic generated by overlay application (e.g., p2p application) and assumed cooperation between ISP and overlay. The traffic that can be potentially influenced by DTM is called manageable traffic. It was assumed that ISP implements a kind of Oracle service and uses e.g., ALTO protocol for communication between overlay and ISP systems. In this way overlay application receives guidelines of content selection preferred by ISP and resulting in different interdomain link used for content transfer.

Here, we describe the mechanism extension for inter-cloud or inter-DC communication. The main assumption in this version is that the mechanism is overlay agnostic, i.e., no communication between ISP’s and DC/cloud systems. The basic principle of the mechanism remains the same, the traffic management decision is based on estimation of reference vector of traffic and measurement based periodic calculation of compensation vector. The main difference is that instead of giving guidelines to overlay (to select content available via preferred inter-domain link [5]) ISP directly manages flows by selecting a path they transferred through network. It is possible thanks to the idea of establishing tunnels for transferring manageable traffic.

The intelligence of DTM version for inter-cloud communication is described below. More technical details including algorithms and communication diagrams can be found in
appendices. SmartenIT system release v1.0 implements most of the functionalities of this version of DTM [8].

DTM is generally a solution addressing the Operator-Focused Scenario described in [6]. It addresses Use case 1 “Bulk data transfers for cloud operators” and also Use case 4 “Video content transfer between storages of independent clouds”. The last one may require some network resource allocation that can be provided by specific tunnel assignment (cf. details of DTM configuration in Appendix 13.1).

DTM application may be of great importance for the Virtual Private Cloud Business Case and the Federated Cloud business case. DTM in its simplest form is cloud agnostic. For example in federated cloud paradigm a few ISPs hosting federated clouds can manage traffic generated by interconnected clouds. The DTM management procedures are focused on lowering ISP’s inter-domain traffic transit costs.

5.1.1 Intelligence

DTM mechanism is explained below in the context of inter DC communication. Clouds may be composed of many DCs. Servers located in DCs as well as multiple DCs constituting a cloud are under supervision of a cloud management system. DCs may be located in distinct domains, therefore traffic sent between them passes inter-domain links. DTM focuses on a management if inter-DC traffic passing inter-domain links.

We assume that DC internal network is connected to ISP’s network via Data center Attachment point (DA). The DA is a device (router or switch) where DC inbound and outbound traffic is passed to an ISP network. The DAs belong to an ISP.

The DTM mechanism does not influence a cloud management system. All management procedures related to the DTM are limited to an ISP domain and the border points for management actions are DAs. This rule can be relaxed if a cloud belongs to an ISP or if it is federated with an ISP. DAs can be virtually or physically moved and located in a cloud network infrastructure. By virtually moved we mean that somewhere, depending on specific needs (cloud operator or ISP), a cloud operator deploys a virtual router or switch which uses the DTM mechanism exploiting information delivered by an ISP.

For simplicity we consider communicating DCs located in different ISP’s domains, let’s take into account DC-A1 and DC-B1 (Figure 5-1). In general, it is not important how many DCs are managed by DTM mechanism, it matters how many distinct DAs serve different DCs. In Figure 5-1, the DA-A1 and DA-A2 are connected to the same DA, the traffic to and from these two DCs can be managed as a single DC. We do not manage traffic between the DA-A1 and DA-A2.

Standard BGP operation allows to reach a network in any foreign ISP’s domain only via single inter-domain link (there is an exception when links are connected to a single neighbor AS). To make it possible to influence the distribution of the traffic among inter-domain links the DTM mechanism requires tunnels set up between remote DAs located in partner ISPs. Tunnels are setup in such a way that they traverse different inter-domain links. DTM manages the traffic by distributing flows amongst these tunnels. As a result the cost of inter-domain traffic may be decreased. The sum of traffic volume on both inter-domain links remains the same but the distribution of the traffic among links is changed in cost effective manner. Traffic directed to a given DC or cloud, localized in other ISP domain, is transferred via selected tunnel and, what follows, through intentionally selected egress inter-domain link. An operator can choose a tunnel technology, e.g., GRE, MPLS,
IPsec and others. By selecting specific type of tunnel technology one can acquire some additional features like security or bandwidth guaranties. Resource reservation and QoS guarantees and differentiation are not considered at the current stage but a more sophisticated traffic management is possible. Currently traffic is categorized only as manageable or non-manageable but always treated as best-effort. In present version of the DTM we use GRE tunnels.

We make a few initial assumptions:

- an ISPs' domains possess a multi-home connection to other domains,
- an ISPs' domains may be connected to each other via transit domains,
- an ISP’s border routers (BGP routers) exclusively announce different prefixes on different inter-domain links.

The first requirement is related to the basic idea of the DTM mechanism in which traffic is distributed amongst a few available links. The second assumption expresses the fact that traffic source and destination can be located in ISP’s domains which are separated by many transit domains. The third assumption is required for the inter domain links distinction procedure. The destination point should be accessible via a few inter domain links. Inter domain links passing traffic to a destination point are distinguished by IP addresses. These addresses are announced by Autonomous System (AS) hosting a destination point.

In Figure 5-1, we present a simple case of Inter-domain links (red solid lines A1, A2, B1 and B2) that interconnect Border Gateway (BG) routers. Cloud A and cloud B are located in the respective domains ISP-A and ISP-B. Each cloud internal network is connected to ISP’s network via data center attachment point: DA-A and DA-B, respectively. BG routers are communicating via the BGP protocol. Each ISP possesses a few pools of public addresses which are announced by the BGP routers. The ISP-A and ISP-B use all available links for data transfer; they announce separate pools of addresses on each inter-domain link. We consider as inter-DC communication the case where the DC-B sends data to the DC-A. The same schema applies symmetrically in both transfer directions and can be used independently in each direction. Let suppose that ISP-A announces the address pools XA, ZA on link A1 and the address pools WA, YA on link A2 (Figure 5-2).
Similarly ISP-B announces the pools XB, ZB on link B1 and the pools WB, YB on link B2, which is not depicted in the above figures since, in the rest of the DTM mechanism description, we consider data transfer only in one direction (from ISP-B to ISP-A). A group of addresses from all these pools are used by tunnels end points. The ISP-B uses routing policies which places appropriate prefixes in the routing tables in BG routers. These policies are composed in such a way that tunnel paths traverse suitable inter-domain links in ISP-A and ISP-B (Figure 5-3). The DCs located in the ISP’s domain may use a subset of pools used for tunnel ends or completely different pools of addresses. A destination end address in a DC is reached via a selected tunnel.

Figure 5-3: Tunnels used by DTM. End-points of each tunnel are addressed with different network prefix this results in different inter-domain link traversing by each tunnel.
The announcement of different address pools on different links is important because this way the data from a remote DC can be sent to a local DC following different inter-domain links. Tunnel ends use addresses from the previously mentioned, selectively announced address pools. For instance, the decision about the destination addresses in ISP-A domain is undertaken in the ISP-B domain. In this domain a right tunnel is chosen. If a packet sent from the DC-B1 to DC-A1 is directed to the tunnel T1, it will go via inter-domain links B1 and A1 (Figure 5-4). There are possible three other options (Figure 5-4) for choosing tunnels, for instance when we choose the tunnel T2 the data from DC-B1 will traverse inter-domain links B2 and A1.

The presented ISP’s physical interconnections and the routing policies used by ISPs offers the most flexible situation for the tunnel setup for link topology in Figure 5-4.

The ISP willing to manage inter-cloud data transfers deploys dedicated server (S-box) which enables trusted information exchange between cooperating ISPs. The S-box is a management decision node in the SmartenIT system (described in [9]). In our example ISP-A cooperates with ISP-B. ISPs hosting communicating DCs should set up tunnels ends at DA routers in each cooperating ISP domain. A BG router can also play a role of a DA router. Such DA router choice can lower the number of required tunnels when a single DC accesses ISP network in many points. This deployment would be especially useful when DTM is exploited in the end user focused scenario. Using a dedicated protocol, such as inter-ALTO (cf. [56], [57], and [58]), S-box receives IP addresses of a partner tunnels ends and recommendation for routing policies which enable flexible tunnel interconnection. These tunnels are set up for long time period and they are not expected to be changed very often. In our implementation we consider them static.

![Figure 5-4: The network resources used by the DTM mechanism.](image)

The DTM mechanism requires constant inter-domain links monitoring. Information about traffic amount transferred through these links is gathered from BG routers. This information is reported regularly to the S-box located in the same domain where the BG routers are (Figure 5-4, purple dashed lines). Let suppose that every $\Delta t=30$ seconds the report from all inter-domain links is be collected. The reports from different links should be synchronized. For this purpose, the network time synchronization offered by NTP is used. Each report is sent with timestamp to the S-box server. The S-box collects information about inbound and
outbound traffic provided by BG routers. It has been supposed that the DC being a source of traffic is located in the ISP-B (DC-B1 in Figure 5-4), the destination DC is hosted by the ISP-A (the same situation is reflected also in Figure 5-1). The inbound traffic reaching the ISP-A domain is reported to S-box in this domain.

The DTM evaluates the amount of traffic which have to be moved from one link to another in order to achieve some optimal monetary goal at the end of the billing period. This amount of traffic is represented in the form of vector – the compensation vector. The details are included in the appendix (section 16.1), also it was described in D2.3.

The S-box starts compensation vector calculation procedure (the part of the DTM mechanism – DTM algorithms are presented in Appendix 13.1) when reports from all inter-domain links marked with the same time stamp are gathered. This compensation vector is related to inbound traffic in the ISP-A domain. In the next step the S-box in the ISP-A domain sends the compensation vector (inbound compensation vector) to the S-box in the ISP-B. The S-box communicates is responsible for management of the DA router. The current implementation separate flow/packet distribution functions from traffic measurement and communication functions. Only flow/packet distribution is done by SDN controller (cf. DTM architecture described in [9]). This SDN controller dispatches flows coming from a cloud or DC to the appropriate tunnels according to the compensation vector.

Simultaneously the same S-box (in ISP-B domain) collects reports on outbound traffic from inter-domain links in ISP-B. The separate compensation vector for outbound traffic is established (outbound compensation vector). These two vectors do not need to be synchronized. Both compensation vectors are used for a selection of a tunnel which is most convenient for ISP-A and ISP-B. The first compensation vector expresses ISP-A requirements (an inbound compensation vector) and the second one illustrates ISP-B traffic pattern expectations (an outbound compensation vector).

The SDN controller in ISP-B domain dispatches flows from the DC to appropriate tunnels. This way there are two levels of management: an inbound and an outbound level. They can be applied exclusively or simultaneously, depending on link topologies and routing policies. It can happen that only one level of management can be applied. Even more, an application of both levels simultaneously may sometimes result in contradictory issues of an ISP-A and ISP-B, this case is described in Appendix 13.1. All final flow distribution decision related to transfer from DC-B1 is undertaken by the SDN controller located in the ISP-B domain which is under the ISP-B supervision. The SDN controller communicates with DA routers using OpenFlow and the proper assignment of flows to tunnels is done on the DA router connected to the DC.

The use of tunnels introduce some overhead (additional header is added to each packet to recognize tunnel). As a result the total amount of traffic exchanged on inter-domain links is in fact a bit greater than for the case without traffic management. However, the cost saving resulting from shifting a manageable traffic between links is greater than potential increase of cost due to overhead. This issue will be further studied to provide more precise assessment of system performance and cost effectiveness.

### 5.1.2 Parameters & metrics

DTM uses a few input parameters such as tariff details (billing rule and cost functions), billing period and reporting period. Within output parameters we distinguish measured metrics (periodically measured traffic volume on links and tunnels) as well as some performance indicators. Table 5.1 summarizes the input and output parameters.
Table 5-1: Parameters and metrics associated to the DTM mechanism.

<table>
<thead>
<tr>
<th>Input parameters</th>
<th>Output parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>• billing schema – volume or 95th percentile</td>
<td>• traffic volume from each inter-domain link gathered during report period</td>
</tr>
<tr>
<td>• link cost functions for particular billing schema (example cost functions are presented in the appendix)</td>
<td>• traffic volume received or send via each tunnel gathered during report period</td>
</tr>
<tr>
<td>• billing/accounting period</td>
<td>• transfer cost for billing period for each link (total volume or 95th percentile billing schema)</td>
</tr>
<tr>
<td>• report period</td>
<td>• ISP’s transfer cost</td>
</tr>
</tbody>
</table>

5.1.3 Evaluation framework

Within the simulation framework we distinguish two main simulation scenarios related to two versions of DTM, namely:

- the first version ([5], [54], [55]) dedicated for overlay traffic management based on cooperation between ISP and overlay and Oracle service
- the new version described in this section and implemented, i.e., DTM for inter-cloud (inter-DC) traffic.

Additionally, separate sets of experiments are dedicated for two versions of algorithms: designed for volume based tariff and 95th percentile tariff, respectively.

For the evaluation of DTM, a proprietary simulator developed by AGH and the ns-3 simulator [59] have been employed. The proprietary simulator is dedicated to evaluation of the first version of DTM, i.e. the case of overlay traffic management based on cooperation between ISP and overlay and Oracle service. It simulates the traffic at the level of flows. The proprietary simulator has been used in all experiments which are reported in [23], [5], [54] and [55]. The evaluation based on ns-3 simulator is in progress and will be reported in D2.5.

5.1.4 Evaluation results

Several experiments have been done for the first version of DTM [5] (dedicated for overlay traffic management based on cooperation between ISP and overlay and Oracle service). In the first set of experiments we inspected the operation of the DTM with links charged on the basis of traffic volume. The results were presented in [5] and in [54]. The second set of experiments has been performed for an AS with two inter-domain links charged according
to 95th percentile rule. Extended description of experiments’ set-up and results can be found in [55]. In the current deliverable we only summarize results.

Figure 5-5: Distribution of observed 95th percentile sample pairs, measured during the billing period; values of achieved traffic vector, reference vector, and related inter-domain traffic costs for the experiment without DTM (red) and with DTM (blue): (a) success rate 40%; (b) success rate 30%.

We present here results only for 40% and 30% success rates in Figure 5-5(a) and Figure 5-5(b), respectively. Success rate represents the level of effectiveness of traffic management: success rate of 30% means that, on average, 30% of available manageable traffic is effectively managed by DTM. Red dots represent pairs of 5-minute samples collected during the whole billing period for the experiment without DTM. Darker red dots represent 5% of the highest samples, i.e., samples above 95th thresholds that determine the cost of inter-domain traffic. The corresponding vector $\mathbf{x}^{95}$ is also plotted (red). Its components represent 95th thresholds on the respective links. The corresponding total cost is 22038 units. Similarly, the distribution of the 5-minute traffic sample pairs for the experiment with traffic management is presented (blue dots and vectors). It is shown that the distribution of sample pairs is less spread out than in the case without DTM.

Moreover, for the success rate of 40% the distribution is more condensed than for 30%. As a result of DTM mechanism operation sample pairs tend to concentrate around the line delineated by reference vector $\mathbf{x}^{95}$. This is because during each 5-minute sampling period the calculated compensation vector strives to influence traffic distribution between links in such a way that the proportion of both traffic samples is the same as the proportion of corresponding $\mathbf{R}$ vector components. The more manageable traffic is effectively influenced the more the distribution of sample pairs is condensed around reference vector.

Therefore, even small samples that are not above the thresholds (do not account for the top 5% of the highest samples) are influenced. The higher the success rate, the higher the concentration around the $\mathbf{R}$ vector. Figure 5-5 also shows the $\mathbf{x}^{95}$ vector value achieved at the end of accounting period for experiments with DTM, as well as the corresponding total cost. In the case of the 40% success rate the $\mathbf{x}^{95}$ vector almost covers the $\mathbf{R}$ vector (Figure 5-5(a)). Additionally, the achieved cost of 15654 is very close to the optimum
15622. For the 30% success rate the optimal solution was not achieved. The $\vec{X}^{0.95}$ vector diverges from the $\vec{R}$ vector and the traffic cost is higher; it equals 16465 units (Figure 5-5 (b)). This stems from the fact that the DTM mechanism was able to shift insufficient volume of traffic from the inter-domain link 2 to link 1.

Figure 5-6: Sorted 5-minute samples with (blue) and without (red) DTM compensation and corresponding 95th percentile thresholds, link 1 and link 2, success rate 40%.

Figure 5-6 shows sorted 5-minute samples on inter-domain links 1 and 2. On each plot we show the distribution of samples for the experiment with and without DTM. It is clear that not only did the respective thresholds change but the whole distribution was modified. Samples on link 1 increased, while on link 2 they decreased since some amount of manageable traffic was shifted.

The results show that total transfer cost may be decreased. There is a potential for cost optimization in presence of manageable traffic for which the inter-domain link may be selected. Next steps encompass simulator update to assure that it works exactly as in prototype release (implement all features). Then not only testbed experiments may be supported and verified by simulations but we will be able to simulate more complex scenarios and tune DTM algorithms.

The basic simulation experiments planned for data center/cloud oriented scenario will be performed using topology presented in Figure 5-1. The optimization and management for volume based and 95th percentile rule based tariffs will be evaluated. On this topology we also plan to conduct separate simulations for inbound and outbound traffic management as well as for combined management. Next experiments are planned for more complex use-cases, e.g. including simultaneous traffic management performed by two operators (both having 2 inter-domain links). In the most complex use-case there will be multiple clouds/DCs, multiple ISPs.

5.2 Inter-Cloud Communication (ICC)

The main rationale of the ICC mechanism is to utilize the transit link traffic in an optimal way given the 95th percentile charge so as to reduce the transit charge by means of controlling the rate of the portion of the traffic that has been marked as time-shiftable (details in Appendix 13.3). This separation of control over time-shiftable and real-time traffic allows the ICC mechanism to improve the QoE of the real-time traffic, to reduce the transit charge of the ISP, and to meet a given goal for the target rate ($C_{target}$) that will determine the ISP charge, thus also removing uncertainty over ISP's charge.
This is achieved by means of controlling the rate and shifting in time the time-shiftable traffic transit link transmissions in epochs where the transit link utilization is lower and thus more traffic can be sent for free without violating $C_{target}$. The mechanism also has built-in support for cloud services, allowing the cloud and network layer to exchange information of the current status of the network and cloud infrastructure and thus enable educated decisions benefiting both the cloud and network infrastructure (e.g. perform a bulk data transfer for back up reasons to the cheapest destination Data Center (DC) when the network is lightly loaded). Hence, the ICC mechanism involves two layers: the network layer and the cloud layer.

### 5.2.1 Intelligence

There are a set of unique features in the ICC mechanism intelligence that differentiate it from related work, namely:

a) The mechanism operates in very small time scales, in the order of seconds, and less than the 5-min interval where most other mechanisms operate (e.g. NetStitcher [60]), introducing the notion of slice, a portion of the 5-min interval; This allows a finer granularity in decision making and control over the rate of the traffic that is to be sent, further empowering the ISP to attain significant savings even in cases where his expectations regarding traffic may be wrong.

b) The mechanism has built-in support for communication among ISP and Cloud layers, thus communicating each layer’s preferences to the other.

c) Though ICC is primarily a network layer mechanism, it fully respects the cloud layer business models and decisions (e.g. selection of destination cloud for bulk data transfers) and is built around the design-for-tussle principle, allowing the major stakeholders to conduct their business as usual without promoting any stakeholder in expense of others, while revealing more inter-layer information that could help stakeholders optimize their decisions leading to win-win outcomes for all (cloud and network) stakeholders involved.

d) The mechanism is applicable even in cases of single-homed ISPs, i.e. when there is only one outgoing transit link.

e) ICC has built-in support for both federated and non-federated DCs/clouds.

The constituent elements of the mechanism include (cf. Figure 5-7):

**SmartenIT Information Service (SmaS):** SmaS is responsible for characterizing a set of destinations for a specific amount of data to be replicated within a specific time interval. Specifically, SmaS receives as input from the cloud layer: i) the amount of data to be transferred in MBs, ii) the priority level for this amount of data and iii) the set of candidate destinations (i.e. IP addresses of data centers of the same or different cloud operator).

**Cloud Scheduler (CloS):** CloS is a centralized service running in the cloud layer and is responsible for making decision of where to allocate data, i.e. to which cloud(s) to send data, e.g., either for fault-tolerance, or for QoE enhancement. In the case where no federation is established, a CloS instance runs in each cloud, else the federated clouds are assumed to trust a third-party entity running CloS, which is responsible for scheduling data exchanges within the federation.

Moreover, the successful operation of the proposed mechanism also relies on the following supportive or complementary functionality:
- **Inter-SmaS communication protocol**: This protocol supports the exchange (or exposure) of information between the SmaSs of the NSPs involved in the inter-cloud communication. This communication protocol can follow and extend the inter-ALTO IETF approach.

- **(Cross-layer) CloS-to-SmaS communication protocol**: This protocol is required for the communication of the network layer (SmaS) and the cloud layer (CloS).

- **CloS-to-Federated Cloud communication protocol**: This protocol facilitates the periodic feed of CloS by CloIs with overlay information necessary to make its scheduling decisions.

![Image of ICC constituent elements](image)

**Figure 5-7: ICC constituent elements.**

### Cloud Layer

CloS provides as input to SmaS a list of destinations and receives back a list where a weight (representing network cost) is assigned to each one of them. CloS makes the final decision \( d^* \) for a specific traffic flow of a given cloud operator based on the total cost \( C_d \) that characterizes each potential destination, where \( C_d \) includes i) the network cost \( P_d \), i.e. the weight provided by SmaS, and ii) the cloud cost \( L_d \) associated to each destination \( d \) by the scheduler. Ultimately the CloS may be aiming to achieve either an optimum for that specific flow:

\[
d^* = \arg \min_d C_d(P_d, L_d)
\]

or an optimum for a broader set of flows, or even for the entire cloud federation, considering at the same time the other flows as well and the impact of each decision on them.

Note that CloS treats the weights provided by SmaS as network cost. Nonetheless, SmaS reveals only relative values, this is also why they are called “weights”, and not actual network cost, as the latter might result in revealing critical information to competitors and
outsiders. We provide a detailed specification on how these cost metrics can be computed in Appendix 13.3: the simplest yet generic approach wrt to cost is to consider as cost metric a monetary ask, i.e. a monetary compensation sought. For brevity reasons, we omit the detailed presentation of these issues in this section and instead focus on the network layer.

Network Layer

Periodically, SmaS gathers values of several cost metrics related to the underlying IP network such as latency (i.e. seconds), congestion (i.e. number of timeouts or packets dropped), available bandwidth, number of hops, geographical distance and BGP information as a proxy for cost related to interconnection agreements (e.g., transit or peering). Then, the cost characterization of the network path leading to each specific destination is performed by calculating the minimum "weight" or cost (corresponding to network cost) \( P_d(p) \) for each considered destination \( d \) based on BGP hop count. SmaS is consistent with the ALTO approach and the FI design principles. We now henceforth restrict attention to the ICC operation over a single (transit) link:

The main idea of the ICC mechanism is that a portion of the transit link(s) traffic, i.e. the "time-shiftable" part, can be delayed so as not to violate a target threshold which will determine the ISP charge according to the 95th percentile rule. Along with the ISP cost savings due to the lower transit link charge, the shaping of the time-shiftable traffic will also result in monetary compensations that need to be paid by the ISP to his respective customers (such as Data Centers, Clouds) who experienced the additional compared to Best Effort transmission – delays. The ICC mechanism is more meaningful to apply if the per unit-time delay penalties per Mbyte to be incurred by the ISP are expected to be considerably lower than the corresponding extra transit costs.

Home ISP applies the following algorithm (also referred to as Shiftable Traffic Scheduling – StraS) within \( y \) equally-sized epochs of every 5-min interval within a billing cycle, i.e. within the period within which the transit links are metered and charged (i.e. a month): Initially the target value of the transit link 95th percentile \( C_{\text{target}} \) is set, given the past history of traffic patterns and the Cumulative Aggregate Growth Rate (CAGR) estimated from statistics. Throughout each billing period where the 95th percentile is computed and the ISP is charged, i.e. for each month, the algorithm decides when (i.e. in which 5-min intervals – see below) to send the time-shiftable traffic and at which rate; the real-time traffic is forwarded as usual in the Best Effort Internet. On the contrary, the time-shiftable traffic may experience further delays. In particular, each of the 5-min time intervals is sliced to \( y \) epochs, e.g. for \( y=2 \) each 5-min interval is sliced to two 2.5min epochs. The algorithm uses these epochs in order to control the amount of time-shiftable traffic to be transmitted taking into account the definition and calculation of proper thresholds that can be in general different among different (5-min/y)-size epochs so that this traffic combined with the real-time traffic (which is not known a priori) does not exceed the \( C_{\text{target}} \) rate over the 5-min interval.

The ICC mechanism also takes advantage of the fact that the \( C_{\text{target}} \) can be violated at 5% of the 5-min intervals of the billing period without any cost by sending with maximum rate (thus constrained only by the transit link capacity) at the last 5-min periods so that more data can be sent without additional cost for the ISP. This basic definition of the ICC mechanism is henceforth referred to as "ICC Fixed At max rate at the end" or ICC_FA for short. Additional variations of the ICC may prescribe different ways of choosing the 5% 5-min intervals where the transit link will be utilized with max rate; in this document we deal
with an additional such variation prescribing that the mechanism selects the 5% peak periods in order to send with the maximum rate. This variation is henceforth referred to as “ICC Fixed At Peak periods with max rate”, or ICC_FAP for short. ICC_FA and ICC_FAP are cross-compared and evaluated in the remainder subsections of this section.

5.2.2 Parameters & metrics

The key parameters are provided in Table 5-2. Note that the cloud layer parameters are currently not supported since the current evaluation framework supports only the network layer of the ICC mechanism.

Table 5-2: Key parameters for ICC.

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transit link capacity $C$</td>
</tr>
<tr>
<td>ICC version</td>
</tr>
<tr>
<td>Traffic traces</td>
</tr>
<tr>
<td>Target 95th percentile $C_{target}$</td>
</tr>
<tr>
<td>Number of epochs $y$</td>
</tr>
<tr>
<td>Threshold parameters</td>
</tr>
<tr>
<td>Sensitivity analysis range</td>
</tr>
<tr>
<td>Type of service and customers</td>
</tr>
<tr>
<td>Network topology</td>
</tr>
<tr>
<td>Collaboration form</td>
</tr>
<tr>
<td>Cloud layer parameters</td>
</tr>
<tr>
<td>(currently not evaluated)</td>
</tr>
</tbody>
</table>

The transit link capacity $C$ can be provided as input if it is known. In case it is not known, this can also be dynamically computed by the trace values by means of extracting the maximum rate in the trace and projecting it by epsilon to a $C$ value.

The version of the ICC mechanism to be employed defines the behavior of the algorithm used for the estimation of expected real time traffic and the selection of the (less than 5%) 5-min intervals where ICC will transmit with maximum rate.

Input traces and assumptions on mix of time-shiftable and real-time traffic provide the link traffic upon which ICC will be applied. Traces are obtained from real-network monitor tools such as [55], while it is also possible to set specific data values for theoretical study purposes. Additional trace files have been obtained by SmartenIT partners and third parties. There is a variety of formats that has been processed and different information that can be extracted, with the transmission rates being the minimum information available.

$C_{target}$ is the target ISP charge when employing the ICC mechanism, i.e. the target rate over the transit link ($C_{target}$) that should not be exceeded at the 95% of the 5-min intervals of the billing cycle (month).

The value of the number of epochs per 5-min interval $y$ defines how many times the ICC control rate algorithm is applied. The threshold values per epoch determine how close to the $C_{target}$ will be the attempted rate decided by the ICC algorithm within each $y$-epoch: this is needed due to the a priori unknown values of the real time traffic.

The sensitivity analysis range over input parameters of interest (e.g. the target that ICC must achieve regarding the transit link traffic savings) enable the simulator to check the robustness of the results and the impact of fine-tuning the mechanism parameters.
Type of service is the service associated with the evaluation, such as video delivery, bulk data transfers, load and VM migrations among clouds and data centers for data replication and backup purposes to assure fault-tolerance, content placement and load or VM migration to enhance users’ QoE. Based on the type of service we can also characterize the traffic to be delivered over the inter-cloud communication, e.g., real-time, delay-tolerant, etc.

The type of customers for Inter-cloud communication is generated by service offered to customers of clouds; these can be either other service providers, called Over-The-Top providers (OTTs), such as content providers and gaming providers, or end-users, both corporate, e.g., banks, organizations, and residential ones.

Regarding network topology, this parameter mainly refers to the inter-connection of different ISPs or ISPs and DCs, as well as respective interconnection agreements, e.g., transit or peering. This parameter is required once the Cloud layer of the ICC mechanism is instantiated as well and thus is not currently used in the simulation results provided later in this section.

The form of collaboration may take various forms once the Cloud layer of the ICC mechanism is instantiated: The type of collaboration between the various stakeholders is associated to the various business agreements that may be established between them and may significantly influence the operation of the proposed mechanism. Specifically, we identify three types of collaboration: a) Collaboration in the network layer between ISPs and particularly their SmaS servers, b) Collaboration in the cloud layer between cloud operators, i.e. the cases of non-federation and federation, and c) Cross-layer collaboration between the network and the cloud layers, between ISP(s) and clouds (or a cloud federation), between SmaS and CloS.

Finally, there is a set of cloud layer parameters to be integrated in the simulator once the cloud layer is supported, such as the number of DCs, the absence or presence of federation, the energy/load models for the respective DCs and the network topology interconnecting the respective DCs.

The key metrics are provided below. Note that the cloud layer metrics are currently not supported since the current evaluation framework supports only the network layer of the ICC mechanism.

- Volume of traffic transferred (under Best Effort and ICC),
- Volume of time-shiftable traffic delayed,
- Delay penalties for the shifted traffic,
- 95th percentile for the transit link under Best Effort and the variants of the ICC that are evaluated,
- ISP cost savings over the transit link, i.e. the 95th percentile reduction,
- Traffic variability over time, i.e. the modified traffic pattern as a result of applying the ICC mechanism,
- Cost/energy savings for the ISPs and the DCs,
- Distribution of data/workload among DCs, and
- DC Federation energy consumption.

5.2.3 Evaluation framework

The evaluation framework is a set of two applications implemented in J2SE: The SimulatorICC application performs the simulation and measurements of the ICC
mechanisms’ performance under various parameter values, while the PlotterApp application performs the visualization of the produced results via a Graphical User Interface.

The current implementation flavor focuses on the network layer and the impact of ICC on transit link cost savings and additional delays incurred to the time-shiftable traffic due to ICC. The addition of the cloud layer is on-going work.

All the components of the SimulatorICC application are implemented in a modular OOP fashion as a set of Java classes. There are two main applications, the Simulator which performs the simulations and the PlotterApp which visualizes results and statistics attained per experiment.

5.2.4 Evaluation results

We provide in this subsection some indicative evaluation results. In particular, we evaluate two versions of ICC, the ICC_FA and ICC_FAP (see Section13.3) and cross-compare their performance with that of Best Effort Internet. For presentation and brevity reasons, we provide a subset of results for a limited input trace, which has been provided by DTAG, taken from a link whose capacity is 300 Mbps. The trace is of one hour duration, hence there are 12 5-min intervals. For demo reasons, we set the 95th percentile, which is formally undefined for such a small sample, to be the second highest value of the dataset. For the simulation results presented below, the parameter y is set to 2 - once again this small value is set for simplifying the respective plots. The threshold value for the first epoch (of duration 2.5mins) is set to 95%, hence a rate of 0.95 * Ctarget must be met for the first epoch, while for the second (last) epoch the volume of data is such that the CTarget goal is met for the 5-min interval. The time-shiftable traffic is assumed to be the 25% of the total traffic throughout the trace.

The evaluation indicates some interesting findings:

- ICC can ensure that the target 95th percentile is met assuming that there is accurate prediction of the expected real-time and time-shiftable traffic.
- ICC_FAP always outperforms ICC_FA in terms of delays. Simulations indicate that the extra delays incurred due to ICC operation can be substantially reduced under ICC_FAP in comparison to ICC_FA. This is due to the fact that sending with max rate at peak periods, that are in general not always at the end of the billing cycle, allows the ICC mechanism to shorten the queue length of the data waiting for transmission compared to ICC_FAP that utilizes the link with max rate only at the end, thus delaying more packets for more time.
- ICC_FA and ICC_FAP may transmit less data than Best Effort in a given time window in order not to violate the Ctarget 95th percentile (charge). For a given time window also ICC_FA may outperform ICC_FAP in terms of volume of bits transmitted but both ICC_FA and ICC_FAP performance in terms of volume of data come really close or are identical.
- The more aggressive the setting of Ctarget is (i.e. the lower value is set given the utilization of the link) the higher the respective delays and delay penalties are, since more data are delayed in time so as not to violate Ctarget. Indeed, a low value of Ctarget - compared to the value of the Best Effort 95th percentile – indicates that the effective rate under ICC will be considerably lower, thus increasing the number of intervals and the respective number of packets that will be queued and thus delayed at all these intervals where the link is utilized at values beyond Ctarget.
• ICC can shape the traffic over the transit link according to the values of Ctarget, and
the traffic threshold values of the y epochs. Thus, the ICC traffic patterns are
predictable, as opposed to Best Effort Internet.

In Figure 5-8, Figure 5-9 and Figure 5-10, we provide some plots and indicative results for
C\text{target} = 285 \text{ Mbps}, one of the many values for which a sensitivity analysis of ICC
performance has been carried out, starting with the traffic patterns attained under Best
Effort, ICC\_FAP and ICC\_FA. The value of C\text{target} is lower than the 95\text{th}
percentile attained under Best Effort, which as indicated below is 293.75Mbps. Note that ICC\_FAP
by sending with max rate at the peak 5-min interval (i.e. the second 5-min interval)
manages not to have any time-shiftable traffic queued at the 9\text{th} 5-min interval: This results
in substantial lower cumulative delays compared to ICC\_FAP that by definition transmits
with maximum rate at the last slot, which emulates the last 5% periods of the billing cycle.

Figure 5-8: The Best Effort traffic pattern and 95\text{th} percentile.

Figure 5-9: The ICC\_FAP traffic pattern, the 95\text{th} percentile and the extra delays
incurred.

Figure 5-10: The ICC\_FA traffic pattern, the 95\text{th} percentile and the extra delays
incurred.
Figure 5-10: The ICC_FA traffic pattern, the 95th percentile and the extra delays incurred.

The difference in the rationale of ICC_FA and ICC_FAP is also displayed in a separate plot of Figure 5-11, that is cross-comparing the two ICC variations’ resulting traffic patterns. In Figure 5-11, we also provide a similar plot for the comparison of ICC_FA with Best Effort. Note that in all the plots the amount of bits transferred in the simulation time window are also displayed as legends at the bottom of the plots.

Figure 5-11: Comparison of traffic patterns attained under Best Effort, ICC_FA, ICC_FAP.

Note that the delays, the amount of data transmitted and the traffic patterns are different if a lower Ctarget is set, thus demonstrating the impact of the Ctarget value. Below we provide as Figure 5-12, Figure 5-13, Figure 5-14, and Figure 5-15 some results for Ctarget=270Mbps. Figure 5-12 provides the Best Effort Internet traffic pattern when ICC is absent.

Figure 5-12: The Best Effort Internet traffic pattern, the 95th percentile: 270000.0 Delays due to ICC_FAP: 3.12657004E6

Figure 5-13: The ICC_FA traffic pattern, the 95th percentile: 270000.0 Delays due to ICC_FAP: 4.02657004E6
The evaluation of the ICC network layer has demonstrated that the mechanisms can result to transit link savings for the ISP. There is a tradeoff between the amount of these savings and the extra delay incurred to time-shiftable traffic, which is depicted to the setting of the $C_{target}$ value. An additional benefit of the ICC is the better performance of the real-time traffic which is essentially prioritized at peak periods; also since some of the time-shiftable traffic is not present, the performance of the real-time services in terms of delay are expected to improve when it really matters, i.e. when the network is loaded. Moreover, the shifting of traffic at off peak periods load balances the ISP inter-domain traffic. The latter is especially true for the ICC_FA variation, as opposed to the ICC_FAP variation that is capable of attaining lower extra delays for the time-shiftable traffic compared to those attained under ICC_FA but does not help the real-time traffic at peak periods. Therefore, critical traffic such as real-time or zero-tolerance cloud services traffic under ICC will experience better performance compared to current Best Effort Internet, thus benefiting the user, the cloud service provider and also the ISP.

In terms of applicability, it is worth emphasizing that ICC is designed to work on top of transit link traffic aggregates, respecting current business practices and agreements. It does not require any kind of deployment at business customers’ or end users’ premises or any protocol modifications. Thus is readily applicable for ISPs and can operate over the Internet as we know it. A detailed discussion of the design decisions that ensure the mechanism’s applicability as well as guidelines/alternatives on the deployment options are provided in Section 13.3.
Finally, the evaluation of the mechanism is on-going. More traces and traffic models are used for the evaluations, while larger traces when available will allow a deeper understanding of the fine-tuning and learning capabilities of the ICC network layer (i.e. the STraS algorithm). Last but not least, the cloud layer of the ICC mechanism which is responsible for deciding on the optimal destination datacenter selection when multiple alternatives are available – for the inter-cloud/-DC use case scenario – is also on-going: this selection will be taking into account both the network load and the energy cost of the IT infrastructure of the datacenters. This layer will enable further optimizations on the management of inter-domain traffic for the specific case of cloud services.

5.3 Multi-Resource Allocation (MRA)

The Multi Resource Allocation (MRA) mechanism allows to achieve a fair allocation of multiple resources (CPU, RAM, disk, bandwidth) between VMs or even customers (aggregate of VMs) and therefore addresses Use Case 3 “Resource Allocation” and potentially Use Case 2 “Automated storage tiering in the federated clouds” of the Operator-Focused Scenario [6]. The latter connection holds, because mass storages with different access speeds and volumes also constitute different resources and therefore could be allocated by the MRA mechanism. The MRA mechanism helps in allocating heterogeneous resources, such as CPU, RAM, disk space, and bandwidth, in a fair manner between customers of a cloud or a cloud federation. In particular, it makes the multi-resource consumption profiles of customers comparable and thereby allows making decisions on which Virtual Machine (VM) should receive more resources in case of overload and which VM should “live migrate” (see terminology in Section 5.3.1) to another Physical Machine (PM) or even to another cloud in case of federations. Since the MRA enforces fairness, it provides incentives to customers to not overuse cloud resources, as this may lead to constraining the resources allocated to their VMs or live migrating them (in favor of VMs of customers that consume resource more appropriate). The most relevant terms are introduced in Section 13.4.1 and the most relevant methods in Section 13.4.4.

5.3.1 Intelligence

The MRA mechanism collects consumption information from all PMs in a cloud or federation about the resource consumption of VMs. This consumption information includes resources as diverse as CPU, RAM, disk, and bandwidth. Based on this information the greediness (cf. Section 5.3.1) of each customer, which is a single number that increases with immoderate consumption of the customer, is calculated. These numbers are announced to all PMs. If a resource of a PM gets overloaded, the hypervisor component of the PM takes the greediness into account to reallocate the overloaded resource, i.e., VMs of customers who are rated as “moderate” (i.e., have a lower greediness) receive more of the scarce resource. The novelty of this approach is, that it considers heterogenous resources and allows to enforce a fairness metric, which does not make assumptions about the VM or customer preference functions (this ensures applicability to many scenarios). Also the approach is novel in that it allows to reallocate resources between VMs that ensures fairness between customers, i.e., although atomic reallocations of individual resources of a PM are made, these reallocation aids the cloud or even federation-wide fairness between customers.

Consumption Collector

In order to achieve a fair resource allocation, information about the resources utilized is needed. Since customers utilize resources via VMs, the resources utilized by VMs need to
be inferred and aggregated to their owners/customers. It is important to note that the resources utilized by a VM are not equal to its VRs. Rather, the VRs of a VM are an upper bound for the PRs it utilizes. Therefore, it is insufficient to consider the VRs the VMs of a customer have but the actual consumption profiles of VMs need to be inferred from PMs periodically. Since individual PMs can conduct measurements constantly and return cumulated results, a periodicity of 10 sec to 1 min is expected to be optimal for returning results to the central nodes (as no measurement granularity is lost).

**Greediness Calculator**

The information gathered by the consumption collector is fed to the greediness calculator, which further needs to receive information about how many PR each PM has and how many VRs each VM has. However, since PRs are static and VRs fixed upon VM creation, this information can easily be obtained from the orchestration layer. Based on this information the greediness calculator determines the greediness of each customer according to the definition given below. The greediness vector (a vector with the greediness of every customer) is then returned to the hypervisor of each PM. To announce the greediness to other clusters of the cloud or other clouds in the federation it is sent to the respective central nodes. A central node adds the received greediness vectors to the greediness vector it did most recently calculate and distributes the result among its supervised PMs.

The greediness of a customer is defined as the sum of greediness of his VMs. The greediness of a VM depends on its endowment, on the resources it utilizes, the endowments and utilizations of the other VMs that run on the same PM, and the PRs of the PM. Subsequently, greediness is formally defined.

Let \( R = \{r_1, r_2, \ldots, r_n\} \) be a set of \( n \) PRs under contention, e.g., \( R \) may include CPU, RAM, storage, and bandwidth. Since for every PR under contention, also a VR exists, quantities of PRs and VRs can be denoted by vector \( x \in \mathbb{R}^n_{\geq 0} \) where \( x \) contains amount \( x_j \) of resource \( r_j \). Thus, \( res(y) \in \mathbb{R}^n_{\geq 0} \) either denotes \( y \)'s PRs or VRs, depending on whether \( y \) is a PM or VM, \( virt(z) \in \mathbb{R}^n_{\geq 0} \) is the set of VMs running on PM \( z \), if \( z \) is a PM, or the set of VMs at \( z \)'s disposal, if \( z \) is a customer, and the quota of customer \( c \) is denoted by \( quot(c) \in \mathbb{R}^n_{\geq 0} \).

Let \( p \) be a PM with \( virt(p) = \{v_1, v_2, \ldots, v_m\} \). An allocation of \( p \)'s resources is a matrix \( A \in \mathbb{R}^m_{\geq 0}^{n \times n} \) with

\[
\sum_{i=1}^{m} a_{ij} \leq res(p)_j
\]

for all \( j \in \{1, 2, \ldots, n\} \), where \( v_i \) receives amount \( a_{ij} \) of \( r_j \), i.e., the \( i \)th row of \( A \) is \( res(v_i)^T \). The endowments of VMs are given by matrix

\[
B \in \mathbb{R}^m_{\geq 0}^{n \times n} \text{ with } \sum_{i=1}^{m} b_{ij} \leq res(p)_j \text{ for all } j \in \{1, 2, \ldots, n\},
\]

for all \( j \in \{1, 2, \ldots, n\} \), where \( v_i \)'s endowment to \( r_j \) is \( b_{ij} \). The amount of \( r_j \) that \( v_i \) utilizes beyond its endowment is then \( a_{ij} - b_{ij} \) (if the difference is negative, \( v_i \) does not utilize its full endowment).

If \( a_{ij} > b_{ij} \), VMs other than \( v_i \) have to cede some of their endowment of \( r_j \) in order to cover for \( v_i \)'s additional demand. Therefore, the amount utilized additionally by \( v_i \) is added to the greediness of \( v_i \).

If \( a_{ij} = b_{ij} \), \( v_i \) exactly utilizes its endowment, wherefore it does not change \( v_i \)'s greediness. In particular, if \( a_{ij} = b_{ij} \) for all \( j \in \{1, 2, \ldots, n\} \), \( v_i \)'s greediness is zero.
If \( a_{ij} < b_{ij} \), \( v_i \)'s cession of \( r_j \) is credited to \( v_i \), i.e., subtracted from \( v_i \)'s greediness, to the extent that other VMs profit from this cession, which is the case, when they utilize \( r_j \) beyond their endowment. This extent not only depends on how much of \( r_j \) is utilized beyond endowment by other VMs but also on how much is ceded of \( r_j \) by other VMs. Therefore, the ratio of what is ceded of \( r_j \) to what is consumed beyond endowment of \( r_j \) is the credit factor for the cession of \( r_j \). To capture this notion formally, define \( \alpha(r_j) \) as the sum of what VMs utilize beyond their endowment of \( r_j \), i.e.,

\[
\alpha(r_j) := \sum_{i=1}^{m} \max \left(0, a_{ij} - b_{ij}\right)
\]

and \( \beta(r_j) \) as the sum of what VMs cede of \( r_j \), i.e.,

\[
\beta(r_j) := -1 \cdot \sum_{i=1}^{m} \min \left(0, a_{ij} - b_{ij}\right).
\]

Multiplying the amount that \( v_i \) cedes of \( r_j \) with factor \( \alpha(r_j) / \beta(r_j) \) then implements the considerations above. Note that \( \alpha(r_j) / \beta(r_j) \) differs from the resource load.

The greediness of \( v_i \) is defined as

\[
g(v_i) = \sum_{j=1}^{n} o(i,j) \div q_j,
\]

where \( q \in \mathbb{R}_{\geq 0}^{n} \) is a vector to normalize resource units and \( o(i,j) \) is the offset for \( v_i \)'s consumption of \( r_j \) and defined as

\[
o(i,j) := \begin{cases} a_{ij} - b_{ij} & \text{if } a_{ij} \geq b_{ij}, \\ \min \left(1, \frac{\alpha(r_j)}{\beta(r_j)} \right) \cdot (a_{ij} - b_{ij}) & \text{else.} \end{cases}
\]

Note that, if \( \beta(r_j)=0 \), the else-part of Equation above is never reached and that

The minimization in the else-part ensures that at most as much credit is given as is actually ceded of a resource. In particular, if the minimization was not in place, and endowments of VMs would not add up to the physical supply of a congested resource \( r \), the release of \( r \) would be credited with a factor greater than 1.

The subsequent equation defines the greediness of a customer \( c \), where the subtrahend reduces to a scalar to credit \( c \)'s unused quota.

\[
g(c) := \sum_{v \in \text{virt}(c)} g(v) - q \times \left(\text{quot}(c) - \sum_{v \in \text{virt}(c)} \text{res}(v)\right)^T
\]

**Redelegation**

Since resource overload, i.e., resource scarcity, is isolated to PMs, resource redelegations are determined locally. If a PM/hypervisor detects resource overload, it redistributes resources based on the received greediness vector. The redistribution process depends on the overloaded resource. In particular, some resources can be easier redelegated than others: non-persistent resources, such as CPU and bandwidth, can be easily redelegated by changing access rates, while persistent resources, such as disk space and RAM, require a more elaborated approach. For example, once RAM is allocated it cannot easily
be removed from a VM, as it would likely crash the VM. Also for mass storage a reallocation is not trivial, as the stored content must not be deleted, when the respective disk space is reallocated. Thus, content stored on disk space, which is to be reallocated, should be moved to larger storage devices with worse access speed. The movement of data between storage devices with different qualities is in particular investigated in “UC2: Automated storage tiering in the federated clouds”, wherefore the integration potential of mechanisms developed within in the framework of UC 2 and the MRA mechanism will be investigated.

Reallocation Target

The target allocation of a scarce resource of the reallocation process is calculated as follows.

Let $S$ be the supply of an overloaded resource $r$, i.e., $S$ is what is physically available of $r$ before any allocation. Assume the PM hosts $m$ VMs $\{v_1, v_2, \ldots, v_m\}$. Let $g \in \mathbb{R}^m$, where $g_i$ is the greediness of the customer owning $v_i$, if $v_i$ receives all non-bottleneck resources as requested (these values can be reconstructed, by deducting from the original greediness vector what $v_i$ currently receives of $r$). If $r$ is overloaded, it cannot be known, how much VMs would utilize of $r$, if it would be abundant. Thus, let $w \in \mathbb{R}_{>0}^m$, where $w_i$ is the estimate of how much $v_i$ would utilize of $r$, if $r$ would be abundant. Based on this data a vector $a \in \mathbb{R}_{\geq 0}^m$ is calculated, which describes the fairest allocation of the overloaded resource according to the greediness metric, i.e., VM $v_i$ should receive amount $a_i$ of $r$. $a$ is determined by solving the following optimization problem (which can be done in $O(n \log n)$ time.

\[
\max \sum_{a_i \in a} a_i \tag{1}
\]

\[
\sum_{a_i \in a} a_i \leq S \tag{2}
\]

\[
\forall a_i \in a: a_i \leq w_i \tag{3}
\]

\[
\forall a_i \in a: a_i > 0 \implies \forall j: a_j + g_j < a_i + g_i \implies a_j = w_j \tag{4}
\]

(1) postulates that all available resources are distributed. (2) ensures that resources are not overbooked. (3) postulates that no VM gets more resources than requested. (4) ensure that $r$ is given to those VMs first, which have the lowest greediness. Since this will increase their greediness, at some point they reach the same greediness as other VMs, which also then have to get $r$ allocated, and so forth. In other words, $r$ is never given to a VM if there is another VM with currently lower greediness requesting $r$.

The enforcement of the returned allocation has to be implemented by different hypervisor resource shaping calls depending on the overloaded resources. In particular, practical constraints (which depend on many factors) may not allow reallocating resources exactly as outputted by the function (e.g., because real resources can only be allocated with a certain granularity), wherefore the return value of function is rather a target value for the allocation of the scarce resource.
5.3.2 Parameters & metrics

The following three parameters are to be determined with respect to the reallocation of resources.

The first parameter is the **overload threshold**, which is in particular relevant for non-persistent resources: while persistent resources can be considered overloaded, when they are completely booked, overload is harder to determine for non-persistent resources, as overload may occur temporarily (and thus negatively affect VMs), while the average utilization of the resources is not 100%.

The second parameter is an **estimation factor for the additional need of a VM of a scarce resource**. More precisely, when overload is determined, VMs, which fully deploy their current share, are likely to utilize more of \( r \), if given the possibility. Since calculating the target allocation requires a vector \( w \), which describes how much each VM would utilize of \( r \), if given the possibility, \( w \) needs to be estimated. It is planned to arrive at this estimation by multiplying the current utilization with a factor between 1 and 2, which has to be determined in dependency on the resource type.

The third parameter to be determined, is a **reallocation threshold for the deviation of the actual allocation to the target allocation of \( r \)**. In particular, the reallocation of a resource will come with a certain cost (in particular for persistent resources), wherefore a reallocation is not meaningful, if the deviation of actual and most fair allocation is small. Thus, the deviation threshold of actual and most fair allocation that triggers a reallocation has to be determined. This threshold will likely have to depend also on the type of considered resource.

To assess the performance of the MRA mechanism three metrics will be used:

**Degree of fairness in the cloud/federation.** The degree of fairness can be quantified by mapping the (global) greediness vector to a fairness index with various single-resource fairness metrics, such as Jain’s index. However, the greediness of customers with satisfied demands (i.e., who’s VMs all receive their requested resources) and non-positive greediness should not be fed to the fairness metric, since these are exactly those customers who will not contest their share and who’s share cannot be contested by others (as they cede more to others than they receive). Instead, their greediness should be incorporated afterwards into the greediness index, as percentage of perfectly fair treated customers. This exclusion is important, because customers with little demands are likely to be satisfied and have a low greediness, wherefore measured unfairness would inevitably increase, if they were not treated separately.

**Cost and performance loss due to resource reallocation.** The reallocation of every resource comes with certain cost and possibly temporary performance loss. In particular, the reallocation of persistent resources may be costly and impede performance, as stored content has to be transferred, which increases system load. To measure these effects, each reallocation operation should be associated to cost and performance loss. Subsequently it should be counted how often each of these operations is carried out for exemplary runs of the MRA mechanism, to conclude on the increase in cost and performance loss.

**Performance Isolation.** As stated in the OpenStack documentation [61], VMs that heavily consume resources can negatively impact other VMs. Since such heavy behavior is classified as greedy by the greediness metric defined in Section X, the MRA mechanism should reallocate resources from heavy VMs to moderate VM, which should result in a degree of performance isolation for the moderate VMs. The degree of performance
isolation should be measured by the performance decrease a moderate VM exhibits, when heavy VMs are started on its PM.

The evaluation will be conducted by the model presented in Section 4.5. Evaluation results will be provided in Deliverable D2.5.

5.4 DTM++

There is strong synergy between ICC and DTM, due to the fact that both optimize transit link traffic but in different ways and by different means. In particular, ICC deals with the reduction of the 95th percentile over a single transit link by optimizing the traffic transmission rates in time, possibly delaying some traffic, while DTM optimizes the distribution of traffic over the multiple transit links available so that the cost of the domain interdomain traffic is minimized. The superposition of the two mechanisms allows the combination to be applicable in both single- and multi-homed ISPs and optimizing the traffic link distribution over both space (DTM) and time (ICC). This creates a powerful new mechanism, henceforth referred to as DTM++. The rationale behind this new mechanism and its constituent mechanisms, as well as the synergetic effects of combining ICC and DTM into DTM++ are discussed in the remainder of this section.

5.4.1 Intelligence

The main rationale of the ICC mechanism is to utilize the transit link traffic in an optimal way given the 95th percentile charge so as to reduce the transit charge by means of controlling the rate of the portion of the traffic that has been marked as time-shiftable (see Appendix 13.3). This is achieved by means of controlling the transmission rate of the transit link(s). In particular, a portion of traffic is a priori marked as being time-shiftable traffic so that it can be delayed when the transit link is heavily loaded and is sent later, i.e. when the transit link utilization is lower than a target value $C_{target}$.

The mechanism also has built-in support for cloud services, allowing the cloud and network layer to exchange information of the current status of the network and cloud infrastructure and thus enable educated decisions benefiting both the cloud and network infrastructure (e.g. perform a bulk data transfer for back up reasons to the cheapest destination Data Center (DC) when the network is lightly loaded). Hence, the ICC mechanism operates individually over one or a set of transit links of the ISP without coordination amongst them and involves two layers: the network layer and the cloud layer.

DTM rationale is a bit different but complementary. DTM focuses on the management of the traffic transferred between DCs or clouds over DC links supervised by ISPs assuming multi-homed ISPs in order to apply the 95th percentile rationale by “load-balancing” optimally the portion of the traffic that is to be sent/received per link: In this approach an important ISP’s network point is a Data Center Attachment point (DA). The DA is a point (router or switch) where DC inbound and outbound traffic is passed to an ISP network. In the current specification of the DTM mechanism, DTM is designed to improve the traffic distribution on inter domain links thus this mechanism affects inter DC or inter cloud traffic between DCs or clouds located in different ISP’s domains.

In order to simplify the presentation and without loss of generality we restrict attention to the case where a cloud (or DC) delivers data to many clouds (or DCs). As already explained in the DTM spec, under the DTM approach the S-boxes in each receiving ISPs (ISP-A and ISP-B) sends its own inbound compensation vectors to the S-box in ISP-C,
these vectors are different for each ISP to ISP connection. They are selectively used for distributing flows to specific tunnels related to respective inter-domain links. The S-box in the ISP-C establishes one common outbound compensation vector for all connections to other ISPs.

In the next step, the pair of compensation vectors (inbound and outbound) is applied by DTM mechanism for the communication between DCs located in the respective ISP’s domains. In Figure 5-16, we have two communication relations ISP-C to ISP-A and ISP-C to ISP-B. The inbound compensation vector delivered by the S-box-A is used for distributing traffic to tunnels: T-A-C-1, T-A-C-2, T-A-C-3 and T-A-C-4. The S-box-B sends the inbound compensation vector for spreading traffic amongst all tunnels connecting the ISP-C with ISP-B.

Figure 5-16: Reminder on DTM operation over the inter-DC bulk data transfer scenario.

Combining DTM and ICC can be beneficial in following cases:

- apply ICC in cases where DTM is not applicable (e.g. single-homed ISPs)
- use DTM and ICC sequentially when both applicable in order to be able to optimize the traffic distribution over both the space (multiple transit links, DTM rationale) and time (delaying traffic when network is heavily loaded, ICC rationale) axes, as depicted in Figure 5-17.

Clearly both mechanisms need to have the same goal: We define this to be the 95th percentile charge reduction at a maximum cost of $C_{target}$. Also, we need to define the time scale upon which the DTM-ICC mechanism operates: we prescribe that both mechanisms should operate on an epoch of 30secs, thus $\Delta t=30$ secs (for DTM) and $y=10$ secs (for ICC). Also the scope of the combined mechanism, i.e. the traffic that is to be handled from the mechanism, must also be clearly marked (see below).

The DTM-ICC combined operation rationale is based on executing the following list of steps, already specified in the individual specifications of the two mechanisms:
Step 1: Mark the traffic that is to be handled by the DTM-ICC combined mechanism. We argue that the most straightforward way of marking for the DTM-ICC mechanism would be to use the method of delivery over a different Point of Interconnect (PoI), as already specified in the ICC specification. In practice, a separate GRE tunnel could be set up for the delivery of the traffic that is to be optimized.

Step 2: DTM operation for optimal vector computation and splitting of manageable traffic to the two (or more) transit links in an optimal way.

Step 3: ICC operation to enforce $C_{target}$. In particular, the ICC mechanism will take as input the traffic vector decided by DTM and check it individually per transit link for conformance with the $C_{target}$ goal and the respective thresholds that are to be met for the traffic within each epoch. In the case of a violation of these thresholds, e.g., due to the fact that both transit links are heavily utilized and thus shifting of traffic among links does not suffice to reach the $C_{target}$ goal, the ICC TM mechanism will delay a portion of this traffic in time. Thus, the combined DTM-ICC operation allows the optimization of traffic in both the space and time axis.

Note that in the case of multiple DCs in the ISP domain this additional delay incurred will be applied to the aggregate traffic crossing each transit link, thus it will inevitably affect a portion of the traffic of each DC customer. In this case, the DTM operation (Step 2) will take place in all customer DC DAs, while ICC operation (Step 3) will be performed on the aggregate traffic flow over each transit link.

This DTM-ICC combined mechanism described so far is a combination of the DTM mechanism and the ICC network layer traffic management mechanism. In order to be able to support both federated and non-federated bulk data transfers and allow for cross-layer...
optimizations between the cloud and the network layer, we specify that the cloud layer of the ICC mechanism can be invoked in the cases of a) federation b) independent bulk data transfers where there can be multiple potential destinations of traffic. Thus, we need to add to the specification an additional Step 0:

**Step 0:** Cloud-network layer communication and destination datacenter DC* selection. This step is readily available from the ICC specification, we briefly summarize below:

In the case of independent, non-federated data transfers, the source datacenter queries all potential destinations <DCd> for the current network and cloud layer cost metrics and uses them to specify the optimal target DC*. DC* is selected using metric given by the following formula: 

$$DCd^* = \text{argmin} [a * \text{load}(DCd) + b * \text{ec}(DCd) ] , a + b = 1$$

The exact values of the normalized weights a,b depend on the relative weight of the load and energy cost in the decision making process of selecting the most appropriate destination DC. If energy efficiency is not considered, then $a = 1$ and $b = 0$. If both are considered to be of equal value, then $a = b = 0.5$.

The aforementioned procedure is slightly modified in the federated case where the destination DC - if such a selection is possible under the use case, i.e. the vector <DCd> contains more than one element, where the mechanism is applied and not fixed – is again selected by the cloud layer but from the federation cloS.

5.4.2 Evaluation framework and preliminary results

The evaluation framework of the DTM-ICC combined mechanism depends on the progress of its constituent parts, i.e. the ICC and DTM simulation environment. DTM++ is envisioned to evolve in the near future with updated specifications, however some preliminary evaluation can be performed for the case of two transit links even at the point of writing this deliverable. In particular, the DTM++ evaluation framework is attained by combining that of DTM and ICC in a sequential fashion and over the same time scale of 30sec samples. Hence, the DTM simulations are run first so as to generate the vectors of traffic that are to be sent to each of the two transit links. Then, these values are stored in files, which are provided as input traces to the ICC simulation environment. Evaluation is currently in progress and extensive results will be provided in the forthcoming D2.5, however some preliminary results are provided below:

We consider the case of two transit links where traffic is initially managed a la DTM for a week. DTM manages the traffic optimally according to the reference vector. This allows for the efficient utilization of the two links myopically, thus responding to the traffic vector and deciding on the transit link to be used. The resulting values for the 95th percentile in the two links are (1402461191, 868336997) Bytes, which in terms of rates is (4674870, 2894457) bps. Then we apply ICC and try to see if we can further reduce the 95th percentile of the two links. Given that there is a substantial amount of manageable (time-shiftable) traffic in both links (typically at least 30% of the total traffic) we decide to set the Ctarget value for both links (i.e. the target 95th percentile) to be the 80% of the maximum trace value. This is not necessarily the “optimal” target value by any definition, just an attempt to see if we can indeed achieve a reduction of the 95th percentile of the two links; sensitivity analysis over multiple Ctarget values is left for D2.5.

For both cases ICC attains the target value of Ctarget. The respective values for Link 1 and Link 2 are (1.3143E9, 7.731E8) bytes, i.e. (4381000, 2577000) bps. Note that these values are considerably lower compared to those attained with pure DTM, thus demonstrating the benefits of DTM++. Figure 5-18 and Figure 5-19. A comparison with the input trace of DTM (marked as BE in the plots) is also depicted. ICC operates over y=10
epochs, i.e. traffic shaping is performed per 30sec with different thresholds over the rate for the first 9 of 10 total epochs per 5-min interval, hence the shape of the traffic when shaping is performed.

![Figure 5-18: Traffic shaping in Link 1 performed by ICC after DTM is run for a trace of 1 week duration.](image)

Also note that further gains can be attained since the variability of the traffic is quite high and ICC_FA does not even to send with maximum rate at the last intervals of the billing period (here week). Obviously the amount of bytes sent both under DTM and DTM-ICC is the same, hence no Bytes were queued when the experiment terminated.
Figure 5-20: Distribution of observed 95th percentile sample pairs, measured during the billing period of 7 days: experiment with DTM (blue) and with DTM++ (green).

Figure 5-20 presents a distribution of sample pairs (similarly to the DTM evaluation presented in Section 5.1.4). The distribution of samples for DTM is the same as shown on Figure 5-6(b). Green dots on Figure 5-20 shows that further improvement is possible if ICC mechanism is used to lower the highest 95th percentile samples of the traffic. It can be noticed that only higher samples are affected by ICC. They are decreased and consequently the inter-domain traffic cost (based on 95th percentile tariff) is lower for DTM++ than DTM. The height of low size samples is not changed (green dots cover blue ones).

It is important to stress that these initial results, demonstrate the potential of DTM++ to make the most out of the intelligence of its two constituent parts (ICC and DTM) and further lower the 95th percentile charge. A full presentation of simulation results along with the sensitivity analysis over the parameters and metrics, such as the $C_{target}$ and the attained delay penalty, is beyond the scope of this deliverable and is left for D2.5.
6 Specification of Mechanisms for EFS

In this chapter, traffic management mechanisms that utilize end-user resources or information provided by end-user applications, like social networks, to address the EFS are specified. The obtained information and resources are used to manage traffic locally at the end-user premises but they can also be used by ISPs to optimize services and traffic management.

Below, five basic mechanisms and synergetic solutions in the end-user-focused scenario are specified. Parameters and metrics for performance evaluation of the mechanisms are provided. The evaluation framework for each mechanism is described. The results derived from the evaluation frameworks are presented and analysed to evaluate the performance of each mechanism, its benefit for end-users and operators and its capability to manage and save traffic.

6.1 RB-HORST

The RB-HORST mechanism is a combination of the Replicating Balanced (RB) - Tracker and Home Router Sharing based on Trust (HORST) mechanisms. The RB-HORST mechanism eases data offloading to WiFi by sharing WiFi networks among trusted friends. Moreover, it places the content near to the end user such that users can access it with less delay and higher speed, which generally results in a higher QoE. An overlay based on RB-Tracker is formed to predict content demand and to prefetch and deliver content accordingly. RB-HORST provides incentives for all involved stakeholders, namely end users, ISPs, Data Center Operators, and Cloud Service Providers. ISPs benefit from RB-HORST, as the mechanism introduces content placement and caching close to the end users. Together with the selection of the best content resource to download, RB-HORST reduces, as a side-effect, inter-domain traffic and energy costs in the network. Moreover, RB-HORST uses end user devices for content placement and caching which lessens the need for ISP caches and reduces the load on servers of Cloud Operators and Application Providers. This leads to fewer expenses for hardware and energy for those stakeholders.

Mobile end users have increased WiFi access as RB-HORST eases WiFi sharing among trusted friends. Thus, RB-HORST users can benefit from increased throughput over WiFi access points with capacity, which improves the perceived quality of Internet services and reduces energy consumption by shortening transmission delays. Sharing WiFi for ubiquitous Internet access has already been tackled by commercial services as well as research work. Fon [103] already started to build a WiFi-sharing community in 2006 by offering a home router with a public and a private encrypted SSID. The public WiFi could then be accessed by everybody who joined the community. Similar approaches are Karma [104] which adds a subtle social layer to WiFi sharing, and WeFi [105] and Boingo [106] which provide hotspot databases. Moreover, the social prediction and respective content placement improves the QoE of all end users for download-intensive services (e.g., YouTube video streaming).

The RB-HORST mechanism runs on a user-controlled Nano Data Center (uNaDa) which is established on the end users’ home router. Thus, in order to participate, a user needs only a flat rate Internet access at home, he has to install the RB-HORST firmware to his home router, and he needs a Facebook account. On Facebook, he has to install a Facebook app which is part of the RB-HORST mechanism and is responsible for user
management and sharing of WiFi credentials. Furthermore, part of the prefetching prediction is based on data available from Facebook.

The RB-HORST firmware establishes two separate WiFi (a private and a shared WiFi) and enables mobile data offloading by broadcasting a special SSID (of the shared WiFi) which can be recognized by a mobile device. Through the Facebook app, a user can request access to the shared WiFi which triggers a decision if the mobile user can be trusted. In case the user is found trustworthy he is allowed to connect to the shared SSID and he can access the Internet.

RB-HORST powered uNaDas interconnect in a P2P style location-aware overlay network, which allows for content storage and delivery. Existing overlay networks do not provide the necessary locality information required to reduce inter-domain traffic. However, file transfer mechanisms from BitTorrent can be partly reused. RB-HORST intercepts content requests (i.e., YouTube video requests), similar to a transparent proxy, and queries the overlay to find a close uNaDa (in terms of AS hops) which offers the content. Alternatively, the content could also be provided by any other peer, e.g. BitTorrent or BTLive could be applied here. If no close uNaDa is found to provide the content it is loaded from the original source, e.g., the YouTube datacenter provided by the DNS. The content is cached and distributed among all participating RB-HORST devices based on the content ranking in the overlay. The interception of content requests for YouTube is still a critical open issue of the RB-HORST approach as it is not clear at this stage, if that is both legally allowed and technically doable. The legal constraints and exploitation and business potential of the proposed approach need to be reviewed again at a later stage, e.g., YouTube could be interested in running the uNaDa, or the approach could be used for any other content (BitTorrent, Netflix, etc.)

Every uNaDa monitors the Facebook news feed of his owner with a dedicated client and the requested content in the system. It ranks the content based on history, popularity, age, distance, and social information. RB-HORST predicts the content which will most likely be requested, i.e., the contents with the highest ranks. If a predicted content is not yet available in the local cache, it will be prefetched during an off-peak period (e.g. at night/early morning). The reason for doing so is to avoid volume peaks in the operator network. If the user is connected to a friend’s home router, a prefetch command is sent to the RB-HORST system on the friend’s router. Since the prefetched content can be offered to other peers, caching efficiency is expected locally within a single ISPs network. However, content items to be prefetched have to be selected carefully, in order to avoid additional traffic without any gain. Also, for prefetching, RB-HORST chooses the best source based on overlay information, and fetches the desired content. In regular intervals, HORST checks if the content in his own local cache is still relevant (either for local consumption or as a source in the content delivery network) and decides whether to keep or replace it.

RB-HORST addresses the three end-user focused use cases: mobile data offloading, service and content placement, and locality in uNaDas, while initially its focus is mainly on the service and content placement use case [6].

In the following we make some basic assumptions and set some requirements. We assume that users have a flat rate Internet plan and that users can install RB-HORST on their uNaDa. It is required that a user needs RB-HORST installed on his uNaDa and that a user needs the RB-HORST Facebook App in his Facebook account.
6.1.1 Intelligence

Overlay

RB-HORST enabled uNaDas organize themselves in an overlay network. The basis of the overlay is a distributed hash table (DHT) which is used as register for all uNaDas and all contents cached in uNaDas. Therefore, the DHT is used to find addresses of uNaDas and to find providers for a resource. Once a uNaDa has at least one provider for a resource it asks these providers for more providers. To support locality those providers return providers which are as few as possible AS hops away.

Instead of just loading resources from the cloud or a data center, RB-HORST considers the location of providers for a resource. To reduce inter-domain traffic a provider that is as few AS hops away as possible is used. Therefore, an additional step in the process of finding providers, called tracking, is required.

The tracking process determines how providers for resources are found. A new node joining the system will connect to the DHT and query it for the Resource ID. The DHT returns addresses of nodes providing this resource. All nodes check periodically if there are still enough entries in the DHT; if not, they add their address.

Content prediction & placement

RB-HORST predicts content that will be consumed by clients of a uNaDa. The prediction consists of two separate processes. The first is based on social information delivered by the Facebook App and the second is based on the overlay. The social predictions consider, among others, content shared by friends, the location of a user and age of a video. The overlay based prediction uses information on cached / watched videos from other RB-HORST enabled uNaDas.

Both prediction processes run in parallel and each has its own limit of cache size. An additional mechanism must be implemented that ensures, that content is not prefetched twice. Whether the two predictions can be integrated into one has to be investigated at a later stage.

The update of the cache runs in parallel, for each connected user and for the overlay based prediction. Each process has a separate virtual cache size, whereas the cache is commonly available for all users. Each process should be able to check if a certain video is already present in the cache, so it must not be downloaded.

Social content demand prediction

Periodically the uNaDa access the news feed of the owner and extracts all video URLs. The URLs are ranked based on five dimensions:

- History: number of accesses to the video by the user
- Popularity: number of global accesses to the video
- Age: publish date of the video content
- Distance: distance of the closest content location
- Social: number of users who posted the video in the Facebook news feed

If the user is currently at this uNaDa, the most relevant URLs are downloaded and stored in the cache. If user is registered at another uNaDa, a message containing the most relevant URLs for this user is sent to the other uNaDa.
Overlay based prediction

Periodically an RB-Tracker node contacts all the nodes known and asks for a list of resources available from each. The additional overhead introduced is minimal since the lists have to be exchanged infrequently, i.e. once an hour, which would be reasonable, if we assume that the interest of a user does not change within the day. The requests are sent periodically and in parallel for each known node. With the responses a ranked list of all possible candidates for caching can be calculated. A simple algorithm that can be used to calculate the ranking is provided.

Mobile data offloading

Currently, RB-HORST offers mobile device offloading by sharing WiFi access among trusting friends. Each uNaDa establishes two WiFi networks - a private and a shared WiFi. The private WiFi can only be accessed by the owner with his private credentials. The SSID of the shared WiFi has a specific format such that it can be recognized as a RB-HORST SSID (e.g., RB-HORST_568cad59ef98574b06 (hash of MAC address)). This generated SSID and the generated credentials are stored in the Facebook app.

When a visiting user discovers a RB-HORST SSID, it can request access via the Facebook app. The trust between the visiting user and the owner is determined and if there is a trust relationship the visiting user receives the credentials of the shared WiFi network. The visiting user and his future content request are taken into account by prefetching content for him. Therefore, his own RB-HORST node sends the prediction for the visiting user to the visited RB-HORST node where the relevant content is considered for caching.

6.1.2 Parameters & metrics

In the following specification items are generally videos or video chunks.

Parameters

The parameters considered for the content delivery simulation framework for RB-HORST determine

- the resource distribution and size,
- the caching and content placement strategy,
- the resource selection strategy,
- the social network of users,
- and the content demand.

Later on we may further consider

- the AS-Topology,
- the video bitrate and chunk-size distribution,
- and the application and QoE.

In the following we briefly describe each of the parameter sets and provide models.

The resource distribution and size determines how video streaming sources are distributed among autonomous systems. The number and size of autonomous systems is specified. The size of an autonomous system is given by the number of end-users located in it. In literature, the distribution of end-users on ASes is characterized as heterogeneous
We use a geometric distribution as a basic model for number of end-users in the ASes. A more detailed model is developed using the Internet Census dataset, c.f. Section 4.3.

Video streaming sources can be a) datacentres of the content provider, b) edge caches of the content provider, c) caches hosted by the ISP, d) home router / UNaDas, or e) end-user devices. For each video streaming source the AS-location and its capacity is specified. The capacity is given by the number of items that can be cached.

The size of the item catalogue is also specified in this parameter set.

The caching and content placement strategy determines in which video streaming source which video item is placed and when. The content placement strategy is defined by the caching strategies of the individual caches. In a distributed approach each cache decides based on the information it has which items to cache. Since RB-HORST can have global knowledge of the item demand, we can also think about content placement strategies optimized with global knowledge. Thus, the availability of items in ASes might be increased, for example. In the following, we distinguish between caching strategies that need global information for their decisions.

Each caching strategy is further defined by its specific parameters (Table 6-1).

Table 6-1: Caching strategies employed by RB-HORST.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRU</td>
<td>local</td>
</tr>
<tr>
<td>RANDOM</td>
<td>local</td>
</tr>
<tr>
<td>SLWND</td>
<td>local</td>
</tr>
<tr>
<td>GEOFADING</td>
<td>local</td>
</tr>
<tr>
<td>SGLRU</td>
<td>local</td>
</tr>
<tr>
<td>LRUAS</td>
<td>AS</td>
</tr>
<tr>
<td>LFU</td>
<td>global</td>
</tr>
</tbody>
</table>

The resource selection strategy determines from which cache instance an item is streamed when requested. The simplest resource selection just selects a random resource. We also implemented local resource selection, which tries to save inter-domain traffic by prioritizing caches in the order: home router / UNaDa in the same AS, ISP managed cache in the same AS, edge cache of content provider, datacentre of content provider.

Further resource selection strategies that are not yet implemented could for example consider load balancing of the request based on the capacity of the caches.

The social network of users determines the friendship relationships between users. A basic model only defines the number of users in the system. The number of friends of the user can be modelled by a power-law or geometric distribution. A more detailed model specifies the friendship graph which consists of a node for each user and edges between users with friend relationships. Friendship graphs have typical properties, such as a heavy-tailed in and out degree distribution. In literature are different models for generating...
graphs with these properties. A model used to generate social network graphs with varying size and density is the forest fire model [63].

We further specify the wall size as parameter that represents the wall of social network platforms. The wall is updated in sharing events. Videos which are on the wall of a user are watched with high probability.

Categories are defined by specifying the probability that a user is interested in a particular category.

The content demand determines the request rates of the video items. Different demand models are implemented in the simulation, that reach from basic models that only consider the popularity distribution of the items to detailed models that consider temporal, spatial and social dynamics.

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZIPF</td>
<td>Basic model</td>
</tr>
<tr>
<td>WALL</td>
<td>Social dynamics</td>
</tr>
<tr>
<td>YTSTATS</td>
<td>Social dynamics</td>
</tr>
<tr>
<td>LI3</td>
<td>Social dynamics</td>
</tr>
<tr>
<td>SNM</td>
<td>Temporal dynamics</td>
</tr>
</tbody>
</table>

The arrival process of video requests is specified by the inter-arrival time of video requests. The request rate depends on the time of day and is generally lower at night. The day is divided in short time slots, where the arrival rate does not change significantly, so that the arrival process can be assumed as quasi stationary. In these time slots the arrival process is modelled as Poisson-process. The parameter lambda of the arrival process depends on the popularity of the item and the time of day.

The probability of sharing a watched video is given by the sharing probability. Finally, Simulation parameters are specified that define the random number seed, the simulation time and the parameter study.

Performance metrics

To assess the performance of RB-Horst several metrics can be used. A list of these metrics is given:

- Prediction accuracy
- Cache hit ratio
- Saved inter-AS traffic
- Offloaded traffic (from cloud to uNaDa, from 3G to WiFi)
- Overhead traffic generated by mechanism

6.1.3 Evaluation framework

The evaluation of RB-HORST is conducted in two separate simulation frameworks. The performance of the content placement and resource selection strategies are evaluated in the content delivery simulation framework considering cache hit rates and inter-AS traffic. The WiFi offloading simulation framework is used to evaluate the offloaded traffic.
**Content delivery simulation framework**

The content delivery simulation framework is implemented in MatLab. The simulation is event-based. There are two major events. First there is the WATCH event, which is processed when a user watches or consumes a video item. Second there is a SHARE event, which simulates a sharing action of a user, where the video is posted on the user’s wall in the social network.

Figure 6-1 shows the process diagram of a WATCH event. The process of a WATCH event starts by selecting a video dependent on the demand model specified in the parameters. A video with video identifier $vid$ is returned. In the next step a cache or datacentre is selected that holds the item with $vid$. The download of the item from this resource is recorded in the statistics. The cache identifier $cid$ is returned and the cached items are updated according to the caching strategy specified in the parameters. The user then decides to share $vid$ on his wall with probability $p_{share}$. Thus a random number is drawn, and in case it doesn't exceed $p_{share}$ a SHARE event is added. Finally, the next WATCH event is added after a randomly drawn inter-arrival time, which depends on the time of day.
A SHARE-event puts a given $vid$, or a random video according to the user’s interest on top of the wall of the user’s friends. The user’s friends are determined by the social graph.

The simulation is initialized with a WATCH event for each user.

**WiFi offloading simulation framework**

The WiFi offloading simulation framework is used to evaluate the offloading potential of RB-HORST. The first version of the framework estimates the static WiFi offloading potential on a city area based on the number of shared WiFi access points. To estimate the static WiFi offloading potential, the WiFi access point locations and the locations of end-users are needed. For that purposes we use two datasets.

The WiFi access point locations in the city area of Darmstadt are provided in [64]. The dataset was obtained by sensing the WiFi access point SSIDs. The WiFi access point locations where interpolated by combining all locations where the signal of the WiFi was received.

The probability of end-users to be at a specific location in the Darmstadt city area is

---

**Figure 6-1: Process-diagram of a WATCH-event.**

```plaintext
WATCH

getVideo() → selectResource()

rand() < $p_{share}$

Yes?

updateCache()

No?

addEvent (SHARE) → addEvent (WATCH)
```
derived by a street map of Darmstadt from OpenStreetMaps [65]. The street map contains waypoints that are interconnected to define streets, or that describe buildings, facilities, local businesses or sights. The waypoints are all set up by users contributing to the OpenStreetMap platform. In that way, the waypoint locations provide a good model for end-user location probabilities.

Figure 6-2: City area of Darmstadt with WiFi access point locations (red) and end-user locations (blue).

Figure 6-2 shows the street map of the city area of Darmstadt with WiFi access point locations and end-user locations. End-user locations are dense in the city centre. The sending range of a WiFi access point is assumed to be constant and is specified as a simulation parameter. We further assume that end-users do not move and change their location during requests. The WiFi sharing probability is also set as parameter. In a Monte-Carlo simulation the WiFi access points, which are shared and the end-user locations during offloading requests are randomly determined. If an end-user is in range of a shared WiFi access point, his request can be offloaded. The offloading potential is determined by the ratio of offloaded requests to total requests.

6.1.4 Evaluation results

The evaluation results for RB-HORST are presented separately for the two simulation framework for content delivery and WiFi offloading according to the description above.

Content delivery simulation results

In the first evaluation of the content delivery simulation framework we evaluate a tiered caching architecture with resource locations at three different tiers, including the main data center of the content provider, content delivery network (CDN) caches and end user equipment. Table 6-3 shows the default parameters of the content delivery simulation. The number of different content items that can be downloaded or streamed from the resources is specified by the catalogues size $N$. Tier-1 resource is the data center of the content provider, where all $N$ content items are stored. Tier-2 resources are edge caches and ISP caches typically organized in a CDN, which are located close to Internet exchange points or in ISP networks. Requests served by ISP or edge caches produce less or no inter-domain costs. We will refer to these caches in the following as ISP caches. The capacity of ISP caches is specified by $C_{ISP}$, the caching strategy of ISP caches is LRU. Each autonomous system hosts an ISP cache. In tier-3 are the caches on shared home routers that have the RB-HORST firmware installed. We will refer to these caches in the following as home routers. The capacity of home routers is 4 content items and their caching strategy is LRU.

The home routers are distributed among 100 autonomous systems according to a power law with slope 1.5 as identified in Section 4.3.1. Each one of 1829425 users hosts a home router. The probability that a user has RB-HORST installed and shares his home router for content delivery is given by $p_{share}$. The probability that a user requests certain content items depends on the content popularity distribution which is specified by the Zipf exponent $\alpha$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6-3: Default parameters of the content delivery simulation framework
A requested item is first looked up in the home router of the user, if it is not found, it is looked up in tier-3 caches in the same autonomous system then in tier-2 caches and finally in the data center of the content provider. The hierarchic caching strategy is leave-copy-down, which means that the video is cached in a certain tier only if it is already available in a higher tier.

In the following we evaluate the performance of the tiered caching architecture. By studying the impact of the home router sharing probability, we identify the amount of home routers, which needs to be shared so that such a mechanism pays off for ISPs. We vary the ISP cache capacity and the Zipf exponent $\alpha$ to investigate to what extent the load on ISP caches can be reduced and how the system performs under various request patterns. For each configuration 5 simulation runs were conducted to achieve significance. The results show the mean value of the 5 runs with 95% confidence intervals.

Figure 6-3 shows the total ISP cache contribution and the mean ISP cache hit rate dependent on the home router sharing probability for different cache capacities. The total ISP cache contribution is the amount of requests served by ISP caches. The total ISP cache contribution decreases with the home router sharing probability. This depends on the fact that more requests are served from home routers with increasing home router sharing probability. The ISP cache contribution also decreases with the ISP cache capacity, which is clear because less content items can be stored on ISP caches, which decreases the probability that a request can be served by an ISP cache. The ISP cache capacity has more influence on the total ISP cache contribution for low sharing probabilities. Hence, especially if the home router sharing probability is low, a larger ISP cache capacity pays off.
Figure 6-3: Total ISP cache contribution and mean ISP cache hit rate dependent on home router sharing probability for different ISP cache capacities.

It is clear that the mean ISP cache hit rate decreases with the ISP cache capacity, since the probability that a requested item is in the cache depends on the number of items stored in the cache. However, it is interesting that for a constant ISP cache capacity the mean ISP cache hit rate also decreases with the home router sharing probability. This depends on the fact that the ISP cache is only requested if an item is not found in a lower tier cache. The total cache capacity of home routers grows with the sharing probability, so that the ISP cache is only requested for rare or emerging new items. The probability that the unpopular items are located in the ISP cache is relatively low itself, which explains the bad ISP cache hit rate. This suggests using a different caching strategy at ISP caches, to store especially rare and emerging times in the cache. Here social information could also help to identify emerging items in advance and to store them on the ISP cache before they are requested frequently.

To investigate how much inter-domain traffic can be saved by the tiered caching architecture we determine the amount of requests that can be served locally. Figure 6-4 shows the total amount of requests served intra-AS dependent on the home router sharing probability for different ISP cache capacities. For low ISP cache capacities the sharing probability has a high impact on inter-AS traffic. This depends on the fact that the ISP cache contributes less to the total autonomous system cache capacity. For higher ISP cache capacities the total amount of requests served locally is less dependent on the home router sharing probability, since the ISP cache can absorb more requests, that could not be served by home routers in tier-3. Hence, a high home router sharing probability is especially important for ISPs with small managed caches to save inter-domain traffic. If the ISP cache capacity holds one percent of the video catalogue size, the more than 60% of requests can be kept locally independent of the home router sharing probability.
Figure 6-4: Total amount of requests served intra-AS dependent on home router sharing probability for different ISP cache capacities.

Figure 6-5 shows the total ISP cache contribution and the mean ISP cache hit rate dependent on the home router sharing probability for different content popularity distributions with Zipf-exponent $\alpha$. As already discussed the ISP cache contribution decreases with the sharing probability. For low sharing probabilities, the popularity distribution has an impact on the ISP cache contribution. For a higher Zipf-exponent $\alpha$ popular items are requested more frequently. Popular items are cached with high probability which increases the mean cache hit rate and the cache contribution for low home router sharing probabilities. The ISP cache can contribute more, if less home routers are shared and fewer requests can be answered directly in tier-3. As stated before, the ISP cache hit rate decreases with sharing probability, since it is only requested for unpopular items that are not found in home routers, which suggests prefetching. Independent of the sharing probability an ISP cache only pays off in a tiered caching architecture for Zipf-exponents $\alpha > 0.6$.

Figure 6-5: Total ISP cache contribution and mean ISP cache hit rate dependent on home router sharing probability for different content popularity distributions with Zipf-exponent $\alpha$. 
Figure 6-6: Total amount of requests served intra-AS dependent on home router sharing probability for different content popularity distributions with Zipf-exponent $\alpha$.

This also suggests investigating social and temporal locality in requests, since it probably increases the cache efficiency.

Figure 6-6 shows the total amount of requests served intra-AS dependent on the home router sharing probability for different content popularity distributions with Zipf-exponent $\alpha$. Again the Zipf-exponent has large impact on the total amount of requests served by the caching system locally, which in turn suggests further investigating locality in content demand patterns since it can boost the potential of the system. For high Zipf-exponents more than 60% of the requests can be served locally, independent from the home router sharing probability. As stated before, this depends on the fact that the local ISP cache can absorb requests that cannot be served locally. This effect does not apply to a low Zipf-exponent, where unpopular items that are not even covered by the ISP cache are requested frequently.

Figure 6-7 shows the ISP cache hit rate for the five autonomous systems with most end-users dependent on the autonomous system rank by size for different ISP cache capacities. The home router sharing probability is set to 0.01%. The ISP cache hit rate decrease with the autonomous system size. In larger autonomous systems more home routers provide their capacity for caching, so that most of the requests can be served by home routers. The demands which cannot be answered by home routers request items that are very rarely requested, or requested for the first time, and thus are not cached in the ISP cache as well. This explains the low cache hit rate for caches at ISPs with many end users. Hence, if enough users share their home router, the total cache capacity of home routers might be large enough to replace the ISP cache. However, upload bandwidth constraints of home routers have to be considered in a later stage of the simulation. Therefore further measurements are needed to specify the number of simultaneous requests a home router can serve to refine the simulation model.
The results of the content delivery simulation framework show that a tiered caching architecture that can be realized by RB-HORST has a high potential to save expensive inter-domain traffic and to take load of ISP caches. However, the results also show that the potential of the caching system is highly dependent on the popularity distribution of the content catalogue and the home router sharing probability.

This suggest as next steps to evaluate the performance of the caching architecture for content requests patterns that exhibit social and temporal locality, using the models described in Section 4.2. To cope with the poor performance of the ISP cache advanced caching and prefetching strategies have to be applied. To this end the potential of the global or regional knowledge about the content distribution in the overlay of RB-HORST and of social networks has to be leveraged. To this end the autonomous system topology can also be refined to consider regions. Finally, the number of requests simultaneously served by home routers has to be limited to get more significant results.

**WiFi offloading simulation results**

Figure 6-8 shows the results of the static WiFi offloading simulation of a city area. The WiFi offloading potential is depicted for three WiFi access point sending ranges dependent on the WiFi sharing probability. The results show the mean WiFi offloading potential of 100 runs with 95% confidence intervals. If a sending range of only 10m is assumed, the WiFi sharing potential is rather low and increases almost linearly with the WiFi sharing probability. For sending ranges 50m and 100m the WiFi sharing potential grows fast within 0% to 10% WiFi sharing probability. For 10% WiFi sharing probability the offloading potential is higher than 60% for 50m sending range and almost 90% for 100m sending range. The offloading potential increases to more than 70% for 50m and more than 90% for 100m sending range for 20% sharing probability.

That means that, assuming that the WiFi sending range of 50m, a decent WiFi offloading potential is obtained if only 10% percent of WiFi access points in a city area are shared. Hence, to obtain a good WiFi coverage, incentive mechanisms have to be designed, such that at least 10% of WiFi access points are shared.
RB-HORST is a mechanism designed specifically to address on demand video traffic which belongs to the foreground traffic class. RB-HORST could be applied to other content of the same class, for instance on demand music which is also shared on OSNs. However, the benefit of improving on demand music traffic would be low since music traffic is much lower than video.

There are other traffic classes that could possibly be optimized in a RB-HORST fashion. One example is P2P file sharing which is typically background traffic. To predict the content demand of P2P networks new prediction algorithms have to be found which do not rely on OSN, since P2P related links are not shared on OSNs for obvious reasons. The principle of RB-HORST, predicting content consumption and then prefetch it, can most likely be applied to other types of traffic which flows from a data center to an end-user. It remains to be investigated for which types of traffic this principle can be applied and to what extent inter-domain traffic can be reduced.

### 6.2 Socially-aware mechanism for Efficient Content Delivery (SEConD)

SEConD is a novel traffic management mechanism for enhancing content delivery (only video content is considered in this case study) over OSNs, exploiting social relationships derived by the OSN, as well as interest similarities and locality of exchange of OSN content. SEConD has been initially designed for enabling efficient P2P-assisted video delivery and placement. Nevertheless, it can be easily extended to provide the capability to efficiently handle any type of content shared in OSNs. Therefore, SEConD is related to the “service and content placement for users” use case [6]. The main objectives of our mechanism are the improvement of the QoE of OSN users in terms of decreased latency (eliminate stalling events), and the reduction of inter-AS traffic, which we use as a proxy for transit inter-connection costs of the ISP. To this end, our mechanism enables targeted prefetching of video prefixes based on social information, as well as caching and peer-assisted video distribution.

Figure 6-8: WiFi offloading potential for three WiFi access point sending ranges dependent on the WiFi sharing probability.
According to the literature, the video viewing in OSN is driven both by social relationships and interests similarities with respect to the content. Based on these two parameters a user influences certain users more than others. SEConD takes advantage of this information and for each user (source user) assumes a categorization of his friends (at most two social hops) in three categories with respect to level the source user influences them. Therefore, SEConD for each user creates three categories of viewers: followers, non-followers, others. This influence is decreasing moving from followers to others. Note, that these categories do not include all the friends of the user but only those that he affects over a low threshold.

6.2.1 Intelligence

SEConD, exploits the information gathered from OSNs in order to predict demand of a specific user for a specific content. This relies on the fact that some socially connected users to an uploader usually watch the videos he uploads. Also, using the same information creates socially-aware messaging overlays per user, in which each user can send demand indications to other potential viewers for content items of a particular topic, in order to proactively store the first piece (chunk).

SEConD introduces a socially-aware proxy server (SPS) in each AS. The SPS is responsible for the formation (also update) of messaging overlays and also trying to achieve high traffic localization by caching content and by operating as P2P tracker for local content-based P2P overlays. Each SPS creates AS-local P2P swarms for every piece of content that is requested by the end users, within the AS the SPS is established. In a swarm for a specific content, participate the users that have stored the content in their cache, the users are interested to download this content and the SPS under certain conditions that we will describe later. The smallest swarm is the one that contains SPS and a user who is interested in downloading the content, thus we can consider it as a client-server type of delivery.

The basic constituent elements of SEConD are as follows:

- **Socially-aware messaging overlays** are created and used to disseminate alert messages from the uploader of a video, to potential viewers of the video are socially connected to the uploader. Each user is considered to have interest in some video categories, and thus maintains a messaging overlay for each video category of his interest. Each messaging overlay contains, as potential viewers, all the *followers* of the uploader, and only the *non-followers* that are also interested in the respective video category. We choose to not include the other viewers, since the percentage of videos they watch is low. Consequently, each messaging overlay is a bipartite graph, with edges only from the uploader to each potential viewer. Figure 6-9 depicts an overlay constructed for an uploader and for one of his interest categories. When the uploader uploads a video uses the appropriate messaging overlay to alert potential viewers and trigger the pull-based prefetching of the video prefix, for QoE enhancement. Later in this subsection, we demonstrate how our prefetching algorithm taking advantage of messaging overlays. Finally, as the authors observed in [66], the 94% of the videos each user watches are at most from 4 video categories. Thus, we assume that each user is mainly interested in 4 categories and creates overlays only for his top 4 categories of uploaded video.
• A **Social Proxy Server** (SPS) is located within each Autonomous System (AS) in order to localize the traffic generated by the activity of OSN users in this region. Each user of the OSN is considered to be aware of the SPS of his home AS, and thus can request videos and video prefixes from the SPS. Also, the SPS communicates with the servers of the OSN provider and the video streaming platform provider. The SPS can be controlled by the OSN provider, the CDN, or the local ISP. The SPS: i) is responsible for the formation of messaging overlays and to keep them updated through monitoring, ii) responds to users' requests for video prefixes, generated from the messaging overlays, by pushing the requested prefixes, iii) caches the video prefixes to serve future requests, iv) operates as local P2P Orchestrator (e.g. like a BitTorrent tracker) for local content-based swarms formed to perform the video streaming. In order to boost the users’ QoE and localize traffic, the SPS caches each video requested by its local users. Therefore, the SPS is capable of participating in the P2P video dissemination as a resourceful peer [67], when needed, as well as to serve as a proxy video server [68].

• **Local Content-based P2P overlays** are created by the SPS for each video that is requested by the local OSN users. Whenever a user requests to watch a video, the SPS checks if there is already a local P2P swarm for this video. If there is, the SPS adds the requesting user in this swarm and stores the video in his cache (if not already). If there isn't, the SPS creates a swarm for this video and stores the video in his cache. So, in this case the new swarm includes only the user requesting the video and the SPS. Consequently, the users interested in watching a video are added by the SPS in the swarm of this video in order to simultaneously download and share the video (leechers). While, the users who have stored a video in their CPE (customer premise equipment), are added in the corresponding swarm in order to assist the SPS in sharing and traffic localization (seeders). Even in the case where only SPS has stored the video and there is one user that requests it, a swarm with only SPS and this user is created. If two neighboring ASes have a peering agreement, then the P2P overlays for dissemination of videos can be extended accordingly. Although the swarms created can be small, effective resource allocation is achieved.

• In SEConD, we follow a **two-level caching** strategy with caching of video prefixes and videos taking place both in the SPS and in the users’ CPE. Caching prefixes in...
CPE aims to decrease latency by eliminating the video start-up delay (or stall time), while caching the videos themselves in users’ caches aims to assist P2P video sharing. On the other hand, caching prefixes and videos in the SPS is done mainly for traffic localization, as done in certain approaches for P2P traffic; see [67]. Due to the fact that the storage capacity of both the users and the SPS is limited, we need to define caching policies in order to determine which prefix(es) or video(s) to replace, when upon a new arrival the cache is already full. For simplicity, we chose the caching policy described below:

a) **Caching in SPS:** When a new video prefix arrives, if necessary the oldest prefix in the SPS cache is replaced. Also, for each video, the SPS maintains a counter. This is increased whenever the relevant prefix is pushed and decreased when the prefix is used or is eliminated from some user’s CPE. Thus, the higher the counter is the higher the possibility for the SPS to get requests for this video is. Therefore, when a new video arrives, the two oldest videos in the SPS cache are considered and the one with the lowest counter is replaced.

b) **Caching in users CPE:** When a new video prefix arrives it replaces the oldest prefix in the user’s cache. We also follow the same strategy when a video arrives.

Finally, we describe the steps of our socially-aware pull-based prefetching algorithm, as depicted in Figure 6-10: 1) When an uploader (source node) uploads a video, he pushes an alert message to each of the users in the messaging overlay that corresponds to the interest category of this video. 2) After a user receives an alert message, he sends a request to his local SPS asking to receive the prefix of the video referred in the message. 3) When the local SPS receives a prefix request, if this is not already cached, it downloads this prefix from the video server where the video is hosted. 4) The local SPS caches the prefix of the video and pushes it to the user who requested it. Finally, the user stores the prefix of the video in his CPE. In Figure 6-10, we present the steps of the prefetching algorithm for the case where the video is uploaded to a third-party owned server, e.g. a YouTube server.

![Figure 6-10: An example of the prefetching algorithm – The source node shares a video hosted in a third-party owned server. (Sequence numbers arrows.)](image)

### 6.2.2 Parameters & metrics

There are several important parameters for SEConD, need to be defined. First of all, it is important for the mechanism to define certain thresholds between three categories in order
to consider a preferred distribution of friends within these three categories. Therefore, we extend the categorization presented in [66] and we also define explicit threshold-based criteria for each category based on the results obtained from the measurement study they conducted. In particular, for each uploader, we consider as: a) Followers: his 1-hop or 2-hops friends that watch over 80% of the videos he uploads. b) Non-followers: his 1-hop or 2-hops friends that watch less than 80% but more than 30% of the videos he uploads. c) Other viewers: his 1-hop or 2-hops friends that watch less than 30% but more than 20% of the videos he uploads. Note that if a friend of an uploader watches his videos, then this friend watches at least 20% of them [66]. The rest of the friends, including also all those in three or more social hops, are not considered as viewers at all.

Another significant parameter for SEConD is the criterion for SPS participation in P2P swarms to assists in video dissemination. The SPS participate in P2P swarms, if the per user available upload bandwidth in the swarm is below the video bit rate. Of course, there are different strategies and thresholds that could be employed for the contribution of the SPS.

Concerning the size of the SPS cache, we take it for simplicity as proportional to the number of users that are connected to it. In particular, the total memory capacity equals the number of users are connected to the SPS times the total size of one video prefix and one video. In fact, the cache size need not scale linearly for large number of users, therefore other cache dimensioning techniques could be apply for close to optimal cache dimensioning.

Some metrics, which we consider as important for our mechanism's evaluation, are:

Inter/Intra AS traffic: We estimate the traffic generated by video dissemination. Inter-AS links traffic may lead to transit traffic charges, while intra-AS traffic affects the congestion created and can be considered as a proxy of the degradation of users' QoE.

Contribution of server hosting the video: We aim to achieve low contribution of the origin server e.g YouTube to reduce the operational costs for the CDN or OSN video server platform and to avoid a bottleneck that can affect adversely the QoE of users.

Caching accuracy of Social Proxy Server: We estimate the percentage of video prefixes or videos that had already been stored in the cache of the SPS when a user requested it. High caching accuracy of the SPS translates to lower contribution of the origin server where the video is hosted, and thus, to lower inter-AS traffic and potentially to lower transit charges.

Accuracy of prefetching: We estimate the percentage of video prefixes had already been stored in a user's CPE when he requests to watch the corresponding video. High prefetching accuracy is expected to eliminate start up delay for the users and to improve users' QoE.

Useless prefetching: By this term, we refer to the amount of video prefixes pushed and never used by the users that received them. A high number of useless prefixes expected to lead to some QoE deterioration due to bandwidth and local storage consumption.

Redundant prefetching: This occurs when the same prefix is being pushed to a user by multiple sources (friends). High redundant prefetching may lead to some QoE deterioration.
6.2.3 Evaluation framework

In order to evaluate SEConD and its components and to compare it with SocialTube [66], we designed and implemented an evaluation framework to simulate a social environment. Our objective is to model the processes of posting and selection of videos by the users of an OSN. To this end, we employed observations in the literature regarding content disseminated over OSNs, as well as users’ behavior and interactions among them due to video viewing and sharing. Our assumptions are based on measurements studies conducted on real OSN data (i.e. [66]).

We have to define a set of possible interest categories for both users’ interests and videos categories. We adopted 19 interest categories, based on the categorization of [69], where the authors present the various video categories and their distribution in YouTube. We designed a discrete-time event-driven simulation framework, where time is slotted in slots of 20 minutes. We assume that every day each user is active on the Internet for 7 20-minute slots, regardless his daily activity in Facebook. On the other hand, only 66% of users are active a given day in the OSN, and each such user is active only for one 20-minute slot within this day. In order to select the 20-minute slot a user is active in the OSN, we perform weighted random choice based on the information extracted by the activity graph in [70]. In the same way we choose the rest 6 slots where the user is active only in the Internet (can seed).

As described in [66], all OSN users watch videos mostly from 4 interest categories (94% of videos). Thus, we used a weighted random choice and we chose 4 categories out of 19 total interest categories, using as weights the share of each category in total distribution. Additionally, we assumed a specific number of ASes in the network layer and we distribute the OSN users among the ASes, using the Zipf distribution. According to [71], Facebook users do manage to reach 35% of their friends with each post and 61% of their friends over the course of a month. In our framework, we set the 61% of friends of each user as his monthly audience at 1-hop. As a result, we achieve to produce an average audience of 34% of friends per post at 1-hop. Based on [66] observations and in categorization of friends in in the introduction of Section 6.2, we specify the distribution of viewers of an uploader in 1 and 2 hops from the uploader over the three categories of viewers, choosing from users with at least one common interest with him: a) Followers: 36.3% of total viewers are characterized as 1-hop followers, while 2.2% of total viewers are characterized as 2-hops followers, b) Non-followers: 40.6% of total viewers are characterized as 1-hop non-followers, while 13.2% of total viewers are characterized as 2-hops non-followers, c) Other viewers: 2.2% of total viewers are characterized as 1-hop other viewers, while 6.5% of total viewers are characterized as 2-hops other viewers.

Moreover, we created a hypothetical pool of videos in order to simulate a video platform like YouTube. We assigned popularity to each video using the Power Law distribution and an interest category by weighted random choice using as weights the percentages of each one of 19 interest categories. Thus, we create a number of videos with a distribution over 19 categories similar to YouTube and long-tailed regarding the popularity. Each user is assumed to have access to the videos published from his 1-hop friends, while viewers at 2 hops arise by means of sharing videos; this is a realistic assumption taking into account the privacy settings employed in OSNs such as Facebook. Each user watches videos related to his interests.

As expected, these videos belong to his top interest, as well as videos with highest popularity are more likely to be watched. Thus, the actual percentage of the videos a user watches from each category depends on his top interests. Next, we specify the number of
videos watched per user. Each user is active in Facebook for 20 minutes on the average per day and considering that the average length of a video is 4 minutes [73], we assume that a user may watch from 1 to 5 videos in this 20-minute interval, where the number of videos watched is chosen according to the uniform distribution.

Based on [74], we observed that the total amount of links that are shared in Facebook every day is 15 times smaller than the total number of Facebook users. Also, most of these links were found to be mainly videos or pages containing videos. Furthermore, on the average, the self-uploaded UGC videos, i.e. videos uploaded and hosted in a Facebook server, account for about 14% of all videos uploaded in Facebook. The remaining videos are hosted in origin video servers with the 80% of them to be hosted by YouTube [66]. Considering the above Facebook statistics, it is rather realistic to assume that the number of videos uploaded daily in our system equal to 5% of the total number of user in our system. For each day, we decide which users will play the role of the uploader, i.e., to upload and share videos, by employing the Bernoulli distribution. That is, the users are chosen with uniformly randomly from the set of this day’s active users.

Additionally, each user can upload none, one or more videos, but only within the 20-minute slot that he is active in the OSN. According to [75], Facebook constitutes currently the second largest source for videos, and it generates about 11.8% of all referred video traffic in the Internet. Based on this observation, we thus assume that the number of videos that a user re-shares 11.8% of the total number of videos that he uploads. We characterize as ‘re-shares’, the videos that a user watched from a post of one of his friends and then posted it on his own OSN profile.

6.2.4 Evaluation results

In this section, we present and discuss the evaluation results for the proposed mechanism, and we perform a comparison of SEConD to the mechanism of SocialTube presented in [66]. In our experiments, we consider only the traffic generated by the video delivery, not considering the impact of background traffic. However, due to the fact that video streaming traffic dominates global Internet traffic [96], the background traffic is not expected to not affect impact of the SEConD mechanism on end-users’ QoE, or the contribution of the original video server, the SPS or the P2P overlaythe results of the evaluation on inter-AS links, but as future work we have to consider it in the intra-AS links in order to avoid congested links. Considering the class of traffic, is delay sensitive videos. Thus, it is assumed that all the mechanisms evaluated, operate in the appropriate way to maintain the QoE high, by avoiding stalling events elimination.

We observed that both SEConD and SocialTube achieve high prefetching accuracy, namely around 88%. This is due to the fact that both mechanisms follow a similar approach in the prediction of potential viewers where a prefix will be pushed. We also observed that, as in SocialTube, the prefetching accuracy under SEConD is higher for users that watched more videos and for those that received more prefixes.

The results obtained show that our mechanism indeed achieves significant reduction of inter-AS traffic. If we assume that prefetching in the current OSN video sharing system architecture, i.e., following a client-server architecture, generates daily a total of 100% inter-AS traffic, then SocialTube is found to generate 66% of inter-AS traffic, while SEConD only 12.6%. This high reduction is achieved due to the fact that in SEConD the users are pushing alert messages through the messaging overlays instead of pushing video prefixes, whose size is larger than that size of alerts. Also, in SEConD the prefix of
each video is downloaded once per AS, and thus the redundant inter-AS traffic due to the same prefix is eliminated.

Moreover, SEConD achieves high reduction of the total inter-AS traffic generated by the complete process of video dissemination, including alerts and prefetching. Under SEConD, the total inter-AS traffic generated accounts for only 13% of the total inter-AS traffic under the client-server paradigm; namely, a 87% reduction of total inter-AS traffic is achieved by our mechanism. For SocialTube, a reduction of inter-AS traffic of 18% compared to client-server is attained. Figure 6-11 illustrates the total inter-AS traffic generated by each mechanism within a specific full day. Clearly, SEConD achieves a high total reduction during the day, while it is very effective during the ‘busy’ hours although users’ activity is higher. SEConD performs better during the “busy” hours, since more users are active and P2P contribution increases.

We investigated the total contribution of the origin server where the video is hosted. SEConD achieves high reduction of the percentage of traffic handled by the origin video server. In the client-server architecture, the origin server contributes 100% of the video traffic. The relevant server’s contribution in SocialTube drops to 55.3%, of the video traffic in the client-server case, while under SEConD it drops to 12.1% thereof. Practically, when SEConD is in place, inter-AS traffic is generated mainly due to the contribution of the origin video server, which is maintained at a low level.

### Table 6-4: Proxy server and SPS contribution w.r.t. the AS size.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>AS1</th>
<th>AS2</th>
<th>AS3</th>
<th>AS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of users</td>
<td>1925</td>
<td>928</td>
<td>634</td>
<td>476</td>
</tr>
<tr>
<td>Proxy contribution</td>
<td>48%</td>
<td>61%</td>
<td>75%</td>
<td>79%</td>
</tr>
<tr>
<td>Proxy cache size</td>
<td>63GB</td>
<td>30GB</td>
<td>21GB</td>
<td>15.7GB</td>
</tr>
<tr>
<td>Proxy cache hit accuracy</td>
<td>90/94 %</td>
<td>78/86 %</td>
<td>68/80 %</td>
<td>60/74 %</td>
</tr>
</tbody>
</table>
Then, we investigate the impact of the AS size on the contribution of the SPS and the origin video server under SEConD. In Table 6-4, we observe that as the size of the AS increases the contribution of SPS decreases. We see that in the largest AS (i.e., AS1), the contribution of the SPS is 48%, and due to the very high (~90%) hit accuracy of the cache, the contribution of the origin server is just 5%. On the contrary, in the smaller AS (i.e., AS4), the contribution of the SPS is 79% and that of the origin server is up to 31%. This is explained by the fact that the resources of the local P2P overlay are lower for AS4, due to lower number of users, and thus SPS supports video dissemination more frequently.

Additionally, the contribution of the origin server decreases with the hit accuracy and the contribution of the SPS cache. Indeed, when a user requests a video from the local SPS, the SPS requests the video from the origin server, only if he doesn’t have it cached. We have also noticed that relatively less SPS caching capacity is needed in large ASes than in small ones, in order to achieve lower utilization of the origin server and also preserve the same (high) QoE level.

Finally, we set up and ran a number of experiments assuming a peering agreement between different pairs of ASes each time. The evaluation results reveal that under a peering agreement both SocialTube and SEConD can be more beneficial in terms of total inter-AS traffic created, additionally SEConD can be more effective in terms of reducing the origin server contribution.

Concluding, SEConD: a) enhances the QoE of OSN users both by achieving high overall prefetching accuracy and by maintaining an adequately high upload bandwidth within swarms, b) achieves high reduction of the inter-domain traffic, and thus, may result in reduced charges for transit inter-AS traffic, c) reduces the contribution of the video server of the OSN (CDN), as well as the relevant operational costs, i.e. mainly costs for bandwidth, and d) eliminates redundant prefix downloads, leading to reduction of traffic congestion within the AS. There are several potential promising future extensions of SEConD. In particular, we can employ direct caching of the entire video in the SPS of the origin AS to achieve high QoE in the first downloads. We also plan to implement and evaluate the monitoring component of SPS. Finally, alternative caching policies and different sizes of the cache of the SPS can be employed and evaluated.

6.3 The vINCENT extension for RB-HORST

The vINCENT (virtual incentive) extension for RB-HORST addresses the EFS use case of mobile offloading from cellular communication to WiFi. It aims at providing a technical means and incentive for accessing untrusted access points, leveraging social data to do so in an ISP friendly way.
The main part runs on a user controlled Nano Datacenter (uNaDa), which is established on the end users’ home router. The has to install a MobileAPP on his Smartphone which is part of the vINCENT extension and is responsible for user authentication and manages the access to WiFi.

The extension consists of two layers: a privacy layer and an incentive layer. The layers are orthogonal to each other. Both layers are described in the following.

- **Privacy layer:** The privacy layer is intended to handle the problem of overhearing at untrusted uNaDas. Considering the context of Figure 6-12, the mobile user offloading at the untrusted uNaDa 1 would expose his traffic to the owner of the uNaDa. As a countermeasure, the mobile user can set up a VPN tunnel to his very own uNaDa or to a uNaDa of a friend (uNaDas 2-4), which terminates the session, i.e., serves as an end point for all mobile traffic generated by the smartphone. The selection of the endpoint is done in a way minimizing cross traffic between ISPs (in Figure 6-12, this means selecting uNaDa 2 over the other trusted uNaDas, as it resides in the same ISP network as uNaDa 1). The tunnel setup is coordinated by the Facebook APP, which has a global view on all uNaDas online and serves as a central trusted entity.

- **Incentive layer:** In order to provide an incentive for untrusted uNaDas to offer service to untrusted mobile users, an incentive scheme is introduced, which is enforced by the Facebook APP. uNaDas and mobile users record service received and granted to the Facebook APP. The Facebook APP calculates a rate R based on this history. R determines the data rate that is granted to the mobile user based on the performance of his/her very own access point.

Since the terminating uNaDa runs a RB-HORST instance, it can intercept the mobile user’s connection to deliver content from the local cache. Thus, while vINCENT cares for good Quality of Experience (QoE) while accessing the mobile network by offloading mobile traffic from 3G to WiFi, RB-HORST can decrease access time for cached content, both while offloading to an untrusted uNaDa.
6.3.1 Intelligence
This section explains the internals of the vINCENT extension. First, the terminology and entities shall be explained, based on the terminology used by RB-HORST:

**HORST-VINCENT** The mechanism which is specified in this document. The following entities are part of the mechanism:

- A Facebook APP running on a web server, it is used to gather information from the Facebook OSN and runs the central algorithms defined in this document.

- A Mobile APP allowing to find other uNaDas for offloading and handling the setup of secure connections.

**Trusted Set** A set of users of a social network which are trusted by a user. The trusted set is managed in the context of the Facebook account of a user (as a friend list, i.e., a subset of a user's friends). Users from the trusted set are allowed to use the user's access point as a tunnel termination point. Furthermore, users from the trusted set are allowed to use the user's uNaDa for offloading without restrictions, if their smart phones are in physical reach of the user's uNaDa. *(Un-)trusted uNaDa*

A uNaDa that is (not) trusted by a mobile user, i.e., the owner of the uNaDa is (not) in the user's trusted set. If the mobile user is offloading at an untrusted uNaDa, the privacy layer and incentive layer kick in to provide privacy for the mobile user and an incentive for the owner of the untrusted uNaDa.

**History** An internal database of the Facebook APP recording past behaviour of the participating users, e.g., the provided service in terms of offloaded connections.

**Access Token(\*)** A Facebook User Access Token as defined in [76]. Whenever the term is used with an Asterisk, an implicit mapping of Access Token to User ID is assumed, i.e., Access Token* = User ID.

**Entities and Preconditions** As we have defined the terminology including the entities of the system, we walk through each entity and define the preconditions to be held by these entities in order to implement the mechanism.

A user associating to the open WiFi is not allowed to access the Internet with the following exceptions:

- The Facebook authentication server in order to login to the social network (similar to RB-HORST).

- The server hosting the Facebook APP in order to enable the setup of a tunnel to the terminating endpoint.

The vINCENT extension has to be able to cut of the connection after a minimal volume was used. The minimal volume is to be chosen as high as necessary to authenticate with Facebook and manage the tunnel setup without allowing the abuse of RB-HORST access points without authentication to, e.g., browse Facebook. Moreover, the vINCENT extension on the uNaDa needs to be able to start a VPN server and terminate a VPN connection coming from a certain IP. This tunnel setup is triggered by the Facebook APP,
which sends the tunnel parameters, the source IP, and a shared secret for the tunnel setup.

The **Mobile APP** uses HTTPS connections for communicating with the Facebook authentication server and the Facebook APP, thus ensuring, untrusted uNaDa(s) offloading the connection cannot sniff on the authentication procedure and the tunnel setup. Moreover, the Mobile APP needs the ability to let a mobile user define the Trusted Set. Finally, the Mobile APP offers functionality to setup a VPN tunnel to an IP as instructed by the Facebook App. Therefore, at untrusted uNaDa(s), the Mobile APP needs to authenticate with the Facebook APP and request the setup of a VPN tunnel. The Facebook APP then sends the tunnel parameters, the destination IP, and a shared secret for the tunnel.

The main algorithms (defined in Appendix 13.7) are run by the **Facebook APP**:  

**Terminating endpoint selection algorithm:** The Facebook APP’s selects a trusted terminating uNaDa with a minimal distance in AS hops (between the point of access and the trusted/terminating uNaDa). For that purpose, the Facebook APP needs to be able to maintain a view on uNaDa(s) being online and has to be able to contact terminating uNaDa(s) upon request. Therefore, a set of terminating uNaDa(s) based on social information (friendship) provided by the user is selected. The candidate list is ranked in an ALTO-like approach with respect to provider friendliness, i.e., using the hop count as a metric. The distance information is either collected from the ISP, RB-HORST or geographic distance, depending on which sources of information are available.

**Incentive algorithm:** For providing the incentive to participate in the system, the Facebook APP needs the ability to specify a granted bit rate for a given mobile user offloading at an untrusted uNaDa. The untrusted uNaDa shapes the tunnel according to the Facebook APP’s mapping to a bit rate. The bit rate is calculated based on past behavior of users, which is recorded in the **History** of the Facebook App. Therefore, the Facebook APP regularly calculates the regression parameters of a linear regression model using the **History**. The regressors in this model are parameters influencing the performance of a uNaDa, such as the size of the trusted set or the geolocation of the uNaDa. The regressand is the measured performance in terms of provided offloading service. Thus, the model can be used to calculate the expected performance for a given uNaDa. Positive or negative deviations of single uNaDa(s) from the model (residuals) are rewarded or punished. A detailed discussion of communication patterns and algorithms used for the respective algorithms can be found in Appendix 13.7.

### 6.3.2 Parameters & Metrics

We evaluate the vINCENT extension using a number of different parameters and metrics. Most interesting, with respect to mobile offloading as a technology, is the system wide energy consumption over all mobile devices participating in the network. This is not so much a question of saving energy on the side of stationary devices, but a question of preventing battery drainage by performing a selection of more energy efficient interfaces when possible.

Moreover, it is interesting to assess the performance of network metrics. In particular, the following metrics are interesting:

- Transferred data volume over cellular access versus WiFi access.
- ISP cross traffic induced by the mechanism.
• Message Overhead, i.e., the additionally needed message overhead to apply the mechanism.

Besides the technological parameters, the economic side of vINCENT as an incentive scheme has to be taken into account. The incentive performance can be measured by evaluating the scenario under two different assumptions:

• Total control, i.e., every node is willing to act in the sense of providing a globally good solution. In our case, this means users are assumed to be altruistic. This leads to all access points being accessible for everyone willing to offload, thus maximizing the overall saved energy and maximizing the offloaded traffic volume.

• Anarchy, i.e., all users act rationally and only open their access point when an incentive is present as it is intended to be provided by the vINCENT scheme.

The difference in performance between both cases (the “Price of Anarchy”) is a measure for the efficiency of the scheme and should be minimized. A more differentiated view on the metrics was provided in D2.3.

6.3.3 Evaluation framework

We use the OMNeT++ simulation framework with the INETMANET extension [107], which is an extension containing additional precise simulation models for wireless use cases. The OMNeT simulation framework provides discrete event simulation down to the MAC layer, which is preferable in the case of vINCENT, as MAC Layer effects can have severe impact on the outcome of the simulation. Moreover, the vINCENT mechanism is probably difficult to evaluate experimentally, as real-world experiments would require a large deployment of nodes and physical presence of client devices at the uNaDas. Thus, we aim at a very precise simulation model. In order to do so, a number of well-known wireless models in combination with real-world traces are used.

Wireless Channel Model: As a channel model for the simulation of wireless access, the Free Space Path Loss (FSPL) [97] model is used, which is dependent on distance and frequency of the wireless transmission.

MAC and IP Layer Model: For the MAC Layer, OMNET’s widely accepted AccessPoint model is used. The model includes a simulation of association and disassociation events. In order to handle the assignment of IP addresses, each access point is associated with a
DHCP server, thus very precisely emulating the usual setup at user’s premises (see Figure Figure 6-13: Basic OMNET++ setup.).

**Backend Model:** For the sake of reduced simulation complexity, we assume the backend to not influence the performance of our system. However, backend connectivity has to be projected to the simulation model. Thus we assume all access points to be connected to one backend server via a single router. Both, router and server are configured to have unlimited resources (cf. Figure 6-13).

**User Mobility Model:** The user mobility model is derived from a set of publicly available traces (cf. Section 4.6). The models are converted into the BonnMotion format which can be read by the simulation framework.

**Access Point Distribution Model:** The access point distribution model is derived from a set of publicly available traces (cf. Section 4.6). The models are converted into a custom CSV based format which can be read by the simulation framework.

User Mobility Models and Access Point Distribution Models are based on the Amherst trace set [98]. The set is based on measurements from bus lines from Amherst, Massachusetts. Two indicated values are important for this simulation: the access point location and the movements of the busses (both annotated as GPS coordinates). Both, user mobility model and access point model are based on GPS coordinates. The projection to the OMNET++ simulation plane is done by using a Mercator projection, i.e., a cylindrical projection. Moreover, the traces contain the MAC address of each access point seen and the time of sight, i.e., the time for which access point was seen.

![Figure 6-14: Visualization of analyzed traces.](image)

All duplicate connections were deleted and only those with the longest connection time for each access point are used. After the reduction of duplicates, about 4000 access points are identified. The measurement’s location is assumed to be the access point’s location. Moreover, the mobility of the busses, which also implies a mobility of the user’s of the busses, is extracted and converted to the BonnMotion format [99]. The preprocessed
traces are visualized in Figure 6-14. The evaluation results of vINCENT will be presented in D2.5.

6.4 Mobile Network Assistant (MONA)

The Mobile Network Assistant (MoNA) is a mechanism to be deployed on smartphones, optimizing the mobile network access in terms of energy-efficiency, while not degrading the QoE of the end-user in the decision process. MoNA addresses the mobile offloading scenario of the EFS use case. The main contribution of MoNa is in the location and usage dependent scheduling, batching and offloading of mobile data requests. This reduces the load on congested networks and shifts it to other locations, times, or network. For this, an extensive network quality map is generated and used, allowing the estimation of the mobile network performance in terms of throughput and delay at a given time and location. This approach considers only delay-tolerant traffic, which may be scheduled in space and time without affecting the QoE.

The mechanism considers the device state, user preferences and past user behaviour to adapt itself to the user’s preferences. Thus, the QoE can be kept high without interrupting the user or delaying important data transfers.

Figure 6-15: Schematic overview of the MoNA mechanism

6.4.1 Intelligence

The main intelligence of MoNA is in the Network Optimizer, deciding when to connect to which network, or how long to defer data requests. Therefore, a number of variables need to be determined. These are the availability and performance of the available networks, the user mobility, the device state, the type of request, and the network requirements of
the requests generated on the mobile device. Based on these, combined with the connectivity options, the QoE of each option is determined. The estimated QoE is then used to decide which action to execute.

The QoE is calculated according to the models described in [78]. This is done by setting the expected system performance with a QoE of 4, once for the network performance, and for the battery lifetime. According to the model, a better performance increases the QoE up to 5, while poor performance leads to lower QoE estimates, where a QoE of 1 renders the system unusable. These metrics are combined by setting user defined weight factors to prioritise performance or energy consumption, from which a combined QoE estimate is derived. This is then used to optimize the scheduling decisions.

Figure 6-15 shows the internal architecture of the MoNA approach. MoNA itself is marked in the dashed box, consisting of a number of modules, some of them depending on external components. These external components are marked in grey. We assume the availability of these and the required behaviour.

The basic operation is the following: An application on the smartphone sends a network request using its built in functions. This request is intercepted by the Connectivity Controller and forwarded to the Network Optimizer. The Network Optimizer analyses the incoming request, forecasts the connectivity given the user's position and likely path, which is then combined with the energy consumption of the interfaces for the given connections. Knowing the network performance and the energy consumption of possible routes, the QoE model combines these to estimate the mean opinion score (MOS) of each option. The Network Action Recommender selects the best option and forwards the decision to the Connectivity Controller, which handles the application request accordingly. The action to execute include whether to connect immediately, specify the interface to be used, and in the case of low QoE or background requests.

The core component, the Network Optimizer is the component gluing together the components providing the required information. These components may already exist, like QoE models or energy models, or be completely new, like the Request Analyser, the Network Model, and the Network Action Recommender. Still, the existing components might need to be extended to support the functions required by the Network Optimizer. In Figure 6-15 this is indicated by colouring these new components in green, while existing components are marked in grey. The Request Analyser, Network Model, Energy Model, and QoE model provide or enhance the collected information, which is used by the Network Action Recommender to decide about how to connect to the Internet.

The detailed specification of the mechanism can be found in Appendix 13.8.

### 6.4.2 Parameters & metrics

MoNA derives its decision based on a number of parameters and metrics. The parameters can be categorized into environmental parameters, device capabilities, and user-dependent variables.

The environmental parameters, the decisions are based on, are the available network technologies with their associated data rates and round-trip times (RTT), the location and time of the request. Here, the data rate and RTT are location dependent, to some degree also on the time of the day and the weekday.

The device dependent parameters are the available network technologies, the cell provider, the battery level, and the current device utilization. For example, if the smartphone supports 3G networks only, not all networks as provided by the Network
Model apply to the current configuration. The battery level defines, how much traffic should be pre-fetched, and also defines the maximum scheduling interval. Furthermore, transmissions of background services may be discarded to keep the QoE high, if the battery is low.

The user dependent variables include the number and selection of applications running on the smartphone, the content being requested, the mobility and dwell times, and the number and availability of WiFi access points allowing the offloading of connections.

Based on the metrics ‘energy saved’ and ‘average QoE’ of the end-user, the mechanism is evaluated. The goal is to save the highest amount of energy while not deteriorating the QoE, possibly even increasing it using pre-fetching of content and local playback with higher a quality and fewer stalls.

6.4.3 Evaluation framework
The performance of MoNA is under evaluation. The evaluations consider the user mobility, network traffic traces, available networks and their performance and power models of the mobile devices to derive the expected power savings. For this, OmNET++ based simulations will be run based on the measured network parameters like throughput and RTT. The required user mobility traces are generated using the BonnMotion framework [100], which allows generating a wide variety of user mobility traces ranging from random walk models to MSLAW, which simulates traces where nodes move between different destinations. From this, combined with the models in Section 4.4.2, the energy cost of each connection can be modelled. The QoE is derived according to the QoE models described in [101]. To calibrate the QoE of the different configurations, user-studies are required to calibrate the parameters used in the QoE model to optimize the performance of the approach.

The target outcome of the simulations is the energy savings possible using different scheduling timeouts under the condition that the QoE of the end-user is not decreased. The scheduling timeouts may be user-defined. Hence, the power saving potential depends on the user preferences as defined in the UI of the MoNA app.

6.4.4 Evaluation results
To evaluate the energy efficiency of using multiple wireless connections simultaneously, the Nexus S was set up with a MPTCP [102] enabled kernel and the energy consumption was measured. MPTCP also allows seamless handovers between different technologies, allowing TCP connections to remain active in the case where usually handovers occur. Furthermore, a single TCP connection can benefit from the aggregated bandwidth of WiFi and cell interface, such increasing the available bandwidth and reducing the overall transmission time.

Both features are important aspects allowing the optimization of the energy efficiency within MoNA by either prioritizing transmission time and QoE, or improving the battery lifetime by migrating connections to the most energy efficient interfaces eliminating the need for costly connection re-establishment. Furthermore, frequent handovers between technologies are expected when using MoNA. Here, MPTCP reduces the reconnection time by keeping TCP connections alive.

Our preliminary measurements show, that the energy consumption when splitting a single fixed data stream to multiple interfaces is less than the sum of the energy consumption of
two data streams with the data rate split on both interfaces (cf. Figure 6-16). Still, the mean power of the combined connections is higher than receiving the same data stream on a single interface only.

![Power Consumption of the Nexus S splitting a fixed rate data stream to multiple interfaces (3G, WiFi).](image)

Figure 6-16: Power Consumption of the Nexus S splitting a fixed rate data stream to multiple interfaces (3G, WiFi).

Next steps are the simulation of the energy saving potential for scheduled connections, and the resulting QoE for the end-user.

### 6.5 MUlti-Criteria Application endPoint Selection (MUCAPS)

MUCAPS stands for MUlti-Criteria Application endPoint Selection and is a mechanism that optimizes overlay applications that have a choice among their connection Endpoints. It improves the initial overlay Endpoints selection by involving awareness on the underlying network topology. Relevant applications are typically download of videos from a CDN or a location retrieved from a CDN, via a P2P or a uNaDas network. MUCAPS addresses the EFS scenario, in particular use cases locality in UNaDa and service and content placement for users.

SmartenIT investigates incentive-based mechanisms using a cross-layer approach for efficient management of traffic generated by overlay applications. MUCAPS provides a mutual incentive for the application layer and the network layer to cooperate, by involving on one hand selection criteria on transport network costs such as routing costs that meet ISP interests and on the other hand, criteria impacting QoE that meet end-users interests such as end to end path bandwidth. The QoE criteria used in MUCAPS abstract end to end performances on application paths. This way, applications take the ISP costs and constraints into consideration provided applications get reliable performance indicators from the ISPs.

The IETF ALTO WG has specified a layer cooperative transport protocol conveying to applications an ISP-centric abstracted view of the transport network topology together with routing costs on end to end paths. ALTO is an information framework with associated transport means and needs the automated intelligence (1) to feed the ALTO Server DB (2) to decide what information to request and what to do with the received information. This automation is performed by MUCAPS. MUCAPS builds the decision making around the information requested and received by an ALTO Client: (i) prior decision on which ALTO metric to request to select Application Endpoints (AEPs) and (ii) AEP performance evaluation and selection upon received ALTO values and application user access.
MUCAPS also extends the base ALTO protocol by introducing new metrics and ALTO transactions with multiple costs. MUCAPS is an instantiation of a general scheme illustrated in Figure 6-17.

The main feature of MUCAPS is a functional block called MACAO, an acronym standing for Multi-Acces and Cost AltO. MUCAPs may be generalized to support automatic selection of both ALTO metrics and ALTO services and allowing service requests from the user side. The MACAO block is a group of functions that is interfaced with the function tracking endpoints for an overlay application and referred here as AEPG for Application EndPoint Gathering function. A typical such function is a DNS Server, a DNS Client, a Peer tracker or a DHT. Using MACAO is relevant whenever the AEPG function outputs several acceptable candidate AEPs.

As illustrated in Figure 6-17, the MACAO functional block has an external interface called MARS for MACAO Request handling Server allowing it to communicate with a MARS Client (called MARC) hooked to the AEPG. MACAO services requests for AEP re-ordering that are sent by a MARC. MACAO enhanced AEP selection goes through the following steps:

- **Step 1**: the user endpoint (UEP) runs an application and requests for application endpoint (AEPs) IDs.
- **Step 2** (depending where the AEPG function is located): the application AEPG does an initial AEP selection (e.g. EP1, EP2, EP3).
- **Step 3**: before sending the selection to the application, it hands it to the MARS Client (MARC) that contacts the MARS via a simple IP socket carrying a request for re-ordering.
- **Step 4**: the MACAO functions performs the re-ordering and sends the revised AEP selection back to the MARC that now hands it to the application.
  - **Step 4.1**: an intermediate step is the request to the ALTO Server for information on transport network related costs.
- Step 5: the MARS sends the revised selection (EP2, EP3, EP1) to the MARC located in either the network or the UEP.
- Step 6: the application connects to the one or more selected AEP.

In-network transparent deployment

MUCAPS has been deployed as an in-network overlay traffic management mechanism as illustrated in Error! Reference source not found.. The MUCAPS service, in this case is equested by a MARC hooked to an AEPG located in the ISP network. In the evaluated deployment, the APEG is the DNS resolver located in the ISP network and connected to the DNS client located at the UEP side.

Figure 6-18: Deployment of MUCAPS in the ISP network.

6.5.1 Intelligence

The MACAO functions by acting at the the ISP network level, preparing Multi-Cost ALTO requests and processing their responses. Given an initial selection of (S,D) pairs or application paths, MUCAPS automatically selects the cost metrics that the ALTO Client will request to the ALTO Server, performs the (S,D) endpoint pairs or paths evaluation and ranking w.r.t. the received ALTO cost values and to this end, optionally fine-tunes the metric weights.

MACAO includes four blocks:

- AMWS: Application Metrics and Weight Setting module: identifies the metrics to use to evaluate the candidate solutions and their weights. According to ISP rules, this module associates performance metrics to application classes and metric weights to the UEP access capabilities.
- ALTO Client (AOC): requests values to the ALTO Server on the identified selection metrics.
- Multi Criteria AEP evaluation module: given cost values provided from AOS and their weights.
• ALTO Agent: federates all the previous components as it is able to communicate with each of them and pass information among them.

MACAO communicates with:
• The ALTO Server,
• The AEPG function: via the MACAO interface called MARS and the MARC hooked to the AEPG,
• An ISP configuration function that sets rules to map AEP selection metrics to the type of applications.

The base ALTO protocol provides applications with an ISP-centric network view and routing policy attempts thus to minimize inter-domain routing paths. ALTO is a Client-Server protocol that specifies the transport means to convey IPS centric information to applications and does neither specify what information a Client will request nor what the application must do with the received ALTO information.

Figure 6-19: Steps involved in MUCAPS and details on the AMWS (Application Metrics and Weights Selection) process.

A detailed sequencing of MUCAPS operations on an example scenario, network-based MUCAPS implementation and execution of the detailed MUCAPS specification are provided in Appendix 13.9 of this document.

6.5.2 Parameters & metrics

MUCAPS is hooked to the AEPG function that identifies the locations from which application content is to be received.

MUCAPS input parameters
The IP address of the Endpoint receiving the application content and referred to as User Endpoint (UEP).

A list of IP addresses of the application Endpoint that hosts the Application Content (for instance the video content). This list is presumed to be set by the application network. For instance an initial selection set by the authoritative DNS Server of the CDN. All the IP addresses of this list are assumed to represent feasible solutions for the application.

Internal parameters

- **Application ID**: an identifier of the type of application ran by the UEP
- **List of selection metrics**: given mainly the Application ID, the list of metrics w.r.t. which the candidate AEPs will be sorted. These metrics are mapped to the identifiers of the ALTO metrics in order to prepare the parameters of the request sent by the ALTO Client.
- **ALTO information**:
  - A network map representing the ISP-centric view of the Internet. It is composed of ISP defined network regions that each have a human readable Provider defined ID (PID) and a list of the IP addresses of the comprised locations.
  - A cost map specifying the ISP defined costs among the PIDs. The costs are defined w.r.t. metrics supported by the acting ALTO Server and comprise at least ‘routingcost’ a unit-less transport cost that may report any ISP policy or evaluation. Optional metrics supported in the MUCAPS ALTO Server are ‘hopcount’ and ‘path bandwidth score’.
- **List of selection metric weights**: associated to the metric used to evaluate the AEPs and depending on the access network type of the UEP.
- **Applicable metrics and weights set by the AMWS module**.

Note that the metrics currently must all have positive on null numerical values. Moreover, the base ALTO protocol has been accepted as a standard on March 2014, whereas the prototype was finalized in January 2014 and implements former versions of ALTO, validated by IETF interoperations. The protocol changes done since the MUCAPS ALTO version have no impact on the system performance.

MUCAPS output parameters

The input list of candidate AEPs re-ordered by MUCAPS.

Performance and their context

The impact of MUCAPS to the optimization of traffic generated by overlay applications is evaluated with a set of objective metrics such as ISP routing cost or end to end path bandwidth score. The latter reflects the user satisfaction level w.r.t. available path bandwidth. The discussion in Section 6.5.4 shows the need to couple user satisfaction metrics with UEP context parameters.

- At the network operator and end-user side: an analytical performance score can be computed on the selected AEPs w.r.t. metrics such as ISP routing costs and end to end path bandwidth performance, as described in Section 6.5.4. Values of these metrics are estimated by the ISP, stored in the ALTO Server. The MACAO Multi
Criteria AEP evaluation module computes the AEP performances with a vector-based algorithm.

- At the end-user side:
  - A device-aware visual quality score can be computed in terms of a ratio between the nominal bit rate and the UEP device capabilities.
  - The ultimate computed performance assessment will be done by measuring QoE metrics at the user side. Composite performance setting functions joining QoE metrics and UEP context are currently being defined and will be proposed in WP4.

The analytical performance score on AEPs is computed by MUCAPS to rank and select them. A network operator can directly infer its overall resulting routing costs from the ones of the AEPs selected by MUCAPS. Quantities such as bandwidth savings over a network can only be inferred by the ISP after complex aggregation of network state information. As a first step to this is, we compare the ISP estimated path bandwidth performance provided in the ALTO Server, between paths selected with and without using MUCAPS.

### 6.5.3 Evaluation framework

MUCAPS has been evaluated so far as follows: MUCAPS has been implemented and demonstrated as a prototype. Functional platform tests have been conducted to verify whether MUCAPS behaves as expected in the specifications. Besides, it has been evaluated w.r.t. the performance metrics mentioned above. MUCAPS attempt to optimize performances that go beyond the sole user perceived QoE, in particular OPEX in terms of ISP routing costs.

The prototype has been tested with a set-up illustrated in Figure 6-20 and composed of the following entities:

- A EU device with a DNS client and a VLC client,
- An AEPG which is a DNS Server with a MARC,
- An ALTO Server,
- A MARS hooked to the MACAO block and with an interface to the MARC,
- 3 video servers corresponding to AEP1, AEP2, AEP3.
The functional behavior of MUCAPS is assessed by computing the numerical performance of the selected EPs in different scenarios, mainly with or without MUCAPS. When MUCAPS is activated, cases are further refined to evaluate performances when one or more metrics are selected.

**Analytical performance computation method of the selected candidate EPs**

The method is described while providing two typical metrics considered as jointly meeting application and ISP needs. The application is video downloading. The AWMS for this has selected the following metrics:

- **Routing Cost**: a positive unitless value $rc(AEP)$ to minimize that reflects the ISP preferences w.r.t. its costs and policy.
- **Bandwidth Score (BW Score)**: a positive unitless value $bws(AEP)$ to maximize that reflects the ISP guidance on the bandwidth availability of the end to end path.

The values $rc(AEP)$ and $bws(AEP)$ are based on the values obtained from the ALTO server and processed so as to be numerically compatible. The candidate AEPs are evaluated and ranked according a positive scalar value to maximize that represents their *relative* utility. The term relative means that the AEP evaluation is endogen, based on their mutual comparison rather than w.r.t. some absolute exogen optimum which is not necessarily known and may have random value distortions that lessens the solution numerical robustness.

So the utility value reflects the proximity of the AEP to an *ideal* vector $(i_{rc}, i_{bws})$ composed of the best values for all metrics observed among the AEPs to compare. Note that the ideal does not necessarily exist.

Each AEP is represented by its AEP-relative performance vector $Pf(AEP)$, where each component maps to a selection metric and represents the proximity ratio $po$, for this metric, of the AEP to the best observed value on this metric.

The analytical AEP performance vector $Pf(AEP)$ takes the form:

$$Pf(AEP) = [po(rc(AEP), i_{rc}), po(bws(AEP), i_{bws})]$$
all components represent a performance w.r.t. the best observed value among the candidate AEPs,

as all components are normalized, they take values in [0,1].

A scalar utility function can thus be computed on the performance vector of each solution. The utility function to maximise may express several selection strategies. For instance:

- a MAX-SUM L1 utility equal to the sum of the component values reflects a compromise between user and ISP interests,
- a MAX-MIN LMIN utility equal to the maximal value on the component reflects a user need to ensure a minimal level of performance. A numerical example is shown in Table 6-5.

### 6.5.4 Evaluation results

Former studies have assessed the impact in general of using network layer awareness and lead to the design of the ALTO protocol, see [108]. Simulations in previous projects on P2P sessions in fixed networks have shown a decrease in download time when “network locality” and bandwidth are used, see [109].

The performance evaluation focuses is on the application flows and their costs evaluated at the ISP and UEP side. It assesses the impact of the re-ordering by MUCAPS of the candidate AEPs. The performance evaluation focuses on application performance, that is, analytical AEP evaluation and visual performance evaluation. The current evaluation criteria are twofold:

- A vector based method as described in the previous section jointly computes the AEP performance w.r.t. routing cost and bandwidth availability,
- A visual evaluation assesses the evolution of the visual quality for the delivered video, w.r.t. the AEP choice.

The functional evaluation shows a change of application endpoint (here a video server), given the utilized decision metric and the network conditions and displays it through a re-ordered list of IP addresses sent to the DNS Client.

#### Analytical evaluation

The three video servers have IP addresses and are respectively named AEP1, AEP2, AEP3 (AEP53 on the figures). Their “absolute” performances as provided via e.g. the ALTO protocol and relative performance w.r.t. an optimal value are displayed in Table 6-5. In order to be reliably compared, the performance vectors must be mappable to a scalar value. To this end, they have components of same nature and equivalent value ranges. The scalar value is produced by a utility function on the AEP performance vector, taking the maximal value for the best AEP.

**Table 6-5:** For each example AEP: utility vector of normalized relative performances and value for 2 utility functions and optimal AEP (= AEP1).

<table>
<thead>
<tr>
<th></th>
<th>AEP1 10.133.10.1</th>
<th>AEP2 10.133.10.2</th>
<th>AEP3 10.133.10.53</th>
<th>Ideal vector</th>
<th>Metric weight</th>
<th>Best EP = MAX Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTO - Routing</td>
<td>10</td>
<td>7</td>
<td>60</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
MUCAPS behaviour evaluation and methodology

The main evaluation scenarios are illustrated in Table 6-6, that summarizes how performance is assessed in different cases by observing the selected EP and its performance score. For simplicity, it is assumed that the application needs to select only one EP. Results are given on the set used in Table 6-5. It is also assumed that the EP performance score is given by the L1-Utility.

However, for a reliable evaluation, it is necessary to introduce contextual elements such as at least the access type and capabilities of the UEP and the network conditions, or simply user expectations on QoE. That is a mobile phone does not necessarily have the same QoE needs as a tablet or a personal computer. Therefore the following top-level scenarios have been defined to assess the MUCAPS performances while taking into account UEP context parameters.

- Scenario 1: without using MUCAPS. The first choice is EP3 (RC=60, BW=12). AEP3 is selected by the authoritative DNS of the CDN for its geographical proximity, w.r.t. the interests of the CDN. However, the RC is very high because EP3 belongs to a different non peering AS.
- Scenario 2: with MUCAPS and equal metric weights. The choice is AEP1 (RC=10, BW=20). The RC cost is lower than for EP3 but not the lowest. EP1 is preferred due to its good bandwidth score, a necessary criterion for video downloading.
- Scenario 3: the UE has a wireless connection with limited capabilities. The weight setting has changed and gives more importance to the routing cost. The choice is now AEP2 (RC=7, BW=10). It has the lowest RC value and a lower bandwidth score, as higher path bandwidth capabilities are considered unnecessary for the UE access type.

Table 6-6: MUCAPS behaviour and performances in 3 scenarios.
The prototype and demonstration set-up have been in fact accordingly evaluated and Table 6-6 shows corresponding results on the example AEPs.

A next step to this work is to define a formal set of context parameters for a robust functional and performance evaluation. This work is being conducted in the framework of a licenced product definition. Further evaluation on AEP selection will be conducted with varying access conditions, metric selection and metric weight settings.

6.6 RB-HORST++

The proposed traffic management solutions for the end-user focused scenario do not tackle completely disjoint use cases. Instead, for each use case they present different approaches and target different goals. Nevertheless, the different traffic management solutions also show several synergies, which can be exploited. Therefore, selected modules of each traffic management solution can be combined to a synergetic solution, which covers all use cases, targets more goals, and thus, shows an improved performance compared to the standalone mechanisms.

RB-HORST, which already tackles all three use cases, serves as a basis for the synergetic solution. Selected modules from vINCENT, SEConD and MONA mechanisms are integrated, either replacing some of the existing functionality, or adding new one to RB-HORST to improve its performance in selected UCs, i.e. content placement and wifi offloading, thus resulting in the RB-HORST++ or RBH++ mechanism which is described below.

6.6.1 Intelligence

Overlay

RBH++ enabled uNaDas organize themselves in a DHT overlay network (see Section 6.1). Moreover, one or several instances of a Social Proxy Server (SPS) (see Section 6.2) participate in the overlay, in order to localize the traffic generated by the activity of OSN users in a specific region, e.g. an AS.

SPS participation

The SPS does not always participate in video delivery, unless the upload bandwidth (proxy for QoS/QoE) received by uNaDas is not adequate. The received bitrate for each video is monitored in each uNaDa, and SPS is contacted only when the received bitrate falls below a certain threshold, i.e. the video bitrate. Therefore, each uNaDa is already aware of the SPS of his home AS, and thus, can request videos and video prefixes from that specific SPS at any time.

Content prediction & placement

RBH++ employs the Social Content Demand Prediction (SCDP) of RB-HORST (see Section 6.1), with the difference that the video delivery is chunk-based in order to achieve an efficient management of storage and bandwidth.

Using the SCDP algorithm, we determine in which uNaDa a specific video is highly likely to be requested in the future. Based on this prediction, we only push the prefix, e.g. first n* chunks of this video to this uNaDa.

In the case when a user requests to watch a video, we don’t download the complete video but chunks of it in a sequence using buffer management techniques. Then, since a video
starts being watched by a user, the remaining chunks, i.e. the ones that were not pushed to the uNaDa of the user as part of the prefetching procedure, are gathered following one of the following policies (algorithms):

- **sequential**: Chunks are fetched in the sequence that they will be watched. Due to the fact that the video delivery is SPS-assisted the probability for starvation is rather low.

- **rarest-first**: The rarest chunk in the overlay is requested first. This policy is inspired by the BitTorrent protocol [72], and requires that information on which chunks are acquired by each uNaDa at a specific time is available to all participating uNaDas as in [72].

- **mixed**: practically, a combination of the two previous policies, i.e. the selection of each odd chunk follows the sequential policy, while the selection of each even chunk follows the rarest-first.

### Cache management

The cache of both uNaDas and SPS is managed according to the outcome of the SCDP algorithm. According to the ranking, a priority is assigned. Thus, in a full cache, if a specific video item is not cached, while a second one with lower priority does, then the lower priority item is removed (swapped) and the higher priority item is fetched, whereas in an empty cache, the process starts prefetching the top ranked videos until the cache is full. If the cache is full videos above a certain age (1 week) and without cache hits will be removed.

### Mobile data offloading

RBH++ enabled uNaDa offers a primary private WiFi, which can only be accessed by the owner with his private credentials, and a secondary shared one, which is open, i.e., unprotected and authentication free WiFi and uses an SSID of the form "RBH++_<hashofwifiMAC>" for simple detection during WiFi scans. However, a user associating to the shared WiFi is not allowed to access the Internet with the following exceptions:

- The Facebook authentication server in order to login to the social network.
- The server hosting the Facebook APP in order to enable the setup of a tunnel to the terminating endpoint.

The management of trusted nodes and tunnelling is performed as described in Section 6.3.

### Energy efficiency

To achieve energy efficiency for the end-user mobile device, RBH++ employs scheduling of connections in space and time, based on a-priory network knowledge. In particular, knowing the RTT and throughput of available networks, i.e. 3G/LTE or WiFi, allows selecting the most energy efficient network connection (interface), considering the QoE of the end-user. Then, all applications running on the mobile device are served over that specific network interface. The traffic is managed for foreground and background traffic, although the optimization potential of background traffic is much larger.

In the basic version, RBH++ intercepts and manages network requests of all applications on the mobile device as describe in Section 6.4. An extended version also integrates
decisions made by the Mobile Traffic Manager (MTM) (see Network Optimizer in Section 6.4.1) into the SCDP algorithm.

In particular, the MTM comprises the following components: Request Analyzer, Network Model, Mobility Prediction, Energy Model, QoE Model, and Network Action Recommender. The latter, i.e. the Network Action Recommender, returns the NetworkOption to execute which is also forwarded to the RB-Tracker.

Energy-aware social content demand prediction

Periodically the uNaDa access the news feed of the owner and extracts all video URLs, which are ranked based on history, popularity, age, distance, social relationships and energy awareness. Practically, if the end-user mobile device is currently connected at any uNaDa, the most relevant URLs are downloaded and stored in the local cache of that uNaDa according to Section 6.1. On the other hand, if the end-user device is connected through a 3G/LTE interface, a message containing the most relevant URLs for this user is sent to the SPS, which is located near the PoI where the Fixed and Mobile Network Operators inter-connect.

6.6.2 Parameters and metrics

The set of parameters and metrics identified in Section RBH, SEConD, vINCENT and MONA are important for the development and evaluation of the RBH++. Moreover, in the context of the chunk-based video delivery, each video should characterized by a video id and a fixed chunk size, while its chunk should characterized by the video id and the chunk sequence number (see Figure 6-21).

![Figure 6-21: Chunk-based representation of a video.](image)

The size of the chunk should be large enough, in order to eliminate the stalling event when the user requests to watch a video and uses the prefetched chunk, while on the other hand it should be rather small so as to generate only minimum redundant traffic (due to unnecessary prefetching). Thus, we choose to define a rather small chunk size, e.g. 250 KB or 512 KB, while one or multiple chunks, say \( n^* \) chunks, could be prefetched per video in order to achieve lowest stalling time. The number of chunks to be prefetched can be determined as follows:

\[
n^* = \min\{N \cdot c \cdot 5\%, 50\},
\]

which practically means that \( n^* \) is equal to the total number of chunks of the video \( N \), multiplied by a factor \( c \rightarrow (0,1) \) that implies the video coding quality; thus, higher resolution, e.g. HDTV, would correspond to a higher value of \( c \) such as 0.9, while lower resolution such as CAM would imply a lower one, e.g. 0.2. Moreover, \( n^* \) is determined by the video duration, in the sense that is should not exceed the 5% of the total duration, while overall it should lower than 50 (i.e. corresponding to 12.5 MB maximum for high quality long duration video items).
7 Coverage of the SmartenIT objectives

In Deliverable D2.2 [5], we have categorized the initial TM mechanisms developed by SmartenIT, spotted their overlaps and their complementarities, and identified the possible synergies or even combinations of mechanisms that will provide more concrete and broader solutions for the management of overlay and cloud traffic. Indeed, based on the categorization performed and the complementarities identified, three synergetic solutions were identified, which are described in this deliverable.

In particular, in Chapter 5 and Chapter 6, along with the complete specification of the TM mechanisms developed within SmartenIT we also presented in detail three synergetic solutions: a) DTM++ incorporating DTM and ICC to address bulk data transfers, b) RB-HORST++ incorporating RB-HORST and SEConD to address content placement, and c) RB-HORST, vlNCENT and MONA to address WiFi offloading.

However, each of these synergetic solutions addresses only one of the two SmartenIT scenarios, either the OFS, or the EFS. In this chapter, we address the fulfillment of the SmartenIT objectives and targets and initiate the study on further potential synergies among the specified TM mechanisms to address a broader part of the SmartenIT playfield covering aspects of both the OFS and the EFS by the same synergetic solution; this study will be concluded in Deliverable D2.5.

7.1 Coverage by individual mechanisms and synergies

In this section, we summarize the coverage of the SmartenIT playfield by the specified TM mechanisms and synergies, as they have been described in Chapter 5 and Chapter 6. To do so, we first need to remind how the SmartenIT playfield is formed based on SmartenIT key objectives and targets defined in [1].

Figure 7-1 presents the mapping of the SmartenIT architecture on the cloud, network and end-user domain [7]. This topological diagram can be used as an overview of the TM mechanism and synergies w.r.t the SmartenIT architecture, rather than as a detailed mapping to components. As apparent from Figure 7-1, three domains of entities are present:

- **Data Center/Cloud Layer**: This layer comprises data centers and their virtual interconnections over Best Effort Internet.

- **(Core and Access) Network Layer**: The core and access layer network contains components in the ISP network and the private networks of data center operators.

- **End User Layer**: The end-user layer covers access network infrastructure as well as the end-user’s devices. In particular, this layer contains the user’s terminal devices and the home router.
Furthermore, in order to assess the coverage of the SmartenIT field by the developed TM mechanisms, we employ the categorization methodology developed in Deliverable D2.1 [4]. In [4], we have extensively overviewed a multitude of related state-of-the-art approaches that addressed partly the objectives of the SmartenIT project [1]. In order to assess the relevance, applicability and potential benefit for SmartenIT for each of these approaches found in literature, we performed a categorization based on the following criteria (briefly described here – a more extensive description is available in [4]):

- **Cloud layer** (mentioned as application layer in [4]): We distinguish TM mechanisms whose modules or functionalities are executed in centralized cloud/datacenter infrastructures controlled by Cloud Service Providers,

- **Network layer** (mentioned as lower layer in [4]): TM mechanisms whose modules or functionalities involve the core or access part of ISPs,

- **End-user layer** (mentioned as application layer in [4]): We distinguish TM mechanisms whose modules or functionalities involve the devices at the end-users premises, e.g. a mobile application, or the access router (CPE),

- **Inter-domain**: TM mechanisms that are applicable within a single network’s administrative domain or it can be generally applied in multiple domains,

- **Collaborative**: TM mechanisms that address the incentives of several entities, either in the same (e.g. cloud federation) or in different layers, so as to reach a goal by means of collaboration,

- **Energy efficiency**: TM mechanisms that consider and/or pursue energy efficiency,
• **Social awareness**: TM mechanisms which employ information derived by OSNs so as to perform some decision making,

• **Resource enhancing**: TM mechanisms that insert extra capacity, in terms of storage, bandwidth or computation, in the considered setup,

• **Long-term time scale**: TM mechanisms whose effect is demonstrable in the course of weeks or months,

• **Incentive-based**: TM mechanisms that pursue incentive-compatibility, rather than being applied compulsorily.

Note that we omit here the “End-user related” criterion, as it is addressed jointly with the “End-user layer” one, while we include two more criteria referring to the “Cloud layer” as defined in [7] and “QoE-awareness” [1]. Thus, mapping the TM mechanisms presented in this deliverable, including the two synergetic solutions, we derive the following table (cf. Table 7-1).

Table 7-1: Mapping of TM mechanism and single-scenario synergies to the layers and objectives of the SmartenIT field.

<table>
<thead>
<tr>
<th></th>
<th>Cloud layer</th>
<th>Network layer</th>
<th>End-user layer</th>
<th>Inter-domain</th>
<th>Collaborative</th>
<th>Energy efficiency</th>
<th>Social awareness</th>
<th>QoE awareness</th>
<th>Resource enhancing</th>
<th>Long-term time scale</th>
<th>Incentive-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTM</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICC</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRA</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>RB-HORST</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>SEConD</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>vINCENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>MONA</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUCAPS</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTM++</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBH++</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As it can be observed, all TM mechanisms addressing the OFS employ modules operating in the cloud layer, while DTM and ICC also operate in the network layer and both address inter-domain traffic. Additionally, MRA is a collaborative mechanism, which also pursues incentive-compatibility, while DTM and ICC are mechanisms, whose impact is observable in larger time-scales, i.e. monthly period to derive 95\textsuperscript{th} percentile values.

On the other hand, all mechanisms addressing the EFS expectedly operate in the end-user layer. Out of the EFS mechanism, RB-HORST, SEConD, and vINCENT are
collaborative and incentive-based, while the two first employ social awareness and consider inter-domain traffic. QoE-aware are RB-HORST, SEConD and MUCAPS, while MONA is the one addressing energy efficiency at the end-user layer.

Regarding the two single-scenario synergetic solutions, we observe that they “inherit” the characteristics of the mechanisms (or their modules) that they employ. Thus, DTM++ operates in both the cloud and network layers, it addresses inter-domain traffic and is applicable in a long time-scale, inheriting the attributes of DTM and ICC. On the other hand, RBH++ practically, addresses all criteria/objectives except the fact that it does not operate in the cloud layer and in large time scales. Thus, we conclude that indeed the integration of the mechanisms into synergetic solutions leads to a broader coverage of the SmartenIT objectives and targets.
8 Summary and Conclusions

Deliverable D2.4 aimed to provide a deeper understanding of the technical, economic and business issues related to overlay and cloud traffic management and to the stakeholders involved, by means of the development and study of new mathematical and simulation models, as well as of the extensive study of related business models. Moreover, D2.4 aimed to provide the mature specifications of the incentive-based cloud traffic management mechanisms that have been proposed in D2.2 [5].

In order to address the first target, a multitude of applicable business models involving Cloud Service Providers, Data Center Operators and Network Operators (or ISPs) have been studied and some major recommendations were derived. Taking the study on business models into account, SmartenIT studied a multitude of technical and economic issues and developed new mathematical models to describe the technical and business dependencies, as well as to facilitate the evaluation of certain aspects of the traffic management mechanisms. For instance, investigation of incentives for cloud federation and resource sharing, either in datacenters, or in network-edge nodes, has been performed. Concerning the cloud federation, new economic models including pricing and game-theoretic aspects are under development, taking into account the economic and business dependencies, which have been investigated in [2] and [5].

Regarding the second target, we maintained the mapping of the TM mechanisms to the two main scenarios of SmartenIT, which were introduced in [5], in order to present in a structured way the mechanism and their potential synergies, namely combinations of mechanisms that allow a broader coverage of the SmartenIT playfield. For each mechanism or synergetic solution, the intelligence including major entities is described, while a more detailed specification including sequence diagrams and algorithms in pseudo-code is given in the Appendix. Furthermore, the associated parameters and metrics, and the evaluation framework for each mechanism or synergetic solution are defined, while the evaluation results obtained by means of theoretical evaluations, and simulations are presented. Finally, we discussed the coverage of the SmartenIT playfield by the developed traffic management mechanisms and the potential integration of multiple mechanisms in a cross-scenario approach.

In the following sections, we summarize the key outcomes of D2.4 and future steps towards the forthcoming deliverable D2.5 and the conclusion of Work Package 2.

8.1 Key Outcomes and Lessons Learnt

In this section, we outline the key outcome of the study of business models reported in Chapter 3, and the theoretical investigations documented in Chapter 4. Moreover, we summarize the lessons learnt from Chapter 5 and Chapter 6, i.e. the development and evaluation of a set of TM mechanisms and synergetic solutions addressing the OFS and EFS, respectively. Finally, we present conclusions from the investigation of the cross-scenario synergies performed in Chapter 7.

In Chapter 3, we performed an extensive study of appropriate business models and derived some key recommendations for SmartenIT. Specifically, the SmartenIT TM mechanisms addressing the OFS are applicable under realistic business models and can serve as differentiated - in terms of statistical guarantees and expected quality – multi-domain network transport services, optimizing network resource usage, supporting multi-domain managed services and federations of datacenters, mitigating operators' network
and energy costs. Combined with market institutions such as open markets where IT and CT services are traded, SmartenIT TM mechanisms can promote competition and allow for the efficient roll out of new cloud services on top of the European ISPs’ networks. SmartenIT EFS mechanisms enhance the end user devices so as to promote energy efficiency and quality of experience especially within the emerging “terminal+service” and “HaaS+Internet” business model.

In Chapter 4, we presented a set of theoretical models employed to characterize content demand, topological distribution of home routers and WiFi hotspots, caching effectiveness, incentive-based resource sharing and workload migration in the case of a cloud federation, VM migration between DCOs based on constraints of multiple physical resources, power consumption in end-user devices and mobile transmissions, and incentive-based sharing of end-user resources. Major outcomes of this investigation are as follows:

Crucial for the performance evaluation of caching systems as deployed in RB-HORST is the study of proper models of content demand. We develop a content demand model to generate requests with temporal dynamics that exhibits a Zipf popularity law. For efficient prefetching of content based on social information we provide a model that predicts individual content requests based on social network traces.

In order to estimate the potential of home router sharing approaches like RB-HORST, the number of home routers available in an ISP’s autonomous system (AS) is required. We develop a model for the global distribution of home routers on ASes by evaluation the Internet Census dataset. Furthermore, the database OpenWiFiSpots was used to gather the addresses of public WiFi hotspots of the sample cities, to derive a model for the geographic distribution of WiFi hotspots, which is used to evaluate the performance of WiFi-offloading approaches as proposed in RB-HORST or MoNA. The evaluation of the Internet Census dataset shows that the distribution of home routers on ASes exhibits a power-law. Thus, the capacity provided by shared home routers highly depends on the size of the AS. The results of the WiFi offloading simulation show that more than 60% of the connections can be offloaded by RB-HORST / MONA, if only 10% percent of WiFi access points in a city area are shared.

Concerning the potential of federation formation among Cloud Service Providers, it has been shown that there is high potential for performance and providers’ revenues global optimization, when the providers cooperate. To achieve this, agreements between the providers participate in the federation should be established. To achieve this, agreements between the providers participate in the federation should be established. Thus, certain payment and compensation functions should be defined, for the providers incurring a revenue increment or a revenue loss respectively. In the future, we aim to study in which cases higher revenue can occur under cooperation compared to the aggregate revenues of the stand-alone providers.

Correctly capturing the interdependencies of multiple physical resources (CPU, RAM, bandwidth, etc.) for VM performance is complex. However, it turned out that such interdependencies are inevitable to make the best possible VM migrations between different physical machines and data-centers and to investigate fairness and efficiency of cloud resource allocation mechanisms. Although resource interdependencies are such an important factor for cloud resource allocation, we were the first to present a preliminary model for it.

The energy efficiency of the SmartenIT solutions is addressed by modeling the power consumption of different components. The power consumption of smartphones and
uNaDas (MoNA) can be derived using system utilization samples. Collecting these on the S-Box and applying the respective power models allows minimizing the overall system power consumption. The developed models are used locally to improve the power consumption of participating devices by scheduling traffic to more energy efficient times and locations. Evaluations of the performance of the considered models are in progress.

Next, in Chapter 5, we described the TM mechanisms developed to address the aspects of the OFS: in particular, we provided for each of the three mechanisms, i.e. DTM, ICC and MRA, its intelligence and key influencing factors, the simulation framework employed for its evaluation and the evaluation results so far derived by the relevant simulations.

DTM is a traffic management mechanism that aims at minimization of inter-domain traffic cost in multi-homed AS by influencing the distribution of the traffic among links. DTM works for tariffs based on traffic volume and 95th percentile. Experiments show that DTM can: 1) find such a traffic distribution amongst links so that the total cost is lower, 2) dynamically influence the traffic distribution during the billing period by influencing the network path for manageable traffic. Given some level of unpredictability of traffic passing inter-domain links, it may happen that the optimal traffic distribution (and related cost) will not be achieved but a solution close to optimum is achievable. The intervention potential depends on the share of manageable traffic in overall traffic, traffic characteristics and cost functions.

ICC is a traffic management mechanism deployable by an ISP for utilizing its transit link(s) optimally in terms of cost and performance. ICC reduces the ISP’s 95th percentile transit charge by means of controlling the rate of a portion of this traffic (e.g. delay-tolerant cloud traffic) that has been marked accordingly by the ISP’s business customer (e.g. cloud/datacenter), and shifting its transmission at off-peak intervals, while real-time traffic is still forwarded a la Best Effort. The evaluation of ICC reveals its efficiency in reducing the transit link charge and balancing the load and the usage of the network since non-critical traffic is delayed and transmitted when the network is underutilized.

Experiments indicate that there is a tradeoff between the amount of transit link cost savings and the extra delay incurred for the time-shiftable traffic. Simulations indicate that the degree of discount sought and the variability of the traffic within the billing period are the primary factors affecting the success of the ICC water-filling algorithm and the resulting queueing delays at (peak) intervals where traffic exceeds the target 95th percentile. Also, critical traffic such as real-time or zero-tolerance cloud services traffic under ICC will experience lower congestion, and thus improved performance compared to current Best Effort Internet, thus benefiting the user, the cloud service provider and also the ISP. Last but not least ICC is applicable to Internet interconnection links of today, being in line with the respective traffic granularities, time scales and business agreements.

Fairness in a cloud (federation) and efficiency of VM scheduling and live migration can only be evaluated significantly, when the interdependency of multiple heterogeneous resources (CPU, RAM, etc.) is considered. Such interdependencies are not straightforward to determine and usually not considered in the literature. Furthermore, the first multi-resource fairness metric applicable to such cloud scenarios was developed and will be built into the allocation mechanism that comes with the SmartenIT architecture. Thus, this MRA mechanism is designed to ensure fair resource allocation in clouds and or cloud federations between customers based on multiple resources.
Furthermore, it is planned that the mechanism determines those VMs, which increase cloud performance most when live migrated (not only in terms of the migrated VM but also in terms of performance of those VMs share a host with it before and after migration). Therefore, the mechanism can ensure that inter-AS traffic produced due to live migration of VMs, aids the technical performance of the cloud or federation to the highest possible degree.

Note that the OFS TM mechanisms have assumed Best Effort Internet connectivity so as to maximize the SmartenIT mechanisms’ applicability and potential adoption and impact. However, as service differentiation and premium connectivity services over the Internet gain acceptance and momentum, the SmartenIT mechanisms could be extended accordingly so as to take advantage of such premium connections in the medium term future. In particular, regarding the OFS TM mechanisms: a) ICC could be extended so as to handle more than the two existing (time-shiftable and real-time) classes of traffic according to the respective traffic urgency in order to provide different statistical guarantees to the respective classes of traffic b) DTM could be extended to use multiple tunnel connections, each pertaining to a different service for the same source-destination pair: DTM rationale could optimize the usage of these tunnels while the tunnel features could result in differentiated performance plus bypassing the BGP single source-destination route.

Furthermore, we specified a synergetic solution combining the so-called traffic “shift in space” of DTM and traffic “shift in time” of ICC to achieve further optimization of traffic distribution across multiple transit links while delaying delay-tolerant traffic when transit links are heavily loaded, so as to ultimately achieve even lower transit charges for the ISP than DTM alone. This synergetic solution will be further investigated in the next months, while evaluation results will be provided in D2.5.

In Chapter 6, we focused on TM mechanisms addressing EFS:

RB-HORST is a mechanism running on uNaDas and home routers that aims to leverage its unutilized resources, which are storage space and bandwidth. The storage space is used to provide caches for content delivery that are close to end users. The bandwidth, which is not used at night, is used to prefetch content and store it in the cache. Furthermore, RB-HORST provides WiFi access to trusted users to offload their mobile traffic from 3/4G to WiFi. RB-HORST is deployed by installing the RB-HORST firmware on the home router. For mobile traffic offloading installing the RB-HORST mobile app is required. The Facebook account is used for identity management and to retrieve social information which is used to optimize content delivery by prefetching content of high interest close to end-users. Furthermore, trusted users are identified through the social network, e.g. from a users’ friends lists. RB-HORST-enabled home routers are organized in a content centric overlay network supported by a distributed hash table that is used to bootstrap new routers into the overlay network and store basic information about the routers. The evaluation of RB-HORST shows that it has a high potential to save expensive inter-domain traffic and to take load of ISP caches. The amount of requests served locally increases with the home router sharing probability and reaches 60% of all requests if the ISP cache capacity holds one percent of the video catalogue size.

SEConD (formerly referred to as SECD [5]) is a TM mechanism employing social-awareness, AS-locality awareness, chunk-based P2P content delivery and prefetching, and a centralized node acting as cache, P2P tracker and proxy to improve the QoE of video streaming for users of OSNs. Evaluations of SEConD by means of an event-driven proprietary simulator under realistic scenarios have shown that SEConD indeed achieves
reduction of inter-AS traffic in the potentially expensive transit links, both in total, and "busy" hours. Moreover, evaluations of SEConD by means of an event-driven proprietary simulator under realistic scenarios have shown that SEConD appears also to be very effective in the cases evaluated, since it achieves: a) a high (~87%) reduction of inter-AS traffic in the potentially expensive transit links, both in total, and peak hours, b) high prefetching accuracy (~88%), and c) minimization of the contribution of the origin server (~13%) of total traffic, in the content delivery due to the P2P dissemination. Investigation of the impact of SEConD in multiple ASes of different size (i.e. number of users) reveals that the larger the AS, the lower the contribution of the centralized node (cache) to achieve similar performance. This provides evidence on the scalability of the mechanism.

vINCENT is a mechanism running on uNaDas and home routers aiming to leverage unused wireless resources. The mechanism addresses the problem that the WiFi spectrum (2.4 GHz and 5 GHz) is largely occupied by private networks, thus making this spectrum unusable for the public. Consequently, vINCENT constitutes a system to motivate access point owners to open up their WiFi to the public. Therefore, two issues are addressed: the issue of security and the issue of providing a proper incentive. For the former, vINCENT uses a system of tunnels to trusted endpoints, for the latter, an incentive algorithm assesses the expected performance of access points and grants their owners an appropriate incentive by allowing them to consume a fair share of the offloading capabilities of other access points. Regarding the incentive, the main lesson learnt is that the incentive cannot be granted without taking an access point’s geographical position into account, as differing user densities have a major influence. vINCENT uses regression modelling to circumvent this problem, i.e., the mechanism takes into account geographic location and user densities to calculate the expected performance of an access point and grants an incentive based on the measured performance compared to the expected performance.

MoNA addresses the energy consumption of smartphones. This is achieved by scheduling data transmissions in space and time to reduce the energy expenses on the air-interfaces. Furthermore, the most energy efficient connectivity option is always selected. The scheduling also considers the QoE of the end-user, which likely is affected positively by extending the battery life of the device. Evaluation of MONA by means of simulations is in progress.

MUCAPS adds network layer awareness to decisions on application resource endpoint selection. Such a decision can complement decisions on social and energy awareness derived by other mechanisms proposed in SmartenIT. MUCAPS addresses the EFS, as the ALTO service supported in its current implementation performs the endpoint cost path evaluation. It may also address the OFS if the supported service is the filtered cost map service which evaluates paths between network regions rather than endpoints. Moreover, it performs the automatic decision making to ALTO Clients, allowing them to decide what ALTO Service to request w.r.t. the application and the User Equipment receiving application data. Initial MUCAPS results observation have shown that for a reliable evaluation, it is necessary to introduce contextual elements such as at least the access type and capabilities of the UEP and the network conditions, or simply user expectations on QoE. In addition, the shortest AS-paths are not necessarily the best ones in terms of delay, resources and ISP costs: these three criteria may even be conflicting and a safe way for efficient layer cooperation is to consider them jointly.

Similarly to the OFS, we specified two synergetic solutions for the EFS: a) one combining trust-based home router sharing based on social observations by RB-HORST, and content
caching, prefetching, and chunk-based dissemination by SEConD to address the content placement UC [6], and b) one combining trust-based home router sharing based on social observations by RB-HORST, and incentive-based reciprocation and energy efficiency by vINCENT and MONA, respectively, so as to address the WiFi offloading UC [6]. These two synergetic solutions will be further investigated in the next months, while results obtained by their evaluation will be provided in D2.5, as well.

Finally, in Chapter 7, we continued the discussion initiated in [5], on the coverage of the SmartenIT playfield, as defined by the objectives and key targets described in [1], by the TM mechanisms and models specified and evaluated in this deliverable. We have found that the various TM mechanisms address different objectives, which are mainly complementary and in fewer cases overlapping, while the synergetic solutions described in this deliverable achieve to address a broader area of the SmartenIT landscape. This investigation will be concluded in the upcoming Deliverable D2.5 (final deliverable of WP2), where also cross-scenario synergies will be discussed.

8.2 Next Steps

Future work towards the completion of WP2 includes:

- The revision and update of use-cases of interest to SmartenIT as well as their positioning with respect to the use cases developed by other projects.
- The refinement and further specification of already identified use-cases.
- The further development and the completion of the analysis of theoretical and simulation models so as to evaluate aspects of TM mechanisms and synergetic solutions not studied so far, e.g. inclusion of background traffic or different traffic classes by the evaluation models.
- The completion of the investigation (and tuning) of important parameters and metrics of interest so as to achieve optimal performance of the TM mechanisms, e.g. the $C_{target}$ value in the case of ICC and DTM++.
- The completion of specification and the evaluation of synergetic solutions DTM++ and RBH++ together with the identification of the exact cases for which they are applicable.
- The further evaluation of some of the TM mechanisms. For example, future evaluation includes the assessment of the ICC learning features with respect to the prediction of the expected real time traffic per epoch using traffic statistics. Moreover, the assessment of the WiFi offloading potential and the performance evaluation of RB-HORST under non-stationary conditions is in progress. Advanced caching and prefetching strategies are applied to improve the efficiency of content delivery.

The outcome of these activities will be documented in the forthcoming Deliverable D2.5 “Report on Definition of Use-cases and Parameters” which is due to M30, i.e. April 2015.

Moreover, WP2 will continue to collaborate with WP3 in the specification of the TM mechanisms and synergetic solutions to be implemented as part of the SmartenIT prototype, and with WP4 in the specification of experiments for the trials of the SmartenIT approach and the overall assessment of the evaluation results.
9 Smart Objectives

Throughout this document, nine SmartenIT SMART objectives defined in Section B1.1.2.4 of the SmartenIT DoW [1] have been partly addressed. Namely, two overall objectives as reported in Table 9-1, and seven specific ones as presented in Table 9-2.

The overall Objectives 2 and 3 are defined in DoW as following:

**Objective 2** SmartenIT will develop theory for new pricing and business models, applicable in example use cases, combined with the design and prototyping of appropriate traffic management mechanism that can be combined with today’s and IETF’s proposed relevant communication protocols.

**Objective 3** SmartenIT will investigate economically, QoE, and energy-wise such models in theory by simulations to guide the respective prototyping work in terms of a successful, viable, and efficient architectural integration, framework and mechanisms engineering, and its subsequent performance evaluation.

The work reported in this deliverable cover to a significant extent both of the above objectives. Indeed: In Chapter 3, we perform an in-depth study of appropriate business models, which generated the development of suitable TM mechanisms to address realistic and interesting use-cases. Next, in Chapter 4, a set of theoretical models is defined; these include models employed to characterize content demand, topological distribution of home routers and WiFi hotspots, caching effectiveness, incentive-based resource sharing and workload migration in the case of a cloud federation, VM migration, power consumption in end-user devices and mobile transmissions, and incentive-based sharing of end-user resources.

Then, in Chapter 5 and Chapter 6, the specification of TM mechanisms developed to address the aspects of the OFS and the EFS, respectively: in particular, for each mechanism, its intelligence and key influencing factors, the simulation framework employed for its evaluation and the evaluation results derived by the simulation are provided. Furthermore, in Chapter 7, we advanced the study initiated in [5] on the coverage of the SmartenIT playfield (as defined by the SmartenIT objectives and key targets described in [1]), by the TM mechanisms specified and evaluated in this deliverable, and we investigated potential cross-scenario synergies that constitute a unified SmartenIT approach to address selected UCs of interest (as described in [6]).

Table 9-1: Overall SmartenIT SMART objectives addressed [1].

<table>
<thead>
<tr>
<th>Objective No.</th>
<th>Specific</th>
<th>Measurable</th>
<th>Achievable</th>
<th>Relevant</th>
<th>Timely</th>
</tr>
</thead>
<tbody>
<tr>
<td>O2</td>
<td>Theory design for traffic management mechanisms</td>
<td>D2.4</td>
<td>Design, simulation</td>
<td>Advanced</td>
<td>MS2.4</td>
</tr>
<tr>
<td></td>
<td>Prototyping of traffic management mechanisms</td>
<td>D2.4</td>
<td>Design, simulation, implementation</td>
<td>Complex</td>
<td>MS2.4</td>
</tr>
<tr>
<td>O3</td>
<td>Simulative investigations</td>
<td>D2.4</td>
<td>Analysis, simulation, evaluation</td>
<td>Complex</td>
<td>MS2.2</td>
</tr>
</tbody>
</table>

Chapter 3 includes the outcome of the theoretical investigations performed in Task T2.4 and provided input both to the specification of the TM mechanisms (Task 2.3) e.g. effective
selection of values for parameters, as well as their evaluation (Task 2.5). Chapter 5 and Chapter 6 provide input to WP3 and specifically, the implementation of the SmartenIT prototype components (Task 3.3). Chapter 7 and the investigation on cross-scenario synergies will be continued in the content of the definition of use-cases (Task 2.2) and will be concluded in D2.5, while they also provide input in WP4 and specifically in the definition of experiment scenarios (Task 4.2).

Table 9-2: Specific SmartenIT SMART objectives addressed; excerpt from the set of Tables of [1] with all SMART objectives of the project.

<table>
<thead>
<tr>
<th>Objective ID</th>
<th>Specific</th>
<th>Measurable</th>
<th>Achievable</th>
<th>Relevant</th>
<th>Timely Project Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1.1</td>
<td>How to align real ISP networks, while optimizing overlay service/cloud</td>
<td>Savings in inter-domain traffic (inMbit/s) and possible energy savings</td>
<td>Design, simulation T2.2</td>
<td>Major output of relevance for providers and users</td>
<td>M36</td>
</tr>
<tr>
<td></td>
<td>requirements?</td>
<td>due to optimized management mechanisms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O1.2</td>
<td>What overlay information is needed for the network to optimize its resources and how can this information be conveyed from overlay operators, application providers, or end users to network operators through relevant and viable incentives?</td>
<td>Number of identified types of information (AS affiliation, requested service, legal requirements, energy costs, traffic demand, traffic volume, latency/QoE requirements, social information and meta-information)</td>
<td>Design, simulation T2.3</td>
<td>Highly relevant output of relevance for providers and users</td>
<td>M24</td>
</tr>
<tr>
<td>O1.3</td>
<td>Which incentive schemes will be needed (for application overlays and clouds) to adapt to the existing physical network structure? For what optimization criteria (revenue, energy efficiency) can this be attained more effectively?</td>
<td>Number of identified incentive schemes and their cost reduction in terms of money, traffic, and energy footprint</td>
<td>Design, simulation T2.3, T2.4</td>
<td>Extremely relevant output of relevance for providers and users</td>
<td>M24</td>
</tr>
<tr>
<td>O1.4</td>
<td>Are decentralized traffic management schemes superior to traditional schemes; if yes, why? If not, what is the associated efficiency loss?</td>
<td>Number of identified and tested scenarios for the comparison of the different traffic management schemes</td>
<td>Design, simulation T2.5</td>
<td>Highly relevant output of relevance for providers</td>
<td>M24</td>
</tr>
<tr>
<td>O2.1</td>
<td>Which key design goals (incentive compatibility, user expectations, etc.) are met by the designed mechanisms in terms of their performance?</td>
<td>Number of identified design goals, number of met design goals (evaluated separately for each mechanism)</td>
<td>Design, simulation, prototyping T2.5</td>
<td>Major output of entire project</td>
<td>M36</td>
</tr>
<tr>
<td>O2.2</td>
<td>Which parameter settings are reasonable in a given scenario/application for the designed mechanisms to work effectively?</td>
<td>Number of parameters identified, where a reasonable value range is specified</td>
<td>Design, simulation, prototyping T2.2, T3.4, T4.2</td>
<td>Highly relevant output of relevance for providers and users</td>
<td>M24</td>
</tr>
</tbody>
</table>
Deliverable D2.4 contributes to the answering of four specific Theoretic questions:

1. **Objective O1.1: How to align real ISP networks, while optimizing overlay service/cloud requirements?**

   The ISP-friendliness of the TM mechanism was a prerequisite during the design and development phases. Thus, the alignment of a real ISP network with the TM mechanism can be achieved due to the transit cost minimization/reduction attained by means of most of the OFS TM mechanisms, i.e. DTM or ICC and their synergies, and selected EFS TM mechanisms as well, such as SEConD.

2. **Objective O1.2: What overlay information is needed for the network to optimize its resources and how can this information be conveyed from overlay operators, application providers, or end users to network operators through relevant and viable incentives?**

   An extensive set of key parameters and metrics has been identified for each single mechanism, while cross-layer algorithms have been specified to facilitate the information exchange between ISPs and overlay/cloud (see Chapter 5 and Chapter 6), e.g. the information exchange taking place between the network and cloud layer of ICC, or the AS-awareness for the construction P2P overlays for video delivery in SEConD.

3. **Objective O1.3: Which incentive schemes will be needed (for application overlays and clouds) to adapt to the existing physical network structure? For what optimization criteria (revenue, energy efficiency) can this be attained more effectively?**

   Incentives for resource sharing and load/VM migration among clouds (with or without federation) or load sharing among home routers have been investigated by dedicated models in Chapter 3. The optimization criteria served are minimization of transit charges for the ISP, cloud metrics, QoS/QoE and energy metrics. All TM mechanisms specified do include appropriate incentive schemes.

4. **Objective O1.4: Are decentralized traffic management schemes superior to traditional schemes; if yes, why? If not, what is the associated efficiency loss?**

   Decentralized TM mechanisms have been compared against traditional (centralized) ones and have been found to be more efficient in certain cases (see Chapter 6). For example, SEConD was compared with SocialTube [66] and
traditional client-server dissemination. Both SEConD and SocialTube, which employ P2P overlay networks to assist the content dissemination, have been found to outperform the traditional client-server approach. Moreover, SEConD constitutes a hybrid approach employing both P2P overlays and a centralized node (acting as cache, proxy and P2P tracker), and was proved to outperform SocialTube, which is distributed to a higher extent. Evaluation of SEConD in ASes of varying size (i.e. number of subscribers) proved that the contribution of the P2P overlay is bigger in larger ASes while the contribution of the centralized node is higher in smaller ASes, where the distributed components and resource of the mechanisms do not suffice.

Deliverable D2.4 contributes to the answering of three specific questions on Simulative Evaluation:

5. **Objective O2.1: Which key design goals (incentive compatibility, user expectations, etc.) are met by the designed mechanisms in terms of their performance?**

For each TM mechanism a set of key metrics has been defined which addressed the key design goals for all involved stakeholders including end-users. Some examples of key metrics related to the end-user are: QoE, e.g. in terms of video bitrate or video stalling time, accessibility, energy consumption in user device, improvement of the performance of non-shiftable traffic in ICC etc.

6. **Objective O2.2: Which parameter settings are reasonable in a given scenario/application for the designed mechanisms to work effectively?**

The parameter settings that are reasonable (or preferable) for a given scenario/application are defined in the description of the evaluation framework and the presentation of the evaluation results derived (see Chapter 5 and Chapter 6). Further evaluations to will be carried out in the remaining duration of WP2 will lead to concrete suggestions on parameter value selections.

7. **Objective O2.3: Which of the designed mechanisms should be selected for implementation and field tests within the project?**

A selection procedure for the TM mechanisms to be implemented by WP3 as part of the SmartenIT prototype took place in several stages within Y2 and is reported in Chapter 2. In particular, DTM was selected for implementation in v1.0 prototype release, RB–HORST in v2.0 release, while in release 3.0, selected modules of ICC and SEConD, vINCENT, MONA were integrated with DTM and RB–HORST, respectively, to generate extended synergetic TM mechanisms, the so-called DTM++ and RB–HORST++.

Deliverable D2.4 contributes to the answering of one specific prototyping question:

8. **Objective O3.4: How to monitor energy efficiency and take appropriate coordinated actions?**

Appropriate models for measuring energy consumption in end-user devices and mobile transmissions have been developed (see Chapter 3). These models are used by MONA and can be employed both in OFS or EFS. Also, although not adopting a specific monitoring mechanism for energy efficiency, ICC can incorporate energy-related costs in the metrics to be optimized.

According to the SMART objectives set within SmartenIT DoW[1], those ones of relevance for D2.4 and the respective tasks within WP2, i.e., T2.3, T2.4 and T2.5, the targeted
objectives have either been already met in D2.4 or advanced adequately so as to be met in the remaining duration of the project.
10 References


[38] http://cloudstore.interoute.com/m
[41] http://contrail-project.eu/
cloud_economics_20_german_stock_exchange_deutsche_boerse_starts_cloud_com-
puting_exchange
[47] https://www.ict-etics.eu/
mobile-apps/
[49] Rochwerger, Benny, et al. "Reservoir—when one cloud is not enough." IEEE
computer 44.3 (2011): 44-51.
federation of cloud computing environments for scaling of application services." Algorithms and architectures for parallel processing. Springer Berlin Heidelberg,
International Conference on Cloud Computing, GRIDs, and Virtualization. 2011.
[54] Z. Dulinski and R. Stankiewicz, "Dynamic traffic management mechanism for active
optimization of isp costs," in Workshop on Social-aware Economic Traffic
398–401.
[55] Z. Dulinski, R. Stankiewicz, K. Wajda, “Mechanism for dynamic optimization of inter-
domain traffic cost in multi-homed ISP’s network”, NETWORKS 2014 - 16th
International Telecommunications Network Strategy and Planning Symposium to be
held in Funchal, Madeira Island, Portugal, September 17-19, 2014
[56] Zbigniew Dulinski, Rafał Stankiewicz, Piotr Cholda, Piotr Wydrych, Burkhard Stiller,
Inter-ALTO communication protocol, IETF Draft draft-dulinski-alto-inter-alto-protocol-
00.txt, June 29, 2010,
[57] Zbigniew Dulinski, Piotr Wydrych, Rafał Stankiewicz, Piotr Cholda, Mirosław Kantor,
Inter-ALTO communication problem statement, IETF Draft draft-dulinski-alto-inter-
problem-statement-00.txt, March 4, 2011
[58] Zbigniew Dulinski, Piotr Wydrych, Rafał Stankiewicz, Inter-ALTO communication
problem statement, IETF Draft draft-dulinski-alto-inter-problem-statement-01.txt, July
11, 2011
[59] www.nsnam.org


[62] http://docs.openstack.org/trunk/openstack-ops/content/projects_users.html


[69] Yang, Weilong, and Zhensong Qian. "Understanding the Characteristics of Category-Specific YouTube Videos."


[73] http://www.sysomos.com/reports/youtube


[76] https://developers.facebook.com/docs/facebook-login/access-tokens/

[77] https://developers.facebook.com/docs/graph-api/reference/v2.0/friendlist


[104] https://yourkarma.com/


11 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>AEP</td>
<td>Application Endpoint</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous System</td>
</tr>
<tr>
<td>BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>CPE</td>
<td>Customer Premises Equipment</td>
</tr>
<tr>
<td>DTM</td>
<td>Dynamic Traffic Management</td>
</tr>
<tr>
<td>GRE</td>
<td>Generic Routing Encapsulation</td>
</tr>
<tr>
<td>ICC</td>
<td>Inter-Cloud Communication</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>MONA</td>
<td>Mobile Network Assistant</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multi-Protocol Label Switching</td>
</tr>
<tr>
<td>MRA</td>
<td>Multi Resource Allocation</td>
</tr>
<tr>
<td>MUCAPS</td>
<td>Multi-Criteria Application Endpoint Selection</td>
</tr>
<tr>
<td>OSN</td>
<td>Online Social Network</td>
</tr>
<tr>
<td>OTT</td>
<td>Over-The-Top</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer-to-Peer</td>
</tr>
<tr>
<td>RB-HORST</td>
<td>Replicating Balanced Tracker and Home Router Sharing based on Trust</td>
</tr>
<tr>
<td>SEConD</td>
<td>Socially-aware Efficient Content Delivery</td>
</tr>
<tr>
<td>vINCENT</td>
<td>Virtual Incentives</td>
</tr>
</tbody>
</table>

12 Acknowledgements

Besides the deliverable authors, as indicated in the document control section, this deliverable was made possible with the considerable help of the WP2 team of SmartenIT. Many thanks to all of them! Special thanks go to the internal reviewers Sergios Soursos (ICOM) and Roman Lapacz (PSNC) for detailed revisions and valuable comments.
13 Appendixes

13.1 Detailed specification of DTM

In this appendix, we present the wider vision of the DTM operation, this enables future extensions. There are also presented some limitations of the DTM for inter-cloud communication which results from the BGP routing policies applied by operators of autonomous systems. We present the limited version of the DTM which is a basis for system release 1.0. We discuss potential tests which can be used for verifying the DTM performance. In the separate sections we present algorithms used in DTM mechanism. That are not final versions of algorithms, they will improved in next releases.

13.1.1 Inter cloud communication - multi cloud transfer to single cloud

Traffic from many clouds (or DCs) may be send to a single cloud (or DCs). Figure 13-1 presents a simplification of a general situation where a single DC downloads data from multiple partner DCs. Compensation vectors are used in the TDM mechanism. In this section we want to present the relation between compensation vectors established in different ISP’s domains. Let’s suppose that DC-AA1, DC-AB1, DC-BA1 and DC-BB1 send data to the DC-C.

According to the previously presented procedure the DA routers in different ISP’s domains are connected via tunnels. The S-box in the ISP-C domain delivers an inbound compensation vector to other partner S-boxes in domains ISP-A and ISP-B. This inbound compensation vector originates from the same ISP so it is delivered in the same form to all traffic source ISPs. This compensation vectors are used for inbound level management. The ISP-A can simultaneously apply the outbound level management. The S-box-A in ISP-A calculates the outbound compensation vector and passes it to the SDN-A which
performs flow distribution. An outbound compensation vector is calculated separately by S-boxes in each traffic source ISP. The physical inter domain connections limits management flexibility. The inter domain link topology presented in Figure 13-1 allows to use only an inbound level of management for communication between the DCs in the ISP-B and DC-C (ISP-B possesses only a single inter domain link).

In Figure 13-1, we presented a deployment version in which SDN controller is not collocated with S-box. There is also possible a solution where only one SDN controller is used in a single ISP's domain managing many DA routers simultaneously. We present solution (Figure 13-1) where a separate SDN controller is used by each DA router. In principle for scalability purposes even a single SDN controller can be used by a single server (virtual router running on a server can be perceived as DA router in some special cases) in a data center but this approach touches an internal network of data center. This requires an agreement between an ISP and data center owner.

The proliferation of DA routers results in increase of a tunnel number. A proper DA router allocation can lower a number of tunnels. For instance if the BG router in the ISP-B network plays a role of a DA router, the tunnel number going to the DC-C can be limited to two (only single DA router is necessary). We can also deploy many instances of S-boxes by reason of scalability and traffic report delay.

We present here approach when a common inbound compensation vector is prepared for inbound traffic coming from all DCs located in remote ISP domains. If there is a need for more granular inbound traffic distinction then the separate compensation vector and reference vector can be calculated for some partner ISP domains. For instance an inbound compensation can be separately calculated for the ISP-A and ISP-B (Figure 13-1). The receiving traffic ISP-A can use the same traffic correction strategy for all delivering traffic ISPs. Inbound compensation vectors have the same direction for all delivering traffic ISPs but the length of a compensation vector for each ISP in general may be different. This follows from the fact that each ISP sends different amount of traffic from their DCs. In our example in Figure 13-1 an inbound compensation vectors \( \vec{C}_A^{\text{in}} \) for ISP-A and \( \vec{C}_B^{\text{in}} \) for ISP-B meet condition expressing common direction: 

\[
\frac{\vec{C}_A^{\text{in}}}{|\vec{C}_A^{\text{in}}|} = \frac{\vec{C}_B^{\text{in}}}{|\vec{C}_B^{\text{in}}|}.
\]

The length of these vectors depends on amount of traffic offered by DCs located in particular ISP. Traffic from different DCs can distributed to distinct tunnels and amount of this traffic can be dosed selectively and independently in each ISP domain. The DTM mechanism offers flexibility in this aspect.
13.1.2 Inter cloud communication - single cloud transfer to multi cloud

In Figure 13-2, we present an example where one cloud (or DC) delivers data to many clouds (or DCs). The S-boxes in each receiving ISPs (ISP-A and ISP-B) sends its own inbound compensation vectors to the S-box in ISP-C, these vectors are different for each ISP to ISP connection. They are selectively used for distributing flows to specific tunnels related to respective tunnels. The S-box in the ISP-C establishes one common outbound compensation vector for all connections to other ISPs. In the next step the pair of compensation vectors (inbound and outbound) are applied by DTM mechanism for the respective communication between DCs located in the respective ISP’s domains. In Figure 6 we have two communication relations ISP-C to ISP-A and ISP-C to ISP-B. The inbound compensation vector delivered by the S-box-A is used for distributing traffic to tunnels: T-A-C-1, T-A-C-2, T-A-C-3 and T-A-C-4. The S-box-B sends the inbound compensation vector for spreading traffic amongst all tunnels connecting the ISP-C with ISP-B.

Again there is the same comment related to the common outbound compensation vector, if an operator offering traffic form its DC wants to use different traffic patterns for traffic delivery to different ISPs, it can define a separate outbound compensation and reference vectors for each an ISP-ISP connection.

13.1.3 Traffic distribution procedures

The DTM mechanism for inter cloud communication uses tunnels. The traffic may be distributed amongst tunnels applying one of the two distribution methods: per packet distribution or per flow distribution. Depending on a type of traffic an appropriate distribution method should be selected. Separate tunnels can be established for each distribution method or the same tunnels can be used for both distribution methods. The
per-packed distribution method is more granular, in this method packets from one flow may be distributed amongst many tunnels. If we use per flow distribution all packets from a single flow go the same tunnel. If we apply the DTM only to new coming flows we will have less traffic allocation freedom for per flow selection than for per packet selection.

In the first stage of implementation we plan to apply the DTM with the per flow selection. The per-flow selection will be used for the new coming flows. We also work on the solution that enables switching existing flows between tunnels. This is rather future aspect of the DTM mechanism.

13.1.4 Physical link topology limitation and routing policy limitation

We have mentioned that physical link connections or routing policies in transit domains may not allow to setup such flexible tunnels configuration like this being presented (Figure 13-2). Below we consider the same link topology but this time there are used different routing policies. Some of these policies may be transposed to a situation where some links are absent. We present three different routing policies applied by ISPs which limits variety of the tunnel setup. They are presented in Figure 13-3, Figure 13-5, and Figure 13-7. Figure 13-4, Figure13-6, and Figure13-8 show the possible tunnels in presence of related routing policies.

Figure 13-3 presents the routing policies in the transit domains ISP-C and ISP-D. These polices do not allow announcig routes W, X,Y, Z between transit domains. There are possible only two topologically distinct tunnels presented in Figure 13-4. In this case, in principle both management levels may be used, but they can express contradictory issues of the ISP-A and ISP-B. For instance the outband compensation vector, representing the ISP-B requirements, indicates that traffic should go thorough link B1. Simultaneously the inbound compensation vector, calculated by S-box belonging to ISP-A, indicates link A1. There is no single tunnel traversing both links. The final decision should be undertaken by an operator who owns SDN controller managing a DA router. In our example the ISP-B decides. If the outbound management level is applied, ISP-A requirements may be fullfilled only (at list partially) when both compensation vectors indicate the same tunnel. Otherwise only requirements of ISP-B are fulfilled. The presented example is equivalent to the situation where a physical link between ISP-C and ISP-D is absent.

Other possible routing policy presented in Figure 13-5 enables to establish two tunnels connecting partner ISPs, Figure13-6. The ISP-D announces prefixes W and Y to ISP-C but does not announce prefixes W, X, Y, Z to the ISP-B. The ISP-C may or not announce prefixes X, Z to the ISP-D, this does not make any change.
Figure 13-3: Available Internet path determined by BGP routing policy (example 1).

Figure 13-4: Available tunnels resulting from routing policies (refers to Figure 13-3).
This time there is no room for applying the outbound level management, the only possible management is related to the inbound traffic. Management request formulated by ISP-A does not make any harm to ISP-B outbound traffic because there is only one way out from the ISP-B dictated by the routing table.

Figure 13-7 represents the situation where the ISP-D announces prefixes W, Y to the ISP-C and ISP-B. The ISP-C does not announce prefixes X, Y to ISP-D. Even if these prefixes are sent to ISP-D by ISP-C, the ISP-D does not announce them further to connected ISPs. This way three tunnels are available (cf. Figure 13-8). The ISP-A and ISP-B may agree that
they perform exclusively only inbound traffic or only outbound traffic management. But if we chose one management level, the second level can be used in limited way. For example, if the ISP-B performs outbound level traffic management then if the link between ISP-B and ISP-C has been chosen, the inbound traffic management is possible. But if the link between ISP-B and ISP-D has been chosen then there is no an inbound level traffic management.

![Available Internet path determined by BGP routing policy (example 3).](image1)

**Figure 13-7:** Available Internet path determined by BGP routing policy (example 3).

![Available tunnels resulting from routing policies (refers to Figure 13-7).](image2)

**Figure 13-8:** Available tunnels resulting from routing policies (refers to Figure 13-7).

13.1.5 **Innovation of DTM mechanism for inter-cloud communication**

The DTM mechanism for inter-cloud communication is dedicated for ISPs who hosts clouds or data centers in their domains and they want to cooperate in order to improve traffic distribution on their inter domain links. This mechanism can also be perceived as a
service which is offered to a partner ISP who wants to manage inter domain inbound traffic. This is completely new approach in that sense that management parameters are prepared by an ISP which receives traffic and traffic management procedures are applied by an ISP originating traffic.

The proposed mechanism operates independently from a cloud supervision mechanism that can be also noticed as an innovation. ISPs may apply their own routing policies without influencing cloud management system and clouds can communicate with each other transparently not being affected by the DTM mechanism. The DTM mechanism reacts dynamically to the existing situation on inter domain links. The usage of the proposed mechanism can be beneficiary for a service offeror and offeree.

An offeror can benefit from management of the inter cloud outbound traffic on its inter domain links. In the proposed form of the DTM mechanism for inter cloud communication the SDN control is exploited. Traffic to a selected cloud may come from a few other clouds located in different ISP’s domains. The basic management information is provided by a node (S-box) located in an ISP’s network where a cloud receiving traffic is located. The coordination of SDNs procedures are maintained by a single S-box. This is also innovative procedure because the coordination tasks are done in the foreign domain but the management decision are undertaken by SDN controllers in ISP’s domains where these controllers are deployed. ISPs where DC traffic originates may also perform their own management for outgoing traffic using the DTM procedures. The DTM management procedures for inbound and outbound management are independent.

13.1.6 Allocation of the DTM functionalities to the SmartenIT components

The DTM mechanism uses a few SmartenIT components. These components are deployed on different nodes. We consider a few alternative approaches for component deployment and communication schemas. Below we list used components with allocation of responsibilities for them. We describe used components locating them in dedicated devices.

Network Device

Some network devices participate in the DTM mechanism. There are allocated special functions to these devices in an operator’s network, i.e. BG and DA routers. The majority of operator class routers offers NetFlow functionality, this functionality will be used for inter domain traffic monitoring. BG and DA routers will require NetFlow configuration (NetFlow monitor) and synchronization by NTP. Data from BG and DA routers will be periodically reported to a traffic collector. The reports should contain information about traffic volume transferred via inter domain links and tunnels. If BG or DA routers can’t run NetFlow monitors for some reasons, one can perform port mapping on these routers and send traffic copy to some dedicated devices where NetFlow monitor will run.

BG routers

BG routers are connected to other ASes via inter domain links. The BG functionality is a physical function assigned to selected physical devices possessing inter-domain links. These routers run BGP. Inter domain links must be monitored for the DTM operation. The traffic reports from these links are prepared by BG routers and the reports are transmitted to the Network Analyzer. Let’s consider unidirectional transfer from one data center to another located in different ASes. In the AS hosting DCs being sources of traffic, BG routers monitor outbound traffic. Inbound traffic is monitored by BG routers in
ASes where DCs receiving traffic are located. The NetFlow can be used for link monitoring, a proper collector should be deployed in a S-box.

**DA-routers**

The DA functionality is a logical function and it can be assigned in principle to any network (router or switch) in operator’s network. By proper choice one can reduce tunnel number or influence traffic flow from DCs in an internal operator network. DA routers are point where tunnel ends are attached. DA routers can be used for monitoring outbound traffic leaving DCs and entering tunnels. This measurement enables more precise control of traffic distribution. DA routers may also be used for monitoring traffic leaving tunnels in ASes where receiving traffic DSs reside. This monitoring procedure is to be done using NetFlow. Collected monitoring data should be reported to the Network Analyzer or in some cases may be sent straight to a SDN controller. DA routers are mainly used for traffic distribution. A dedicated SDN controller is assigned for each DA. This is our assumption: one can use one SDN controller for a few DA routers, for instance a single SDN controller can be used for all DAs connecting one DC or cloud with an operator’s network. By choosing a single controller per DA we want to separate management for each DC or cloud on the level of prefixes used by DC. This controller manages flow or packet distribution on this router. The flow or packet assignment to given tunnels is performed on DA routers. SDN controllers communicate with DA routers using OpenFlow.

**S-box (separate physical device - server)**

The S-box is deployed on the dedicated server. The DTM functionalities should be implemented in a few S-box components.

**Network Analyzer**

This component collects traffic reports about inbound and outbound traffic form BG routers. It also may gather information about traffic amount put into tunnels and extracted from tunnels. In this case, DA routers deliver this data to the Network Analyzer. We have marked the part of the Network Analyzer which is responsible for the DTM functionalities as a **NetFow Traffic Collector subcomponent**. We will reuse some standard solution for this collector or implement our own solution. The Network Analyzer must be synchronized by NTP in order to prepare traffic reports gathered during the same time from DA routers. The Network Analyzer gets information about DA routers localization from the DB component. The Network Analyzer passes these reports in a form of current traffic vectors which components represents traffic amount transferred via a selected link during predefined period of time. There are two receivers of these vectors: the Economic Analyzer and the Fixed/Mobile Traffic Manager.

**Economic Analyzer**

This component collects the data expressing traffic amount transmitted through the inter domain links during payment period. It applies selected cost rules for particular links (95th percentile rule or volume rule). It creates a cost map for traffic at the end of the accounting period and predicts the possible optimal traffic pattern (the expected traffic amount on each inter domain link which decreases total cost of transfer via these links). This information forms the basis for calculating a reference vector. Each reference vector component represents expected traffic amount on a particular inter domain link. The inbound and outbound traffic will be treated separately, preparing separate inbound and outbound reference vectors. Both reference vectors will be prepared by those Economic Analyzers which work in ISP’s domains that are able and want to apply
inbound and outbound level management simultaneously. Other ISPs preferring single management level use Economic Analyzers calculating only inbound or outbound reference vectors depending on their needs. Reference vectors may be calculated offline or online using some optimization procedures. These vectors may be calculated once for each new accounting period or may be established many times during this period. This way an operator can correct its traffic predictions and react to extreme traffic changes. We admit that in our experimental setup. The reference vectors will be established only once and will stay constant for each new accounting period. The reference vector is sent to the respective Fixed/ Mobile Traffic Manager. If the inbound management is applied, the inbound reference vector is passed to the remote Fixed/ Mobile Traffic Manager. For the outbound management, the outbound reference vector is sent to the local Fixed/ Mobile Traffic Manager.

The Economic Analyzer receives information about traffic amount from the Network Analyzer which collects data from network nodes in a local AS. Only some extracted information like reference vectors may be sent to a remote partner AS. The part of software implementing TDM functionalities in the Economic Analyzer is contained in the **DTM Economic Analyzer** subcomponent.

**Fixed/ Mobile Traffic Manager**

In the SmartenIT project the Fixed/ Mobile Traffic Manager is used by a few mechanisms. For the DTM we placed all required functionalities in a dedicated **DTM subcomponent** of the Fixed/ Mobile Traffic Manager. We describe the role of this component in two contexts, namely a data sender and data receiver.

We use an example ISP communication for better explanation. Let’s consider two ISPs, the ISP-A and ISP-B. Figure 13-9. In the ISP-A there are located the DC-A1 and DC-A2 that receives data from the DC-B1 and DC-B2 located in the ISP-B domain. Each considered ISP’s domain exchange data via two inter domain links (ISP-A uses A1, A2 and ISP-B uses B1, B2). In ISP-A the DTM subcomponent in the S-box (S-box-A) functions as a data receiver and in the ISP-B the separate DTM subcomponent plays a role of a data sender, so all mentioned DCs in the ISP-B only send data to DSs in the ISP-A.

The Fixed/ Mobile Traffic Manager functioning as a data receiver receives periodically a current inbound traffic vector from the Network Analyzer. The Economic Analyzer sends the inbound reference vector to the Fixed/ Mobile Traffic Manager. These two vectors are used by the DTM subcomponent for the compensation vector calculation. The calculation is repeated always when a new current inbound traffic vector comes to the DTM subcomponent. When a new compensation vector is ready, it is sent to the partner DTM subcomponent in the S-box in the ISP-B (S-box-B). Also the reference vector is sent to the partner DTM subcomponent when it gets a new value. These functions are performed by the DTM subcomponent being a data sender. The inter ALTO protocol is a candidate for communication between S-boxes.
Figure 13-9: The network nodes taking part in the DTM communication and data transfer between partner ISPs.

The Fixed/Mobile Traffic Manager working as a data sender is placed in an ISP domain where distribution of traffic originating from DCs is done (DC-B1 and DC-B2). It supports inbound and outbound management levels. For the inbound management level the S-box (S-box-B) receives an inbound reference vector and a related inbound compensation vector from an ISP where traffic is directed. These two vectors are passed to SDN controllers. In our example the Fixed/Mobile Traffic Manager in ISP-B receives information about inbound traffic from the S-box-A. These vectors are sent to SDN controllers in the ISP-B (SDN-A and SDN-B). That is all what is done by the Fixed/Mobile Traffic Manager in the inbound traffic management procedures. When the outbound level management is used in the ISP-B, the Fixed/Mobile Traffic Manager carries out almost the same procedures like that undertaken by acting as a data receiver. It receives a current outbound traffic vector from the Network Analyze IN ISP-B and an outbound reference vector from the Economic Analyzer in the ISP-B. The Fixed/Mobile Traffic Manager calculates an outbound compensation vector and passes it to SDN controllers (SDN-B1 and SDN-B2). The outbound reference vector is also sent to the same SDN controllers.

When more precise management procedures are needed and amount of traffic put to particular tunnels is reported to the Network Analyzer from DA routers (the DA-B1 and DA-B2 for the outbound level management, the DA-A1 and DA-A2 for the inbound level management), the Fixed/Mobile Traffic Manager takes into account these traffic reports delivered by the Network Analyzer. In this case the compensation vector calculation procedure may switch the compensation procedure on SDN-controllers when the traffic pattern on inter domain links agrees with the reference vector. Without traffic reports from DA routers the same compensation vector is used until new current traffic vector appears is the Fixed/Mobile Traffic Manager and new compensation vector is calculated. That procedure is repeated periodically with predefined period. The traffic reports from DA routers enable reacting in the meantime and switching off the compensation.

DB
The **DTM-DB** is a part of the DB in the S-box storing information necessary for DTM operation. This database should store information:

- a remote (partner) S-box address,
- a set of DA routers identifiers in local AS,
- a set of served networks (prefix/prefix length) in local AS accessible via a particular DA,
- a set of served networks (prefix/prefix length) in remote AS accessible via a particular DA,
- a set of tunnel identifiers which connect a given set of local served networks with a selected set of remote served networks
- an identifier of SDN controller managing a particular DA router or routers in a local AS,
- information expressing relation: tunnel identifier – local inter domain link identifier - remote inter domain link identifier.

Identifiers in most cases are to be IP addresses of tunnel ends or nodes.

The DB offers this data to appropriate SDN-controllers. We want to admit that the DTM-DB is only needed for a S-box in an ISP network where source DCs are located, if there is no need for precise compensation procedure. In this domain SDN controllers are deployed which perform traffic distribution into appropriate tunnels.

When we consider more precise traffic management procedures mentioned in the section *Fixed/ Mobile Traffic Manager*, this database will be used for the recognition of tunnels traversing respective inter domain links in both ISP domains (sending and receiving traffic). In that case the DB will be used by the network Analyzer.

**SDN controller**

This component is responsible for defining flow or packet distribution criteria. The northbound interface is used for communication with the DB component and with the Fixed/ Mobile Traffic Manager. The DTM-DB provides the information about the inter DC relations together with tunnel assignment. The Fixed/ Mobile Traffic Manager sends inbound and outbound reference vectors and also inbound and outbound compensation vectors. If traffic reports from tunnels from DA routers are taken into account, orders for switching off compensation procedures are also sent via the northbound interface.

For selected sets of served networks in local and remote ASes a specific flow patterns or packet patterns are prepared and assigned to dedicated tunnels. An appropriate flow or packet number is to be sent to a selected tunnel according to compensation and reference vector component values. This information about the flow/packet patterns with rules for tunnel assignment are delivered to DA routers managed by the related SDN-controller. The southbound interface uses the OpenFlow protocol. DA routers perform flow/packet switching tasks.

A SDN controller can be deployed as a standalone solution (an independent physical server) or can be collocated with a S-box. In the latter case if a few instances of a SDN controller are required, SDN controllers should be launched on a separate virtual machines on the same physical machine.

**NTP server**
Traffic reports from different inter domain links in one AS should express data collection gathered during the same period of time and the same moment of time. The same comment relates to the reports about traffic sent to tunnels on a single DA router. The compensation procedure relates to a movement of traffic from one link to others so this procedure also requires time synchronization. All devices: network nodes, a S-box and SDN controllers in a single AS should be time synchronized in this AS domain. In each AS an independent NTP-server may be used.
13.2 Simplified DTM for inter-cloud communication
(The subset of the DTM functionalities for the first system release.)

In this section, we present communication pattern for the DTM mechanism for a very simple case presented in Figure 13-10. A single DC-A located in ISP-A receives traffic from a single DC-B which is hosted by ISP-B. We assume that routing policies in the Internet allow to establish only two tunnels between these partner ISPs, the situation is presented in Figure 13-10.

![Simplified DTM for inter-cloud communication](image)

Figure 13-10: The simplified DTM for inter cloud communication - network topology, network nodes and tunnels (system release 1.0).

This assumption does not limit in any way communication between DCs. As it has been previously explained, the DTM mechanism does not influence on the cloud/datacenter communication. In this specification, we consider only the inbound level management mechanism. The presented topology (single homed ISP-B) enables only this type of DTM management. In this approach traffic sent by the DC-B to the DC-A can be transferred via link A1 or link A2 depending on the ISP-A requirements. Two tunnels T1 and T2 are used for traffic distribution going from DC-B to DC-A. The main ideas of DTM operation has been explained in 5.1.

In the mentioned text the concept of reference and compensation vectors has been discussed. A reference vector is related to the accounting period, for each new accounting period a new reference vector must be established. This vector represents predicted, cost optimal traffic amount on each inter domain link. A compensation vector represents traffic amount which should be moved from one link to another (our simplified DTM case) in order to be in accordance with a reference vector. The correction is performed many times during accounting period (we assume that preparation of a new compensation vector every will be feasible every 30 seconds).

In this simplified mechanism version we use a volume charging pattern for accounting period. In the first release of the SmartenIT system we will use GRE tunnels. The tunnels T1 and T2 start at DA-B router/switch and terminate at DA-A router but they traverse different inter links (this is crucial point of the DTM mechanism for inter cloud communication). Below, we describe the notation which is used in the rest part of the DTM description.
\( n \) – inter domain link number

\( T \) – accounting period (time in seconds)

\( \Delta t \) – report period - time in seconds (report means NetFlow report send by selected nodes, it contains information about traffic volume sent during period \( \Delta t \)),

\( N = \frac{T}{\Delta t} \) – number of reports during accounting period,

\( t_i \) – the moment of time when collecting data for a report \( i \) has been started (\( t_{i+1} = t_i + \Delta t \), \( i = 1, ..., N \))

\( f_s(x) \) – cost function for link \( s \), argument \( x \) represents total traffic volume transferred via link \( s \) during accounting period, an example cost functions are presented in figure 12. In our case cost represents monetary value. We assume that cost functions are continuous function which are linear on each interval \([\alpha_{ij}, \alpha_{ij+1}]\), where \( \alpha_{i0} = 0 \), \( \alpha_{i3} = \infty \).

\( \overline{X^{\text{in}}}(t_i) \) – current inbound traffic vector, where \( i \) refers to report from the period \( t_i + \Delta t \) (and each vector component represents traffic on a selected inter-domain link from period \( t_i + \Delta t \)),

\( \overline{R^{\text{in}}} \) – inbound reference vector, each vector component represents optimized expected total inbound traffic on the selected inter-domain link in the next accounting period, this vector is calculated by the Economic Analyzer (for DTM purposes it is represented by its subcomponent – DTM Economic Analyzer), the reference vector concept has been explained in Section 5.1.1.

\( \overline{Z^{\text{in}}}_{DA-A}(t_i) \) - current inbound DC traffic vector via selected attached tunnels, (the \( DA-A \) index identifies the \( DA-A \) router where the ends of respective tunnels are attached and measurement is performed), each vector component represents sum of traffic going via a group of tunnels traversing a respective single inter domain link (A1 or A2), where \( i \) refers to report from the period \( t_i + \Delta t \) (each vector component represents traffic volume going via all tunnels traversing respective inter domain link during period \( t_i + \Delta t \)).

\( \overline{C^{\text{in}}}(t_i) \) – inbound compensation vector, components of this vector are calculated by the Fixed/Mobile Traffic Manager (for the DTM purposes it uses its subcomponent which name DTM subcomponent), the concept of compensation vector has been explained in Section 5.1.1.

![Figure 13-11: Cost functions.](image-url)
In a separate chapter we present algorithms used by the DTM mechanism, only the simplest case presented in Figure 13-10 is considered and this part will be implemented in the scope of the initial system release. The ISP-A possesses two inter domain links. In this domain the single DC-A connected to a single DA-A router downloads traffic from a single DC-B connected to a single DA-B router located in the ISP-B domain. For the presented topology (Figure 13-10) only inbound level traffic management is possible. The outbound management will be introduced in next releases of the SmartenIT system.

Data format:

\[ n \] – integer

\[ T \] – long integer (time in seconds)

\[ \Delta t \] – integer

\[ f_s(x) \] : linear function \( a_{sj}x + b_{sj} \) on each interval \( [a_{sj}, a_{sj+1}] \),

represented in database by a n-tuple \( (k, a_{s0}, a_{sj}, a_{s1}, b_{s1}, \ldots, a_{sk}, a_{sk}, b_{sk}) \), where \( k \) represents number of intervals

\[ k \] – integer, \( a_{si} \) – float, \( b_{si} \) – float

\[ X_i^m(t_i), R_i^m, Z_{DA-A}^m(t_i), C_i^m(t_i) \] each vector with \( n \) components (in our simple case \( n = 2 \)), each component represented by long integer

We use IPv4 address format, all addresses and net masks will be stored in 4 byte format or in dotted decimal notation.

13.2.1 Initial manual configuration in the ISP-A domain

NTP synchronization, NetFlow traffic monitoring and SNMP data collection (SNMP is optional)

1. The ISP-A configures interfaces for the T1 and T2 tunnel end points at DA-A router. IP addresses from the X and Y address pools (X and Y represent the network prefixes announced by BG routers) are used for addressing the ends of T1 and T2 tunnel respectively (these addresses represent destination address of a tunnel, tunnels are set up by the ISP-B).

2. The ISP-A configures a routing protocol which enables communication of the DA-A router with BG-A1 and BG-A2 routers (X and Y network prefixes are announced in the ISP-A domain, concrete network IP addresses will be established in the scope of BG routers configuration procedure).

3. The ISP-A defines routing policy on router BG-A1 which announces the X network on link A1.

4. The ISP-A defines routing policy on router BG-A2 which announces the Y network on link A2.

5. The ISP-A sends information about the tunnel ends to the ISP-B.

6. The ISP-A registers in the DTM-DB (a subcomponent of the DB used by the DTM mechanism) following data:

   a. IP addresses of tunnel ends and some tunnel identifiers if they are used by management procedures,
b. networks used by the DC-A and DC-B (the DC-A and DC-B network prefixes are used for traffic identification and filtering, a source IP address of any host in the DC-B enables filtering traffic from the DC-B),
c. cost functions for inter-domain links (functions $f_1(x)$ and $f_2(x)$),
d. $T$, $\Delta t$ parameters,
e. the IP address of the S-box-B, the address which is used for setting up communication session between S-box-A and S-box-B


13.2.2 Initial communication between components in the ISP-A domain

Some subcomponents of the S-box-A have to acquire selected information from the DTM-DB. The required communication is presented on the diagram below, Figure 13-12. The command get_infoset()is used by the NetFlow Traffic Collector (a subcomponent of the Network Analyzer used by DTM mechanism), this way the NetFlow Traffic Collector acquires information about networks in partner DCs and tunnel parameters (a set of network IP addresses used by communicating DCs located in the ISP-A and ISP-B, tunnel identifiers if these identifiers are used). This information is necessary for composing traffic filters on DA and BG routers.

The cost functions are used by the DTM Economic Analyzer (a subcomponent of the Network Analyzer used by DTM mechanism) for the reference vector calculation (command get_link_cost_function()). The DTM subcomponent of the Fixed/Mobile Traffic Manager is responsible for communication with partner S-box-B. The DTM gets the S-box-B IP address using command get_S_box_address().

![Diagram of initial communication between SmartenIT components in ISP-A domain.](image-url)

Figure 13-12: Initial communication between SmartenIT components in ISP-A domain.
13.2.3 Initial manual configuration in the ISP-B domain:
1. The ISP-B receives information about tunnel ends from ISP-A.
2. The ISP-B creates the T1 and T2 tunnels.
3. The ISP-A registers in the DTM-DB following data:
   4. tunnel identifiers $Tun_{id}_1$, $Tun_{id}_2$ (each tunnel identifier can be represented by a pair of IP addresses of tunnel ends – source and destination tunnel end),
   5. networks used by the DC-A and DC-B (the DC-A and DC-B network prefixes are used for traffic identification and filtering, a destination IP address of any packet is to be used for flow distribution by OpenFlow switches/routers and SDN controllers)
   6. the SDN controller IP address
4. The ISP-B enables OpenFlow operation on DA-B router and sets up a supervision of a selected SDN controller.
5. The ISP-B assigns the DA-B switch/router to be controlled by the SDN-B controller.

13.2.4 Initial communication between components in the ISP-B domain
Some subcomponents of S-box-B have to acquire selected information from DTM-DB. The required communication is presented on the diagram below, Figure 13-13.

The Traffic Distributor being a part of SDN controller needs information about networks in DC in partner ISP-A and network prefixes used by DC-B in local AS. The Traffic Distributor needs also information about tunnel parameters that will be used for traffic redirection. The get_infoSet() command is used for these purposes.

![Figure 13-13: Initial communication between SmartenIT components in ISP-B domain.](image)

13.2.5 Component communication
In Figure 13-14 and Figure 13-15, we present a communication sequence between components involved in the DTM mechanism for inter cloud traffic management. The whole system periodically performs a sequence of operations. We consider time slots of the $\Delta t$ duration (in Figure 13-14 and Figure 13-15 denoted by $dt$). The respective nodes localized in the ISP-A network are time synchronized (the time pre-synchronization is presented in Figure 13-12).

At the beginning of each time slot each BG router starts registering the amount of inbound traffic transferred via all inter domain links. In our simple topology each BG router is connected to the Internet by a single inter domain link. After period $\Delta t$, the collected traffic volume for each monitored link is to be ready for delivery to the DTM NetFlow Monitor. The same task is performed by DA routers which monitor incoming traffic from tunnels.
The data collecting process for a report from time period $t_i$ is depicted by command prepare_report($t_i$) (ti stands for $t_i$ in figure 15 and 16). This process has started at time $t_i$ and has finished at time $t_i + \Delta t$. It starts again for the next period at the point of time $t_{i+1}$. The data collecting process is running on BG-A1, BG-A2 and DA-A routers separately, that are NetFlow collecting processes (on each device the NetFlow is running). These processes run in parallel and independently on all mentioned nodes (DA-A, BG-A1 and BG-A2 routers). They start at the same moment of time on each node (DA-A, BG-A1 and BG-A2 routers are time synchronized by NTP) and the reports are exported periodically at the same moment of time from all mentioned nodes.

While the DA and BG routers counts bits for report for new time slot, the report for previous time slot is exported on time scheduled basis (just after every $\Delta t$=30 sec), the command export_report() in Figure 13-14. The NetFlow Traffic Collector gathers reports from DA and BG routers from previous time slot. When SNMP optional communication is applied the get_traffic_report() commands are used for gathering this data from all monitored inter domain links and all monitored tunnels, Figure13-15. Figure 13-14 differs from Figure13-15 only in the part that is related to the traffic report gathering procedure.

When all reports are collected, the NetFlow Traffic Collector notifies the Economic DTM Analyzer and DTM about availability of these reports (command notify()). The DTM Economic Analyzer requests the reports from the NetFlow Traffic Collector (commands get_vector(X_vector) and get_vector(Z_vector)). The X_vector represents $\overrightarrow{X}^{in}(t_i)$ and Z_vector represents $\overrightarrow{Z}^{in}_{DA-A}(t_i)$ in command arguments. During the whole accounting period the DTM Economic Analyzer calculates total traffic for each inter domain link and total traffic received via tunnels, this numbers are represented by $\overrightarrow{X}^{V}(t_i)$ and $\overrightarrow{Z}^{V}(t_i)$ vectors ($\overrightarrow{X}^{V} = \sum_{i=1}^{N} \overrightarrow{X}^{in}(t_i)$, $\overrightarrow{Z}^{V} = \sum_{i=1}^{N} \overrightarrow{Z}^{in}_{DA-A}(t_i)$). This operation is depicted in the Figure 13-14 by prepare_total_volume() command.

A new compensation vector $\overrightarrow{C}^{m}(t_i)$ has to be calculated for each time slot before the end of accounting period (command calculate_compensation_vector()). C_vector stands for $\overrightarrow{C}^{m}(t_i)$ in Figure 13-14. This vector is sent by the DTM in S-box_A to the DTM in S-box_B (send(C_vector) command). The inter S-box communication can be done via the inter-ALTO protocol or any other communication mechanism, for instance Jason based communication.

Next, the DTM in S-box-B sends $\overrightarrow{C}^{m}(t_i)$ vector to the SDN controller. For each packet entering the DA-B router/switch a flow table is consulted. If this packet belongs to the existing flow it is forwarded according to the rule in the flow table. If there is no match for this packet in the flow table, the packet is directed to the SDN controller (in Figure 13-14 and Figure13-15 depicted by command OpenFlow(packet). Counters in the DA-B router are read by the SDN controller. They provide information about the amount of traffic directed to respective tunnels. The SDN controller works out the new flow rule for this packet based on the counters state and value of the $\overrightarrow{C}^{m}(t_i)$ and $\overrightarrow{R}^{m}$ vectors. The flow rule establishment procedure is enclosed in the traffic distribution algorithm presented in section “Traffic distribution algorithm for the Traffic Distributor in the S-box-B”. The whole rule establishment procedure is contained in command flow_rule_establishment() (Figure 13-14 and Figure13-15). A new flow rule is established only for packets which cannot be assigned to the existing flows on the DA-B switch/router. If a packet entering the DA-B router matches a flow in the flow table it is served according to the existing rule in the flow table, otherwise a packet is sent to the SDN-B controller via OpenFlow (command...
OpenFlow(packet)) and a new rule is prepared (flow_rule_establishment() is used). The new rule is delivered to the Switching/Forwarding component using OpenFlow communication: OpenFlow(new_rule).

When accounting period expires, a new $\overrightarrow{R^{int}}$ should be calculated before $\overrightarrow{C^{int}}(t_i)$ vector is established. The end of accounting period appears when $t_i \mod T' = 0$. When this condition is fulfilled a new reference vector is calculated (command calculate_reference_vector()) and delivered to the DTM (update_reference($R_{vector}$)). The DTM from this moment follows procedures similar that done in previous time slots. The only difference appears in parameters of commands: send($C_{vector}$, $R_{vector}$) and distribute($C_{vector}$, $R_{vector}$). The $\overrightarrow{C^{int}}(t_i)$ vector must be delivered periodically every $\Delta t$ but the $\overrightarrow{R^{int}}$ is to be sent with period $T'$. 
Figure 13-14: The DTM operation sequence diagram with only NetFlow link monitoring.
Figure 13-15: The DTM operation sequence diagram with SNMP and NetFlow link monitoring.
13.2.6 Reference vector calculation algorithm for DTM Economic Analyzer subcomponent in the S-box-A:

**Input data**

\( f_1(x) \) and \( f_2(x) \) from the DTM-DB in S-box-A, example cost functions in figure 12, parameters from the DTM-DB: \( T, \Delta t, \)

\( \overrightarrow{X_{int}}(t_i) \) vector is periodically retrieved from the NetFlow Traffic Collector (with \( \Delta t \) period),

\( \overrightarrow{Z_{DA-A}}(t'_i) \) vector is periodically retrieved from the NetFlow Traffic Collector (with \( \Delta t \) period and \( t_i=t'_i \)),

\( Tun_{id_1}, Tun_{id_2} \) – tunnel identifiers acquired from the DTM-DB,

\( tol_1, tol_2 \) - parameter which allows to influence on reference vector prediction by decreasing or increasing values for \( \overrightarrow{Z} \)

**Algorithm for the inbound traffic reference vector calculation for the total volume traffic cost patter (two-dimensional case).**

The two first steps are repeated for each accounting period \( T \) in the first time slot of the next accounting period.

1. Calculate total volume traffic vector:
   \( \overrightarrow{X} = \sum_{i=1}^{N} \overrightarrow{X_{int}}(t_i) \).

2. Calculate total volume traffic vector received from the DC-B collected on DA-A router:
   \( \overrightarrow{Z} = \sum_{i=1}^{N} \overrightarrow{Z_{DA-A}}(t_i) \).

Next steps (from 3 to 9) are done just after each accounting period \( T \) elapses.

3. Modify components of the measure vector \( \overrightarrow{Z} \)
   \( \overrightarrow{Z_{v}} = (Z_{1v}^{v} tol_1, Z_{2v}^{v} tol_2) \)

4. Prepare vectors for the DC traffic manipulation freedom:
   \( \overrightarrow{S_1} = (-Z_{2v}^{v}, Z_{1v}^{v}) \),
   \( \overrightarrow{S_2} = (Z_{2v}^{v}, -Z_{1v}^{v}) \).

5. Identify areas for the optimization procedure.

The Figure 13-16 presents an identification idea in the 2-dimensional case.
The area scope follows from the cost functions presented in Figure 13-11. In Figure 13-16, we have depicted the $\vec{X}^V$ vector. The $\vec{S}$ and $-\vec{S}$ vectors determine the line on which an optimal reference vector end must lay: $(x_1 + x_2 = X_1^V + X_2^V)$. We can look for optimal value on the mentioned line between points indicated by the $\vec{S}$ and $-\vec{S}$ vectors. From Figure 13-16 follows that one should look for optimal cost in areas: $A_{13}$, $A_{23}$, $A_{22}$.

The area scope follows from the cost functions presented in the Figure 13-11. In Figure 13-16 we have depicted the $\vec{X}^V$ vector. The $\vec{S_1}$ and $\vec{S_2}$ vectors determine the line on which an optimal reference vector end must lay: $(x_1 + x_2 = X_1^V + X_2^V)$. We can look for optimal value on the mentioned line between points indicated by the $\vec{S}$ and $-\vec{S}$ vectors. From figure 3 follows that one should look for optimal cost in areas: $A_{13}$, $A_{23}$, $A_{22}$.

Detailed procedure:

5a. Identify sets:

$$E_i = \{\alpha_{ij}|X_i^V + \min\{S_{1i}, S_{2i}\} < \alpha_{ij} < X_i^V + \max\{S_{1i}, S_{2i}\} for i = 1,2 \text{ and } j = 0, \ldots, 3\}.$$  

where $\alpha_{i0} = 0$ and $\alpha_{i3} = \infty$. In our example $E_1 = \{\alpha_{11}\}$, $E_2 = \{\alpha_{22}\}$ for our cost functions.

(sets expressed separately: $E_1 = \{\alpha_{1j}|X_1^V + S_{1i} < \alpha_{1j} < X_1^V + S_{2i} \text{ for } j = 0, \ldots, 3\}, E_2 = \{\alpha_{2j}|X_2^V + S_{22} < \alpha_{2j} < X_2^V + S_{12} \text{ for } j = 0, \ldots, 3\}$)

5b. For each $i$:

5ba. If $E_i \neq \emptyset$ then:

identify intervals $D_{ij}$ which will be used for optimization based on the $E_i$ set. For the chosen $i$ for which $\alpha_{ij} \in E_i$, we take into account intervals $D_{ij} = (\alpha_{i,j-1}, \alpha_{i,j})$ and $D_{i,j+1} = (\alpha_{i,j}, \alpha_{i,j+1})$. In our example we obtain following intervals: for $i=1$ we have $D_{11}$, $D_{12}$, and for $i=2$ we obtain $D_{22}$, $D_{23}$.
5bb. If $E_i = \emptyset$ then

Identify an interval $D_{ij}$ containing $(X_1^V + \min\{S_{1i}, S_{2i}\}, X_1^V + \max\{S_{1i}, S_{2i}\})$,

Find $\alpha_{i,j}$ that $\alpha_{i,j} < X_1^V + \min\{S_{1i}, S_{2i}\}$ and $X_1^V + \min\{S_{1i}, S_{2i}\} < \alpha_{1,j+1}$

5bc. If $E_1 \neq \emptyset$ and $E_2 \neq \emptyset$ then

For each $\alpha_{i,j} \in E_1$ find $D_{2k}$ and for each $\alpha_{2,j} \in E_2$ find $D_{1k}$:

$x_2 = X_1^V + X_2^V - \alpha_{1,j}$, identify $D_{2k}$ that $x_2 \in D_{2k}$,

$x_1 = X_1^V + X_2^V - \alpha_{2,j}$, identify $D_{1k}$ that $x_1 \in D_{1k}$.

In our example for $\alpha_{11}$ we obtain $D_{23}$ and for $\alpha_{22}$ we have $D_{12}$.

5c. For all $D_{1i}, D_{2j}$ established in step 5b:

Define areas $A_{ij}$ which will be used for optimization based on $D_{ij}$
identification: $A_{ij} = D_{1i} \times D_{2j}$, where $i, j$ are related to the previous steps of the algorithm. In our example we should use areas: $(D_{11} \cup D_{12}) \times D_{23} = A_{13} \cup A_{23}$ and $D_{12} \times (D_{22} \cup D_{23}) = A_{22} \cup A_{23}$. The only distinct areas are $A_{13}, A_{23}, A_{22}$.

5d. Calculate minimum for the $f_1(x_1) + f_2(x_2)$ function using simplex or interior point method in each area established in point 7 with constrains:

$x_1 + x_2 = X_1^V + X_2^V$,

$X_1^V + \min\{S_{1i}, S_{2i}\} < x_i < X_1^V + \max\{S_{1i}, S_{2i}\}$ for $i = 1, 2$.

The $x_1$ and $x_2$ values representing minimum are components of the reference vector: $\vec{R} = (x_1, x_2)$. We obtain a separate reference vector for each area respectively: $\vec{R}_{13}, \vec{R}_{23}, \vec{R}_{22}$.

5e. Choose the smallest reference vector from these calculated in the previous step: $\vec{R}^{\text{min}}$.

5f. End

Output data

$\vec{R}^{\text{min}}$ vector is available for the DTM subcomponent in S-box-A.

The whole procedure can be expressed in a form of linear programing problem which should be solved in each area $A_{13}, A_{23}, A_{22}$:

for $A_{i,j}$

objective function (minimum): $f_1(x_1) + f_2(x_2)$,

constrains:

$\alpha_{1,i} < x_1 < \alpha_{1,i+1}$,

$\alpha_{2,j} < x_2 < \alpha_{2,j+1}$,

$x_1 + x_2 = X_1^V + X_2^V$,

$X_1^V + S_{1i} < x_1 < X_1^V + S_{2i}$,

$X_2^V + S_{22} < x_2 < X_2^V + S_{12}$.
13.2.7 Compensation vector calculation algorithm for DTM subcomponent in the S-box-A:

**Input data**

\( \overrightarrow{R} \) vector acquired from the DTM Economic Analyzer with period \( T \),

\( \overrightarrow{X} \) vector acquired periodically (with period \( \Delta t \)) from the NetFlow Traffic Collector,

**Compensation vector calculation algorithm**

1. for (each time slot \( (t_i, t_i + \Delta t) \))
   \[
   \overrightarrow{C}(t_i) = \frac{\sum_{k=1}^{n} x_k(t_i)}{\sum_{k=1}^{n} r_k} \overrightarrow{R} - \overrightarrow{X}(t_i)
   \]

**Output data**

\( \overrightarrow{C}(t_i) \) vector prepared for sending it to the DTM subcomponent in the S-box-B via inter ALTO protocol or other communication mechanism.

13.2.8 Traffic distribution algorithm for the Traffic Distributor in the S-box-B (in SDN controller or DTM subcomponent)

**Input data**

\( \overrightarrow{r} \) - normalized reference vector delivered by the DTM being part of the S-box-B via a REST communication, where \( \overrightarrow{r} = \frac{\overrightarrow{C}}{\sum_{k=1}^{n} r_k} \). In the algorithm below \( r[k] \) refers to component \( k \) of the \( \overrightarrow{r} \) vector.

\( \overrightarrow{C}(t_i) \) vector delivered by the DTM in S-box-B via a REST communication. In the algorithm below \( C[k] \) refers to component \( k \) in time slot \( t_i \) of the \( \overrightarrow{C}(t_i) \) vector.

**Algorithm for traffic distribution to tunnels (2-dimensional case)**

1. \( r[1]=1; \)
2. \( r[2]=1; \)
3. for( each \( t_i \) )
4.    \( \text{tunnel}=0; \)
5.    \( \text{compensate}=false; \)
6.    if(\( C[1]>0 \))
7.        \( \text{tunnel}=1; \)
8.        \( \text{compensate}=true; \)
9.    
10.   if(\( C[2]>0 \))
11.      \( \text{tunnel}=2; \)
12.      \( \text{compensate}=true; \)
13.    
14.   counter_start[1]=read_counter(1);
15.   counter_start[2]=read_counter(2);
16.   for(each new packet) 
17.       if(\( \text{compensate} \))
18. if(read_counter(tunnel) - counter_start[tunnel]< 

\[ C(tunnel)/r_{inv}(tunnel) \] 

19. send_SDN_rule(tunnel); 
20. else { 
21. for(k=1 to 2) 
22. counter_start[k]=read_counter(k);#przestawione 
23. send_SDN_rule((tunnel mod 2) +1); 
24. else 
25. compensate=false; 
26. else 
27. counter_start[1]; 
28. traffic_DA-B = traffic_1 + read_counter(2) - counter_start[2]; 
29. if(traffic_1/ traffic_DA-B <= r[1]) 
30. send_SDN_rule(1); 
31. else 
32. send_SDN_rule(2); 
33. } 
34. } 
35. }

Output data

a flow rule which is sent to the Switching/Forwarding component via OpenFlow communication.

13.2.9 Test plan

We plan to test the DTM for cloud communication on two levels. The first level of testing is related to components involved in the mechanism. Following components will be tested:

- NetFlow Traffic Collector,
- Economic Analyzer,
- Fixed/Mobile Traffic Manager,
- Traffic Distributor.

The NetFlow Traffic Collector time synchronization and scheduled reporting must be tested. Three routers have to be deployed in the test network. These devices represent one DA router and two BG routers (devices in ISP-A in Figure 13-17). The DA router is connected to both BG routers. All mentioned routers are connected to the NetFlow Traffic Collector (subcomponent of S-box-A). Routers are time synchronized by NTP.
Each BG router is connected to a packet generator. These BG routers run NetFlow monitors which observe inbound traffic coming via link attached to a traffic generator. On the DA router a NetFlow monitor inspects inbound traffic coming from BG routers via tunnels. The NetFlow Traffic Collector should pull traffic reports form all routers in the same moment. The data should be requested every 30 seconds. There should be measured difference between time deliveries of reports from different routers. Each report should be verified if it represents the same measuring period.

The Economic Analyzer should be verified if it correctly accumulates traffic volume from each measurement time slot represented by $\overrightarrow{X}^{\text{in}}(t_i)$ vectors. The accumulation procedure is done for the accounting period. The performed test should verify if the received reports containing traffic volumes from links or tunnels are properly accumulated in some implemented variables expressing vector $\overrightarrow{X}$. Separately, the algorithm for the inbound traffic reference vector calculation has to be verified. The synchronization verification will be checked by comparison of time when reports from different NetFlow monitors (running on different nodes) have been exported. The detail testing procedure will established during implementation cycle by preparing some reference cost functions.

The Fixed/Mobile Traffic Manager should be tested for proper communication with other partner Fixed/Mobile Traffic Managers and data delivery (reference and compensation vectors). The correctness of the compensation vector calculation should be verified, by simple comparison of analytical calculation with the result produced by the Fixed/Mobile Traffic Manager.

The Traffic Distributor component is an application deployed in a SDN controller. The main topic for testing this component is verification of the flow distribution algorithm. In the simple testing environment there should be used two DA routers connected via two GRE (General Routing Encapsulation) tunnels. One router should be connected to a packet generator. At the opposite tunnel end, a packet sniffer should observe flows traversing different tunnels.

The router connected directly to the packet generator is controlled by an SDN controller, which is provided with a different compensation vector and reference vector. It should be verified if flows are distributed amongst tunnels according to the compensation and reference vectors.

When all subcomponents are verified we can start second level of testing of all components together.
In Figure 13-17, we present the test environment. The data centers are represented by single computers: G-center and R-center. The G-center computer generates traffic which is received by the R-center computer. There are two other computers G1 and G2 generating background traffic. The G-1 sends traffic via link l1 and this traffic is received by R1 computer. We use a separate pair of computers G2 and R-2 for traffic generation and reception on link l2. There should be prepared a traffic pattern for each generator for the accounting period.

During first tests, an accounting period can last a few hours. We run a few simulations for a single accounting period without the DTM mechanism. We get the $\vec{X}^{\text{in}}(T)$ (current inbound traffic vector) and the $Z_{DA}^{\text{in}}(T)$ (the current inbound DC traffic vector via selected attached tunnels) at the end of the accounting period. We prepare a shape of cost functions (offline, preparing plots similar to that with iso-cost lines in Figure 5-4) in such a form that interesting traffic compensation can be done (our algorithm for the inbound traffic reference vector calculation should use a few areas for simplex optimization). Next, we run simulation with the same starting parameters when the DTM mechanism is switched on for a few consecutive accounting periods.

We observe the total cost for traffic transfer. We can use different traffic patterns for the G-1 and G-2 offering the same total traffic volume in an accounting period. A fixed G-center traffic pattern may be used for testing different traffic patterns generated by a DC. One can observe changes in monetary gain in relation to an increase of background traffic. We can inspect how rapid changes of background traffic can be mitigated by the DTM mechanism. The stability of the algorithm should also be tested that may become more challenging with faster response to changes.
13.3 Detailed specification of ICC

We now present the specification of the mechanism for Inter-Cloud Communication (ICC). ICC is indifferent to underlying routing and can be applied to current Internet over heterogeneous network technologies, e.g. IP, MPLS, ATM. In particular, we fully specify the mechanism’s components and respective algorithms, explaining its operation under both the federation and non-federated clouds/datacenters cases.

13.3.1 Terminology – ICC layers and modes of operation

In order to facilitate the presentation and without loss of generality, we first provide a basic scenario: the operation of the ICC mechanism among independent, i.e. non-federated clouds/datacenters. This is the simplest case for the ICC mechanism with respect to the number of distinct layers (and thus components) that need to be in place in order for the mechanism to work. We then fully present and comment on the incremental modifications under the federation scenario, i.e. when the set of interacting datacenters all belong to the same federation. A generic design has been opted, which allows for the support of multiple alternative policies under both the federated and non-federated scenario with minor changes in the way the cloud layer components interact in the two cases. It is worth noting that there are marginal incremental differences of the latter scenario as opposed to the former, thus simplifying the mechanism’s generic implementation, simulation and deployment. These in particular fall in the scope of Cloud Scheduler (CloS), i.e. the cloud service deciding where to send the data, and Cloud Information Service (CloI), i.e. the third-party running CloS for all federated DCs in case of federation, specification and sequence of interactions.

Thus, we begin the specification of the mechanism by initially focusing on the non-federated scenario. We stress to the reader that our mechanism deals with the management of traffic that has been marked as delay-tolerant, i.e. “time-shiftable”, e.g. inter-DC bulk data transfers. The rest of the traffic is referred to in this spec as “real-time” and is not subject to ICC mechanism control, thus it is treated a la Best Effort. There are unique features of ICC that differentiate it from related work, namely: a) The mechanism operates in very small time scales, less than the 5-min interval where most other mechanisms operate (e.g. NetStitcher), introducing the notion of slice, a portion of the 5-min interval; This allows a finer granularity in decision making and control over the rate of the traffic that is to be sent, further empowering the ISP to attain significant savings even in cases where his expectations regarding traffic may be wrong. b) The mechanism has built-in support for communication among ISP and Cloud layers, thus communicating each layer’s preferences to the other c) Though ICC is primarily a network layer mechanism, it fully respects the cloud layer business models and decisions (e.g, selection of destination cloud for bulk data transfers) and is built around the design-for-tussle principle, allowing the major stakeholders to conduct their business as usual without promoting any stakeholder in expense of others, while revealing more inter-layer information that could help stakeholders optimize their decisions leading to win-win outcomes for all (cloud and network) stakeholders involved. d) The mechanism is applicable even in cases of single-homed ISPs, i.e. when there is only one outgoing transit link, as opposed to other mechanisms (e.g. DTM). e) ICC has built-in support for both federated and non-federated DCs/clouds. The sequence of events and message passing that is required in order for the ICC mechanism to work is depicted inFigure 13-18.
Figure 13-18: The operation of the ICC mechanism when datacenters are non-federated.

Figure 13-18 depicts the sequence of actions that takes place in a bulk data transfer among non-federated datacenters. The mechanism is triggered by the request of the source datacenter to its respective home Internet Service Provider (its SmaS instance, i.e. the SmartenIT Information Service that is responsible for computing the network cost over the network path between a source and a given set of destinations, \( ISP_{src} \)) to deliver some traffic to one of potential destinations that comprise the set of candidate datacenter destinations \( <DCd> \). The initial request contains a description of the traffic to be sent as well as the specification of the value of a penalty for the per-unit delay (i.e. per 5-min interval \( d \)); this parameter penalizes the delay of content delivery and compensates the source datacenter for the deterioration of delay when the forwarding of the traffic is delayed by its home ISP. The priority flag allows the source datacenter to mark the respective traffic as time-shiftable, thus turning the operation of the ICC mechanism on or off, i.e. allowing or preventing any delay on the forwarding of the respective bulk data transfers. In practice, this marking can be done in a mutually acceptable way for the source DC and its home ISP: A straightforward way to do that is to use different Point of Interconnect, which are the standard points where traffic from an administrative entity/network is passed to a neighboring network. For instance, different Pols could be different ASBR-ports for the portion of traffic that is subject to the scheduling performed by the ICC mechanism. An alternative way would be to set the DiffServ codepoint to a certain value and use the DiffServ codepoints to differentiate between time-shiftable and real-time traffic over the same Pol, in order to mark the time-shiftable traffic.

**Specification of network cost metric \(<Pd(p)>\)**
The SmaS instance of the home ISP, namely \( \text{ISP}_{src} \), can both periodically and also on demand if no fresh information is available communicate with each of the SmaS of the destination datacenters \( <DC_d> \) to request the network distance metric.

We specify that the retrieval of the network cost metric, namely \( <P_d(p)> \), is performed on demand. This simplifies the implementation since no additional inter-SmaS protocol and caching mechanism (with a TTL for the \( <P_d(p)> \) retrieved in the past) must be implemented; instead a straightforward request is specified to be sent (if needed, see Discussion). Despite the really small size of the \( <P_d(p)> \) requests and responses (also compared to the size of respective bulk transfers), however for mature markets and high frequency of requests for inter-DC transfers within a federation it may be beneficial to indeed support such a protocol in the future: this can be added when needed and when it is justifiable to do so thus its exclusion from the present specification does not prevent the addition of such functionality in the future.

We specify the \( P_d(p) \) metric to be the BGP hop count, which is available to the \( \text{ISP}_{src} \) via BGP (other metrics for lower than Layer-3 protocols can also be used if needed). This selection is motivated by:a) the fact that this metric by definition prioritizes peering to transit routes, thus reflecting the actual routing preferences of the home ISP, b) it is a stable long-lived metric that will not result to path selection oscillations under the varying network conditions, and c) it is a gross yet indicative metric of the delay. The home ISP in practice can also combine this metric with delay estimation so as to generate richer and more accurate weighted ordering of the potential destinations: this is beyond the current state of the specification. It is also worth noting that the ICC mechanism is indifferent to the way the distance is computed and retrieved, i.e. whether there is pure IP routing in place where BGP provides the hop count or alternatively if some other technology used for the routing of the traffic (e.g. a label-switching technology such as MPLS).

**Cloud-network layer interaction**

Once the network metrics set \( <P_d(p)> \) is retrieved, the SmaS of the home ISP makes this information available also to the cloud layer of the source, i.e. the \( \text{CloS} \) of the source datacenter \( \text{DC}_{src} \). This is an important design decision ensuring that the decision on the destination datacenter is taken at the cloud layer (by the source DC at the non-federated use case or by the federation \( \text{CloS} \) at the federation use case) and not at the network layer. We further motivate and elaborate on this decision later in the remainder of this specification, immediately after presenting the incremental changes needed for the ICC mechanism to support the federated DCs as well.

**Specification of cloud cost metric \( L_d \) and destination DC selection process**

Similarly to the way the network costs are retrieved, \( \text{DC}_{src} \) queries all the candidate destination datacenters \( <DC_d> \) for the current ask, i.e. \( L_d = \text{ask(d)} \), that is the compensation (e.g. money) for receiving this traffic; especially under the federation scenario this can also be a metric that is selected from the federation business policy rules as the key metric for the cloud layer decision making process: a normal selection would be the load weight \( L_d \). This metric is computed as a weighted sum of the current percentage of load in the datacenter and the respective expected energy costs, thus reflecting the cost of handling the respective traffic at each of the \( <DC_d> \). For simplicity reason at the first step of the specification, we consider this metric to be an integer in the scale if 1 to 100 reflecting the relative handling of the respective traffic and load in case it is manageable; in case a \( DC_d \) cannot accept this request, the value -1 returns to clearly mark the DC
unavailability to handle this request. We now specify how the optimal target DC, namely $DC^*$ is computed:

$$DC^* = \arg\min \left[ a \cdot \text{load}(DCd) + b \cdot \text{ec}(DCd) \right], a + b = 1$$

The exact values of the normalized weights $a$ and $b$ depend on the relative weight of the load and energy cost in the decision making process of selecting the most appropriate destination DC. If energy efficiency is not considered, then $a = 1$ and $b = 0$. If both are considered to be of equal value, then $a = b = 0.5$

**Modifications for Federation**

The aforementioned procedure is slightly modified in the federated case where the destination DC - if such a selection is possible under the use case, i.e. the vector $<DCd>$ contains more than one element, where the mechanism is applied and not fixed – is again selected by the cloud layer but from the federation CloS. Note that at the federated scenario, the CloS instances per DC are replaced from CloI instances, while there is one instance of CloS, that of the federation. In practice, and in order to ease the implementation both the CloS and CloI functionality could be merged under a single component that can work in federated and non-federated mode. Note that the CloI business logic is limited to reporting the load and energy cost values to CloS which then applies the aforementioned argmin formula to select the optimal DC destination $DCd^*$. In particular, the cloud communication layer is modified compared to the previous sequence diagram as depicted in Figure 13-19.

![Figure 13-19: The ICC mechanism cloud layer when datacenters are federated.](image)

Under both the federated and non-federated DC scenario, the optimal destination $DCd^*$ is decided by the cloud layer and cannot be modified by the network layer, which takes this decision as a given. This decision was made in order to be compliant with the business models of cloud services. The business entities that make decisions on how the cloud infrastructure is managed are solely the cloud entities (with or without a federation), thus making it possible to control the cloud infrastructure and the load that is sent to different DC instances. Note that if instead the home ISP was making such a decision regarding the selection of the $d^*$, it would be impossible for e.g. the cloud federation to decide on how to load balance the tasks that need to be executed or for a source DC to send data to a
remote DC for performance reasons (caching of video): this would result in granting too much power to the network layer business entities (i.e. the ISPs) and loss of control of the DCs over their own business, which is clearly not acceptable.

Note that under the federated scenario the SmS interactions remain the same as before. This ensures that no matter what the selection of the network cost metric is (for possible variations of the proposed algorithm that are to be specified and evaluated), this metric can be properly measured at the path of the traffic. An extra step would be for the ISPsrc to report the vector <P_d(p)> to the federation CloS so that the latter has the required information to compute the optimal destination d*.

13.3.2 Pseudocode – ICC Scheduling Time-shiftable Traffic (STraS)

Once the destination DC d* is decided, the traffic is passed to the home ISP that will manage it. There are two main alternatives regarding the way the traffic is handled by the ISP: One is to use buffering and store and forward technique (a la NetStitcher), the other is to apply TCP rate control. ICC is indifferent to these two implementation options.

The main idea of the ICC Scheduling Time-shiftable Traffic (STraS) is to delay traffic so that the ISP won’t violate a target threshold which will determine the ISP charge according to the 95th percentile rule. We assume that each ISP is aware of its transit traffic patterns, i.e. stores in a database the history of the upstream and downstream link utilization, as well as the respective 95th percentile charge. These historical data can then be used by the ISPsrc SmS in order to shape the traffic so that cost savings can be obtained. On the other hand, the shaping of the time-shiftable traffic will result in monetary compensations that need to be paid by the ISP to his respective customers who experienced the additional compared to Best Effort transmission – delays. The ICC mechanism is more meaningful to apply if the per unit-time delay penalties per Mbyte to be incurred by the ISP are expected to be considerably lower than the corresponding extra transit costs. To this end, we implicitly assume in the sequel that the ISP tries to lower the transit costs, by setting a low yet realistically feasible target value for the 95th percentile, without though trying to attain the lowest such possible value (so as to avoid excessive delays).

Home ISP applies the following algorithm within y equally-sized epochs of every 5-min interval within a billing cycle, i.e. within the period within which the transit links are metered and charged: Initially the target value of the transit link 95th percentile C_target is set, given the past history of traffic patterns and the Cumulative Aggregate Growth Rate (CAGR) estimated from statistics. Throughout each billing period where the 95th percentile is computed and the ISP is charged, i.e. for each month, the algorithm decides when (i.e. in which epochs – see below) to send the time-shiftable traffic and at which rate per epoch; the real-time traffic is forwarded as usually in the Best Effort Internet. On the contrary, the time-shiftable traffic may experience further delays. In particular, each of the 5-min time intervals is sliced to y epochs, e.g. for y=2 each 5-min interval is sliced to two 2.5min epochs. The algorithm uses these epochs in order to control the amount of time-shiftable traffic to be transmitted (taking into account the definition and calculation of proper thresholds) so that this traffic combined with the real-time traffic (which is not known a priori) does not exceed the C_target rate over the 5-min interval. The current specification also takes advantage of the fact that the C_target can be violated at 5% of the 5-min intervals of the billing period without any cost by sending with maximum rate (thus constrained only by the transit link capacity) at the last 5-min periods so that more data can be sent without additional cost for the ISP. Additional variations of the ICC may prescribe different ways of choosing the 5% 5-min intervals where the transit link will be
Utilized with max rate; this is beyond the scope of the current specification which for simplicity reason specifies just one concrete design of the ICC mechanism.

Get Ctarget;  // Target rate for the 95th percentile of the billing period (month)
Set y = 2;  // Indicative value for splitting the 5-min interval to epochs
Set cagr = 0.25;  // Traffic increase year by year
// Check if I am at the end of billing period or not
If (this_interval_in(last 5% of 5-min intervals)) {
Set n = number_of_intervals_where_Ctarget_violated;
If (this_interval_index <= last_interval_index – n)Jump to label send_with_max_Rate;
}
// Following code executed for at least 95% of 5-min intervals:
Get ns_rate_historical[1…y];  // historical rates of same 5- interval and epoch of past year
// Set epoch thresholds to cap transmission. Start conservative
Define epoch_tholds[1…(y - 1)];  // array of (y - 1) positions
For j=1…(y - 1) {
Set epoch_tholds[j] = 0.9;  // can be fine-tuned with trial and error
}
// thold of last position will be computed dynamically … fill the link as much as possible
Foreach (j=1 … y - 1) epoch:  // First (y-1) epochs of 5/y min duration
{
Get ns_rate_previous;  // rate of non-shiftable traffic one year back
Set ns_estimated_rate = ((1+cagr)*ns_rate_historical + ns_rate_previous) / 2;
If (ns_estimated_rate < Ctarget * epoch_tholds[j]){
Transfer(epoch_tholds[j]*Ctarget–estimated_rate) shiftable traffic;
}
} // end foreach
// At end of each epoch get the metered rate of non-shiftable traffic-update the information
Foreach (j=1 … y) epoch {
Set ns_rate_previous = ns_rate_metered;  // actual metered rate
}
If (this_epoch == y) {  // different logic for last epoch
Get ns_rate_previous;  // rate of non-shiftable traffic
Set ns_estimated_rate = ((1+cagr)*ns_rate_historical + ns_rate_previous) / 2;Set thold = 0.95 * Ctarget – rate_of_previous_ (y-1)epochs;
If (ns_estimated_rate < Ctarget * thold )
The algorithm attempts to shape the transit link traffic so as not to exceed the target rate $C_{\text{target}}$. Note that as stated earlier, the selection of $C_{\text{target}}$ itself given the traffic matrix also implies a resulting delay penalties cost, depending on the amount of the time-shiftable traffic that is to be further delayed per 5-min interval so as to reduce the transit link throughput. We now specify what the feasible values of $C_{\text{target}}$ are and how this value can be estimated and set by the ISP. Since traffic constantly increases in communication networks, by storing the historical rates (i.e. rates metered one year back) and computing the CAGR over the transit link, the ISP can predict a rate of \( \text{cagr} \times C_{\text{target}}(\text{year}-1) \). This rate is expected to suffice in order to deliver the customers’ traffic over the transit link, assuming traffic patterns remain the same year by year. However, traffic patterns exhibit fluctuations, this the attainable realistic value to be expected even under traffic shaping of time-shiftable traffic will be less than the theoretical bound of \( \text{cagr} \times C_{\text{target}}(\text{year}-1) \), which could be attained under perfectly constant bit rate traffic pattern. Furthermore, it is also straightforward to argue that the larger the portion of the customers that will subscribe portions of their traffic to be handled under the ICC mechanism becomes, then the more efficient the shaping of the traffic and the larger the reduction of the target $C_{\text{target}}$ will be. At any case, $C_{\text{target}}$ can never be less than the 95th percentile that would occur if only real-time traffic was crossing the link; this comprises the lower feasible value of $C_{\text{target}}$ (which however is expected to result in high delay penalties). On the other hand, the upper bound of the amount of time-shiftable traffic to be sent over the link where the ICC mechanism is applied will be \( w_{\text{goal}} \times \text{cagr} \times [C_{\text{target}}(\text{year}-1) - \text{Sum(avg_shiftable_rate)}] \), where \( w_{\text{goal}} < 1 \) and \( \text{Sum(avg_shiftable_rate)} \) denotes the sum of average rates of the time shiftable traffic of all ISP customers who have agreed to allow portions of their traffic to be handled by the ICC mechanism. Clearly, the closer the value of \( w_{\text{goal}} \) is set to 1, the more the respective time-shiftable traffic is transmitted (especially at peak hours) and thus the lower the cumulative delay penalties to be paid by the ISP to the customers experiencing worse than Best Effort performance for the time-shiftable traffic will be. In this spec we refrain from providing an optimal way of computing $C_{\text{target}}$ given the link characteristics, the knowledge of the traffic patterns and the values of 95th percentile charge and delay penalty $d$. Instead, we opt for a trial-and-error external loop.

### 13.3.3 Information – DB schema

We now provide the required information that must be kept as database tables:

<table>
<thead>
<tr>
<th>Table: Transit_rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
</tbody>
</table>

© Copyright 2014, the Members of the SmartenIT Consortium
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_id</td>
<td>INTEGER</td>
<td>The id of the transit link</td>
</tr>
<tr>
<td>when</td>
<td>DATE</td>
<td>The billing period</td>
</tr>
<tr>
<td>rate</td>
<td>INTEGER</td>
<td>The rate metered</td>
</tr>
</tbody>
</table>

### Table: Metered_rates

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_id</td>
<td>INTEGER</td>
<td>The id of the transit link</td>
</tr>
<tr>
<td>interval</td>
<td>DATE</td>
<td>The 5-min interval</td>
</tr>
<tr>
<td>epoch</td>
<td>INTEGER</td>
<td>The y index</td>
</tr>
<tr>
<td>ns_rate</td>
<td>INTEGER</td>
<td>Rate of non-time shiftable traffic metered</td>
</tr>
<tr>
<td>s_rate</td>
<td>INTEGER</td>
<td>Rate of time shiftable traffic within epoch</td>
</tr>
</tbody>
</table>

### Table: Metered_rates_per_customer

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer_id</td>
<td>STRING</td>
<td>The customer for which the rates are stored</td>
</tr>
<tr>
<td>link_id</td>
<td>INTEGER</td>
<td>The id of the transit link</td>
</tr>
<tr>
<td>Interval</td>
<td>DATE</td>
<td>The 5-min interval</td>
</tr>
<tr>
<td>Epoch</td>
<td>INTEGER</td>
<td>The y index</td>
</tr>
<tr>
<td>ns_rate</td>
<td>INTEGER</td>
<td>Rate of non-time shiftable traffic</td>
</tr>
<tr>
<td>s_rate</td>
<td>INTEGER</td>
<td>Rate of time shiftable traffic within epoch</td>
</tr>
</tbody>
</table>

### 13.3.4 API and Interfaces

We now provide the prototype of the functions that comprise the API per ICC component.

**CloS/Clol:**

```java
cloSICCinit(Vector<DC> candDCs, int volumeOfData, int shiftableFlag, int d);
```

This function triggers the mechanism for the data transfer. It returns a unique transfer id.

```java
cloSrequestLd(int volumeOfData, DC candDC);
```

This function allows the CloS to request the Ld of the candidate DCs.

```java
cloSjoinFederation(int fedId, DC DCid);
```

This function is needed to support federated DCs. In particular, this allows a DC to join a federation of DCs. (This is a simplification of the respective real-world process, useful mostly for simulation purposes)

```java
cloSleaveFederation(int fedId, DC DCid);
```

This function is invoked from a DC so as to leave a federation of DCs. (This is a simplification of the respective real-world process, useful mostly for simulation purposes)

```java
cloSoptDest(Hashmap<DC, Integer>Ld, Hashmap<DC, Integer>Pd, int transferID);
```

Returns the id of the DC which is considered to be the best choice for a bulk data transfer.

```java
cloShandleTraffic(int id, DC destination, Byte[] data);
```
Passes the actual data from the PoI where the data will be delivered.

**SmaS:**

```java
boolean passNetCosts(HashMap<DC, Integer>Pd, int CloSID);
```

This function is invoked to pass the `<Pd(p)>` vector to the CloS/CloI.

```java
void ICCack(int trafficID, DC DCid);
```

Acknowledges the handling of the bulk data transfer from the mechanism to DCsrc.

### 13.3.5 Discussion

The ICC algorithm attempts to dynamically adapt to the varying network conditions and maximize the utilization link without violating the $C_{target}$ threshold. The splitting of the 5-min interval to $y$ epochs allows a finer control of the rate thus allowing for a better chance not to violate the $C_{target}$ threshold, as opposed to the case where such a partitioning is not carried out (as assumed by most of the related work in the literature). To this end, the setting of initially conservative thresholds `epoch_tholds[]` also assists in a finer discipline of the sending rate. These thresholds will also determine the percentage of 5-min intervals within which the $C_{target}$ threshold will be violated due to unexpected rise in the $ns\_rate\_metered$ of all the $y$ epochs within a 5-min interval. We have already specified that this algorithm is to be run at the SmaS of each sending ISP. Note that the ISP will inevitably multiplex multiple bulk transfers, thus it is best to prioritize the most urgent transfers if such different priorities exist. A straightforward way to do that is to use different Point of Interconnect (i.e. different ASBR-ports) or DiffServ codepoints (or MPLS labels) for the different portions of time-shiftable traffic. The clear separation of the cloud and the network layer is an intended design choice that ensures that ICC is compliant with current business models and does not provide unwanted power to some stakeholder.

The current specification prescribes that the metric of network cost used is defined as the number of BGP hops between the source and the destination DC. This comprises a really attractive feature of our mechanism since IP networks typically exchange only BGP information and data, and also are very reluctant to disclose sensitive information such as detailed topology and delay metrics. However, if such information is indeed made available, the current generic specification of the mechanism can take full advantage of it since it can acquire it via the inter-SmaS communication messages. The specified mechanism also relies on having a DB within the ISP domain holding information such as the metered and historical rates of the transit links. We argue that this constraint is not significant since networks already rely on local database systems in order to perform various operations: the Traffic Engineering Database is the most prominent example.

Concluding, the mechanism is tailored for the needs of the Operator Focused Scenario. It is compliant with current business models and TE best practices, thus making it attractive for practical deployment. The current specification comprises one of multiple variations being evaluated also by means of simulations.
13.4 Detailed specification of MRA

The following pseudo code is to be executed by every central node. The return value of function greediness() is also referred to as greediness vector.

Figure 13-20: SmartenIT System Architecture as shown in [9].

Figure 13-21: Sequence Diagramm of the MRA Mechanism.
13.4.1 Terminology

**VM**

A virtual machine that belongs to a customer and executes his workloads. Customers may have more than one VM at their disposal. A VM has Virtual Resources (VR), such as VCPU and VRAM.

**PM**

A physical machine that is part of a cloud. PMs have a certain amount of Physical Resources (PR) and host VMs. How many VMs can be hosted by a PM depends on the PM’s PRs and how active the hosted VMs are. In particular, how many VMs a PM can host depends less on the VMs’ VRs (because VRs are only an upper bound for actual (physical) resource consumption of VMs) but rather on how much of their VRs the VMs actually deploy.

**Hypervisor**

The operation system of a PM is called hypervisor. Since the PM hosts VMs, the hypervisor is in literature also referred to as host operating system.

**Orchestration Layer**

The orchestration layer is the software that coordinates a cloud. For example, it determines which VM is started next and on which PM.

**Live Migration**

Is the process of moving a VM between different PMs without interrupting its operation. VMs may also be live migrated between PMs of different clouds. During the time of the migration the performance of the VM may suffer, as all RAM contents and other time critical data has to be transferred between PMs.

**Quota**

A customer’s quota defines the maximum of VRs that his VMs may have in total. Quotas may differ among customers. Especially for private clouds, where VMs are not payed for individually but the cloud is a public property, quotas are a sound approach to establish high-level resource allocations.

13.4.2 Utilization Information

The subsequent table outlines the resources to be monitored and taken into account when applying the MRA mechanism.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory</td>
<td>Discrete</td>
<td>MB RAM utilized by/allocated to the VM, when measured.</td>
</tr>
<tr>
<td>cpu_util</td>
<td>discrete</td>
<td>Average relative CPU utilization.</td>
</tr>
<tr>
<td>vcpus</td>
<td>discrete</td>
<td>Number of VCPUs.</td>
</tr>
<tr>
<td>disk.access</td>
<td>discrete or</td>
<td>May be measured either in number of requests or byte, represented as total</td>
</tr>
<tr>
<td></td>
<td>cumulated</td>
<td>or average per second, and separated for read and write.</td>
</tr>
<tr>
<td>disk.root</td>
<td>discrete</td>
<td>MB root storage utilized by the VM, when measured.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Root disk stores the VM’s base image.</td>
</tr>
<tr>
<td>disk ephemeral</td>
<td>discrete</td>
<td>MByte ephemeral disk utilized by the VM, when measured.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ephemeral disk stores, contrary to the root disk, data only for the lifetime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of an instance, i.e., it is</td>
</tr>
<tr>
<td>network</td>
<td>discrete or cumulated</td>
<td>non-persistent storage, existing as long as its VM.</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------</td>
<td>---------------------------------------------------</td>
</tr>
</tbody>
</table>

May be measured either in number of packets or byte, represented as total or average per second, and separated for in- and outgoing. If different network interfaces exist for a PM, this also has to be taken into account.

### 13.4.3 Pseudo code for the consumption collector

Subsequently the pseudocode for the central node is given. It mainly consists of the greediness calculation.

```plaintext
greediness()
    greed_cust[number of customer]

    for each PM supervised by the central node

        PRs[n] = get_PRs()
       designparameter // this parameter is needed for normalization, it determines the magnitude of greediness that is returned

        norm[n] // array needed to normalize resource units when greediness is calculated
        for every entry i in norm[] do

            norm[i] := designparameter/PRs[i]

        util[m][n] : array with util[i][j] = utilization of VM i of resource j, to be generated by ceilometer, i.e.,

        calls get_consumptions(PM)

        endw[m][n] : array with endw[i][j] = endowment of VM i to resource j

        dev[m][n] := array with dev[i][j] = util[i][j] - endw[i][j] // this array contains aij-bij of Section X

        pos[m][n] : array with pos[i][j] = max(dev[i][j],0) // needed to calculate alpha

        neg[m][n] : array with neg[i][j] = min(dev[i][j],0) // needed to calculate beta

        alpha[n] : array with alpha[i] = sum of column i in array pos

        beta[n] : array with beta[i] = sum of column i in array neg

        credit[n] : array with ratio[i] = max(-1, alpha[i]/beta[i]) // this array holds the credit factors for coding resources

        // Multiply aij-bij with the respective factors before adding up

        pos2[m][n] := array with pos2[i][j] = norm[i]*pos[i][j]

        neg2[m][n] := array with neg2[i][j] = -1*norm[i]*neg[i][j]

        // add up pos2 and neg2 to arrive at greediness of each VM

        greediness[m] : array with greediness[i] = sum of column i of pos2 + sum of column i of neg2

        for every entry i in greediness[] do

            get customer j of i and add greediness[i] to greed_cust[i]

        return greed_cust

send greediness vector to other central nodes
receive greediness vector from other central nodes
add received greediness vectors to calculated greediness vector
return result to supervised PMs
```

### 13.4.4 Internal methods

- **get_consumptions(PS):**
  - Receive the consumption profiles of all VMs on PM via Ceilometer
  - Best values for the units and measurement interval of resources have already been determined in the framework of Ceilometer.
  - For each VM a field as in Section 13.4.2 shall be returned

- **get_PRs()**
  - Receive the PRs of all PMs in the cloud

- **greediness()**
  - Return the greediness of each customer as described in Section 5.3.1
o Calls get_consumptions(), get_VRs(), and get_PRs().

- get_allocation(resource)
  o Return target allocation of an overloaded resource for all VMs running on a PM
  o Called by a PM/hypervisor, when a resource gets overloaded

- reallocation(resource, VM, amount)
  o For resource add/remove remove amount from VM
  o Amount may be positive or negative
  o This function’s implementation will strongly depend on resource. This function may only be called, when get_allocation()’s deviation from the actual allocation crosses a certain threshold. This threshold will also depend on which type of resource is reallocated (especially for RAM a reallocation is costly, wherefore a reallocation should only be triggered when the current allocation is highly unfair, where unfairness can be deduced by mapping the current greediness vector to a scalar with a suited single resource fairness metric, i.e., jain’s index). Note that a threshold for the deviation of two vectors has to be determined, which also depends on the considered resource, wherefore the determination of such threshold is complicated and part of the research.
13.5 Detailed specification of RB-HORST

The RB-HORST mechanism eases mobile data offloading to WiFi by sharing home-based Access Points among trusted friends. Moreover, it allows the placement of content near to the end user such that users can access it with less delay and higher speed, which generally results in a higher quality of experience.

To achieve the aforementioned goals, RB-HORST performs a number of operations. First of all, it enables the ‘federation’ of home-gateways/access points through the use of social networks. Each owner of a home-gateway (called uNaDa), registers its uNaDa to the social network, by providing its GPS coordinates and the credentials required accessing its private network/SSID. This information becomes available to the friends (in terms of social networking) of the user. In the same manner, through the social network, a user can learn about the existence of ‘friendly’ uNaDas.

Once the uNaDAs are registered, the first goal of RB-HORST can be achieved. When a user carrying a mobile device enters the coverage of a ‘friendly’ uNaDa, he is informed through the social network of the opportunity to switch from the mobile data network to the WiFi, using the credentials provided, by asking the WiFi owner’s permission (through the social network).

The second goal of RB-HROST involves the prefetching of content to remote uNaDAs that have high probability to be consumed by their attached (visiting) users. To achieve this, the uNaDa monitors the social network of its owner for interesting content. Once a content, that is considered interesting (according to some popularity and location criteria) is published on the social network, the uNaDa, tries to prefetch it (either from the original source or from a trusted uNaDa) before the user actually asks for it.

The WP2 specification is mapped to the WP3 architecture, and the following diagram visualizes the components that will be active in the instantiation of RB-HORST by the SmartenIT prototype. Note that the role of the uNaDa as a proxy/cache and access point is not displayed here. It is assumed that the uNaDa device acts as an access point with two SSIDs, acts as a DHCP/NAT server and also hosts a proxy/cache server, with the latter being triggered by the Cloud Traffic Manager.

![Diagram of components involved in RBH](image-url)

Figure13-22: SMT components (in orange) involved in RBH.
13.5.1 Functionalities of RBH

Performing RBH involves the following functions and steps, detailed in the WP2 specification document.

- Create and maintain an overlay network
  - Form DHT
  - Exchange messages between uNaDas (unicast or broadcast)
  - Exchange files between uNaDas
- Track content
  - uNaDas advertise cached content
  - other uNaDas request content location(s)
- Content popularity prediction
  - Overlay-based prediction
    - Determine AS distance between two UNaDas
  - Social prediction
    - Get user’s social graph
    - Get friends’ (Vimeo-related) posts
- Caching/Prefetching
  - Intercept and cache Vimeo Requests/Traffic
  - Extract content from and inject content to remote uNaDas’ cache
  - Cached/prefetched content management
- Mobile data offloading
  - Facebook authentication
  - Register uNaDas (coordinates and credentials)
  - Access to RB-HORST Facebook app
  - Retrieve credentials for near-by ‘friendly’ uNaDas

13.5.2 SMT components involved in RBH

RB-Horst involves three major components of the architecture. The Cloud Traffic Monitor provides the main functionality of the RB-HORST mechanism and provides the main intelligence. Moreover, the Cloud Traffic Monitoring component is the main point of integration for the code. The second heavily involved component is the Overlay Manager, which manages access to the Distributed Hashtable running below the mechanism. Third, the Proximity Monitor is the main component for querying social data. The component queries social data once in a while and caches the results for access by other components. All other components and their involvement in the mechanism are listed in Table 13-1.

The table consists of three columns: the Component name as defined by the SmartenIT architecture, the Entity, i.e., the physical machine on which the component will be running and the functions of the RB-HORST mechanism defined by the WP2 specification. It is important to note, that from an implementation perspective there is no necessity to run an additional FacebookAPP server. Moreover, Table 13-1 lists for each component, the RBH functions to be integrated.

Table 13-1: Integration of the implemented functions of RB-HORST in the SmartenIT architecture components.

<table>
<thead>
<tr>
<th>Components</th>
<th>Entity</th>
<th>WP2 RBH Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology / Proximity Monitor</td>
<td>uNaDa</td>
<td>getASVector(Destination IP address),</td>
</tr>
<tr>
<td>Module</td>
<td>Functionality</td>
<td>uNaDa Python Functions</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>------------------------</td>
</tr>
<tr>
<td><strong>Overlay Manager</strong>&lt;br&gt;(all DHT/Overlay related functionality)</td>
<td></td>
<td>Join(UNaDa ID, IP Address), update(UNaDa ID, Address), advertiseContent(ContentID), getProviders(ContentID), announce(ContentID, BloomFilter), getPrediction(), sendMessage(UNaDa ID, message), sendBroadcastMessage(message), requestFile(UNaDa ID, Content content)</td>
</tr>
<tr>
<td><strong>Social Monitor</strong>&lt;br&gt;(Directly accesses Facebook and makes queries)</td>
<td>End-User-Device</td>
<td>getFriends(), getWallPosts(FacebookID), getFeed(FacebookID), getVideoInfo(ContentID)</td>
</tr>
<tr>
<td><strong>Social Analyzer</strong>&lt;br&gt;(Uses data gathered by social monitor to make prediction etc.)</td>
<td>End-User-Device</td>
<td>getPrediction(), getOwner(SSID), getTrust(userID A, userID B), addTrust(), removeTrust(), requestWiFi(SSID)</td>
</tr>
<tr>
<td><strong>Mobile Network Traffic Manager</strong>&lt;br&gt;(manages the WiFi connection to connect to RBHORST uNaDa)</td>
<td></td>
<td>Monitors WiFi SSIDs for RBHORST SSID, Scans for RBH WiFi, Connects to RBHORST SSID, Authenticates user to host uNaDa</td>
</tr>
<tr>
<td><strong>User Interface</strong></td>
<td>End-User-Device</td>
<td>Configuration of the mobile app is done by a Facebook login. Facebook User ID is stored.</td>
</tr>
<tr>
<td><strong>Cloud Traffic Manager</strong>&lt;br&gt;(contains the main logic of RBHORST and connects the components)</td>
<td>uNaDa</td>
<td>Start(), Login(UNaDaID,username,password), requestTrust(username), authenticate(username, password)</td>
</tr>
<tr>
<td><strong>User Interface</strong>&lt;br&gt;(direct interaction with user)</td>
<td>uNaDa</td>
<td>Facebook Authentication Configuration</td>
</tr>
<tr>
<td><strong>Proxy / Cache</strong></td>
<td>UNaDa</td>
<td>proxyRequest(request), cacheContent(content), updateAccessLog(), adjustCache(socialPrediction, overlayPrediction)</td>
</tr>
</tbody>
</table>
13.5.3 Component-based sequence diagram for each functionality

The following section presents component-based sequence diagrams for each functionality provided in Section 6.1.1.

13.5.3.1 Facebook authentication

The first step for the owner of the uNaDa is to login to his home gateway. It will be required for him to login using his Facebook account, thus accept the permissions of the SmartenIT Facebook application.

If it is the first login of the uNaDa owner, then his Facebook-related information, such as his access token, facebook profile ID, name, etc. will be stored to the uNaDa DB.

The user is then free to browse to the uNaDa UI and configure required parameters, such as uNaDa execution parameters, social prediction parameters, cache policies, etc.

13.5.3.2 Overlay network

The functionality of the overlay networks is described in the following.

13.5.3.2.1 Create

On bootstrap, the cloud traffic manager joins the overlay network. It retrieves the unada information (address, location) and also the stored contents from the DB component and puts them to the overlay network.
Support functions:

- `findUnadaConfiguration` (from UnadaConfiguration DB table): get uNaDa address, location, etc.
- `findAll` (from Content DB table): get all stored contents, to be advertised

13.5.3.2.2 Maintain

Every interval t (which should be configurable through the UI and DB components), the cloud traffic manager retrieves the currently stored contents and advertises them to the overlay manager, which will put them to the overlay network.
Figure 13-25: Maintenance of overlay network executed in a fixed time interval $t$.

Support functions:
- `findAll (from Content DB table)`: get all stored contents, to be advertised
13.5.3.3 **Track content**

![Diagram of Track content]

Figure 13-26: The tracking process is executed periodically for each content shared.

13.5.3.4 **Overlay based prediction**

The overlay based prediction is performed and initiated from cloud traffic manager every interval $t$. The overlay manager requests all the contents from neighboring uNaDas and runs the overlay prediction algorithm, returning the sorted list of contents.

![Diagram of Overlay based prediction]

Figure 13-27: The Overlay Based prediction is initiated by the CTM and executed in the OM.
13.5.3.5 Retrieve Facebook information

Every interval $t_1$ (also configurable from UI and DB components) – could be every midnight- the cloud traffic manager initiates the facebook data collection. The social monitor requests from the Facebook Graph API the friends, wall posts, feed, and video information of the owner with facebookID and then stores (or updates) them to the database.

The following figure presents the retrieval and insertion of owner’s friends. The same would apply for news feed, wall posts, and video information.

Support functions:
- **getFriends(facebookID)**: It is exposed by Social Monitor and queries all friends of a facebookID
- **getWallPosts(facebookID)**: It is exposed by Social Monitor and queries all wall posts of a facebookID
- **getFeed(facebookID)**: It is exposed by Social Monitor and queries all news feed of a facebookID
- **getVideoInfo(contentID)**: It is exposed by Social Monitor and queries all videos of a contentID
- **insertFriends(), insertWallPosts(), insertFeed, insertVideoInfo**: Insert and update these parameters to the respective tables.
- **insertTrustedUsers()**: Inserts the found friends into the TrustedUser table.
13.5.3.6 Social prediction

Similar approach with overlay prediction is also performed for the social prediction. Instead, the social analyzer will request all the statistics from the social monitor, and then runs the social prediction algorithm.

After both the overlay and social prediction are performed, the cache component would re-adjust the cache.

Support functions:
- `findSocialPredictionParameters` (from the SocialPredictionParameters DB table): retrieves all the required social prediction parameters

13.5.3.7 Determine AS distance between two UnaDas

Figure 13-29: Social prediction uses information stored in the Social Monitor.

Figure 13-30: To find the closest UNaDa a traceroute is requested from all providers.
13.5.3.8 Intercept Vimeo Requests/Traffic
This functionality sits in the Proxy/Cache component. All http requests are proxied through this component, and if they are Vimeo requests, then the proxy/cache component checks the db if they are already stored and are served from the uNaDa http server, otherwise the proxy/cache component will cache the video and updates the db.

Support functions:
- findContentByID (from the Content DB table): checks whether a content with contented is stored locally.
- insertContent (into the Content DB table): inserts a cached content to the db
- updateCache (into the Cache DB table): updates the related cache

13.5.3.9 Cache / Prefetch content Management
This functionality sits in the Proxy/Cache component. Since the functionality involves only one component and therefore a component based sequence diagram is left out.

Support functions:
- findAllContents (from the Content DB table): finds all cached contents to the db
- deleteContents (from the Content DB table): deletes cached contents to the db
- insertContents (into the Content DB table): inserts cached contents to the db
- updateCache (into the Cache DB table): updates the related cache

13.5.3.10 Mobile Application Facebook authentication
The mobile UI needs to show the notifications plus a configuration view which shows a Facebook login button. On first time use if the App the OAuthToken is used to retrieve the FB ID from Facebook which is then used to authenticate on a friendly UNaDa.

Figure 13-31: Before the app can be used the user has to be authenticated with Facebook and the user ID will be stored.
13.5.3.11  **Mobile data offloading**

After successful Facebook login, the end-user application will search for open RB-HORST SSIDs, and if it finds one, it will send a login message to the cloud traffic manager, including the user's Facebook ID. The cloud traffic manager of the uNaDa will check whether the user belongs to owner's trusted users, and if yes, it will respond with the credentials of the private SSID. Otherwise, the end-user application will send a trust request to the cloud traffic manager.

![Diagram](image)

Figure 13-32: The mobile App has the Facebook ID of the user stored and uses this to authenticate with the foreign UNaDa.
13.6 Detailed specification of SEConD

SEConD is a traffic management mechanism that exploits social information to enhance content distribution, by enabling socially-aware targeted prefetching, as well as caching based on OSN users demand pattern and peer-assisted video distribution. According to the literature [66], the video viewing in OSNs is driven both by social relationships and interest similarities with respect to the content. Based on these two parameters, a user influences some users more than others. SEConD takes advantage of this information and for each user (source user) assumes a categorization of his friends (at most two social hops) in three viewers’ categories with respect to the level the source user influences them. Therefore, SEConD for each user defines three categories of viewers: followers, non-followers, others. Note, that these categories do not include all the friends of the user but only those that he affects over a low threshold. The definition of these categories is an extension of the categories presented in [66]. In particular, for each uploader, we consider as: a) Followers: his 1-hop or 2-hops friends that watch over a high-threshold (80%) of the videos he uploads. b) Non-followers: his 1-hop or 2-hops friends that watch less than high-threshold but more than a low-threshold (30%) of the videos he uploads. c) Other viewers: his 1-hop or 2-hops friends that watch less than low-threshold but more than 20% of the videos he uploads. Note that if a friend of an uploader watches his videos, then this friend watches at least 20% of them. SEConD can decide a categorization of viewers based on OSN historical data in the early stage of the deployment and later by monitoring the users’ interactions. Finally, SEConD assumes that each video belongs to a specific interest category. This is realistic, since in YouTube one (or more) interest categories assigned in each video [69]. Similarly, each user has interest in specific video categories.

SEConD exploits the information gathered from OSNs in order to predict demand of a specific user for a specific content. This relies on the fact that some users socially connected to an uploader (mainly followers and non-followers), are usually watching the videos he uploads. Also, using the same information creates socially-aware messaging overlays per user, in each of which each user can send demand indications to his potential viewers (followers, some of non-followers) for content items of a particular topic, in order to proactively store the first piece (chunk). The demand indication messages enable targeted prefetching, which boosts users’ QoE by eliminating videos start up delay. Finally, SEConD introduces a Socially-aware Proxy Server (SPS) in each Autonomous System (AS). The SPS is responsible for the formation (also update) of messaging overlays, to operate as P2P tracker (BitTorrent like) for local content-based P2P overlays (chunk based video delivery) and to achieve high traffic localization, by caching content to assist in sharing, when the local P2P is not adequate.

13.6.1 Socially-aware messaging overlays (messaging clusters)

The socially-aware messaging overlays are created and used to disseminate alert messages from the uploader of a video, to his friends that are potential viewers of this video. Each uploader is considered to have interest in specific video categories, and thus maintains a messaging overlay for each video category of his interest. Each messaging overlay contains as potential viewers, all the followers of the uploader, and only the non-followers that are also interested in the respective video category. Consequently, each uploader has a number of messaging overlays, equal to the number of video categories that he is interested in. Figure 13-33 depicts an overlay constructed for an uploader (“source”) and for one of his interest categories. Later, we demonstrate (cf. subsection 13.6.5) how our prefetching algorithm is taking advantage of messaging overlays. Finally,
as the authors observed in [66], the 94% of the videos each user watches are at most from 4 video categories. Thus, for each uploader, we create overlays only for his top 4 categories.

13.6.2 Socially-aware Proxy Server (SPS)
The SPS is a socially-aware proxy located within an AS, in order to localize the traffic created by the activity of OSN users in this region. The SPS is the orchestrator of the mechanism and has several responsibilities and functionalities: 1) Monitors user interactions related to videos. 2) Forms/updates the messaging overlays, based on monitoring information. 3) Pushes video prefixes (first \( n^* \) chunk) to users, in order to server requests produced my messaging overlays. 4) Adds users requesting a video to the local content-specific P2P swarm (P2P Tracker). 5) Assists swarms with inadequate upload bandwidth, acting as super-peer. 6) Caches video prefixes and videos following requests.

13.6.3 Local Content-based P2P overlays
The Local Content-based P2P overlays are created by the SPS for each content item stored in its cache. The SPS of each AS also operates as a P2P tracker and in each swarm adds the users who have stored in their UD the specific content (seeders) and also the users who request to watch the video at the moment (leechers). Thus, when a user requests a video, the SPS checks if the requested video is in his cache and so a swarm for this content exists. If the swarm exists the SPS adds the user to the swarm, if does not exists he downloads the video and stored it in his cache and also creates a new swarm, in order to perform dissemination of this video within the same AS. Finally, If the per user upload bandwidth available by other sharers in the swarm exceeds the video bit rate, then the SPS stops assisting the sharing for this swarm. This happens, since the resources the SPS can offer are not unlimited.

13.6.4 Caching
Caching is very important to SEConD for QoE enhancement, and for achieving a high reduction both in inter-AS traffic created by video dissemination and in the contribution of the server hosting the video. In SEConD, we follow a two-level caching strategy with caching of video prefixes and videos taking place both in the SPS and in the UD. Caching prefixes in UD aims to decrease latency by eliminating the video start-up delay (or stall time), while caching the videos themselves in users’ caches aims to assist P2P video sharing. On the other hand, caching prefixes and videos in the SPS is done mainly for keeping traffic local.
13.6.5 Socially-aware pull based prefetching algorithm

The Socially-aware pull-based prefetching algorithm comprises the following steps, also depicted in Figure 13-34: 1) When an uploader (source node) uploads a video, he pushes an alert message to each of the users in the messaging overlay that corresponds to the interest category of this specific video. 2) After a user receives an alert message, he sends a request to his local SPS asking to receive the prefix of the video referred in the message. 3) When the local SPS receives a prefix request, if this is not already cached, then it downloads this prefix from the video server where the video is hosted. 4) The local SPS caches the prefix of the video and pushes it to the user who requested it. Finally, the user stores the prefix of the video in his UD.

![Figure 13-34: Steps of the SEConD's prefetching algorithm.](image)

13.6.6 Stored information in different components

**SPS**

SPS_id, usersID_list – list of users (user_id's) connected to current SPS.

interest_categories – table for mapping interest categories into category IDs. In each position of the table we store the “tag/keyword” of a category, while the id of this category is its position in this table.

friends_categorization_thresholds – SPS maintains three thresholds followers_thr, non_followers_thr, other_viewers_thr.

interests_thr – SPS maintains a threshold (percentage of uploaded or watched videos) in order to determine if a user is interested in a specific category of interest.

For each connected user the SPS monitors by means of the Social Awareness (see module D3.1 [3] and stores in a local DB the following information per user:

- **viewers** – list of viewers (user_id’s)
- Lists of users (user_ids) in three viewers’ categories: followers, non_followers, other_viewers
- **viewers_watched_perc** – list with the actual percentage of the video chunks viewed by each one of his viewers.
- **interest_categories_distribution_table** – table of interest categories. In each position of the table is stored the actual percentage of the videos have been watched and belong to the corresponding category.
• **tableof_messaging_overlays_ids** – Table of messaging overlays of this user (messaging_overlay_id's). The size of table is fixed and equal to the number_of_interest_categories. If the user is not interested in one category, the respective position in the table is set to zero. While, for the categories the user is interested in is set equal to the corresponding messaging_overlay_id.

**activity_information (by monitoring)** – SPS keeps activity information for all the user are connected to him. This information is kept in as a list of events. Each element of the list is a record and each record defines the:

- type of the event: Upload, Re-share, Watch
- source and destination (only for watch and re-share events) user of the event
- content_id of the content where the event refers and interest_category this content

**messaging_overlays** (messaging_overlay_id's) – SPS constructs a list to maintain all messaging overlays and their ids. Each element of this list, contains the id of the messaging overlay referring to and a list of this messaging overlay's users.

**stored_videos** – list of videos (content_id) are stored in the SPS’s cache

**stored_videos_caching_parameter** – list of caching parameters. For each video stored in the SPS we maintain a counter representing the possibility to get a request for this video in near future. When SPS push a prefix of a video to a user, this parameter increases. When the SPS receives a watch request for a video, the respective parameter decreases.

**stored_prefixes** – list of video prefixes (content_id) are stored in the SPS’s cache

**p2p_swarms** – SPS maintain a list of local swarms for each piece of content have stored in its cache cache. Especially, for each swarm SPS maintains:

- the **swarm_id** and a **list of users** participating in this swarm (list of user_id's)
- **participation_counter** – This counter is increased when a user requests the SPS to participate in the respective swarm for better QoE and decreased when a user send request to stop participating. For each swarm the SPS participates only in swarms where the counter is > 0.

**Client**

**user_id, SPS_id** – the id of SPS the user is connected to.

**tableof_messaging_overlays** (messaging_overlays_id's) – Table of messaging overlays (messaging_overlays_id's) of this user. This is the table is kept in the SPS side. Periodically, SPS updated this table and sends it to the user. For the categories the user is interested in the respective position is set equal to the corresponding messaging_overlay_id.

**user_messaging_overlays** – a number of lists, each one of them has a unique messaging_overlay_id. This id's contained in tableof_messaging_clusters.

**stored_videos** – list of videos (content_id) stored in the user’s UD.

**stored_prefixes** – list of video prefixes (content_id) stored in the user's UD.

**p2p_swarms** – lists of local swarms (list of swarm_id's) the user participating.

13.6.7 Methods

**SPS**
Initialization methods

set_thresholds(followers_thr, non-followers_thr, other_viewers_thr, interests_thr)
  o set percentages in all the thresholds are needed: followers_thr, non-followers_thr, other_viewers_thr, interests_thr.

viewers_categorization (user_id, viewers, viewers_watched_perc)
  • Based on the videos percentage a viewer watches from a user and on viewers thresholds, we distribute viewers into: followers, non-followers, other viewers.

clusters_formation (user_id, followers, non_followers, other_viewers, interest_categories_distribution_table)
  • For each category of interest, if the percentage of the videos the user watches or uploads is over the interest_thr, then SPS creates for this user a messaging overlay for this category of interest.
  • In each messaging overlay, the SPS adds the followers and only the non-followers that are interested in the respective video category
  • Updates the tableof_messaging_overlays_id of the user and the list of messaging overlays.

Methods activated by events (Internal functions executed by the SPS)

bind_response (SPS_id, user_id)
  • after getting a request from a user, the SPS adds the user to usersID_list
  • maintain information for this user interaction with other users

response_clusters (user_id, tableof_messaging_overlays_ids, messaging overlays)
  • Responds to user's request for clusters with
    o tableof_messaging_overlays_ids, assign clusters to interests
    o and a number of lists of users, the messaging overlays

response_requested_video (content_id, user_id, source_user_id)
  • request_and_store_video
  • if swarm for content_id exist add_to_swarm
    else create new swarm and add_to_swarm
  • send response with swarm_id and decreases the caching parameter of this content
  • updates activity information in source and destination user

add_to_swarm (content_id, user_id)
  • add user_id in swarm with content_id and update p2p_swarm list

response_requested_prefix (content_id, user_id)
  • pushes prefix of video with content_id to user with user_id.
  • increases the caching parameter of this content

request_and_store_video (content_id)
• requests video from the video server
• stores video with content id and updates stored videos list
• if cache is full and a content deleted due to caching policy SPS leave_swarm for this deleted content.

**request_and_store_prefix** (content_id)
• requests prefix from the video server
• stores prefix with content id and updates stored prefixes list
• if cache is full eliminate the least recently arrived prefix

**response_quit_swarm** (swarm_id,user_id)
• remove user_id from swarm with swarm_id and update p2p_swarm list
• If the swarm is empty delete_swarm

**join_swarm** (swarm_id): SPS start participating in the swarm for content_id

**leave_swarm** (swarm_id): SPS stops participating in the swarm for content_id

**delete_swarm** (swarm_id): remove the list for content_id from p2p_swarm lists.

**response_SPS_p2p_participation** (swarm_id, mode)
• if mode==start
  o Increase participation counter
  o if SPS does not have the content_id request_and_store_video
  o SPS join_swarm for support
• If mode==stop
  o decrease participation counter
  o if participation counter == 0 leave_swarm

**caching_policy** (stored_videos_caching_parameter)
• considering the two least recently arrived video we choose to eliminate the one with the lower stored_videos_caching_parameter.

*Periodically or continuously executed methods*

**update_interests** (user_id, activity_information)
• Update interest_categories_distribution_tablebased on activity information.

**update_viewers** (activity_information)
• Update the viewers_watched_perc based on activity_information.
• For each user calls: viewers_categorization, clusters_formation

*Client*

**Initialization methods**

**bind_request** (SPS_id ,user_id)
• user with user_id requests to connect to the local SPS with SPS_id
request_clusters (user_id, SPS_id)
  - request from SPS (SPS_id) to send him updated clusters belongs to user_id.

clusters_formation (user_messaging_clusters, tableof_messaging_clusters)
  - Each client, after its clusters, updates the user_messaging_overlays and tableof_messaging_overlays using the information he received.

Methods activated by events

request_video (user_id, content_id, SPS_id, source_user_id)
  - user with user_id requests video with content_id from SPS with SPS_id
  - and SPS adds user to a swarm for this content_id
  - also informs the monitoring tool of SPS (source_user_id, content_id)
  - QoE_monitoring for possible request to SPS for p2p participation
  - After watching the video, if the support of SPS had been requested
  - SPS_p2p_participation with mode=start and store_video

request_prefix (user_id, content_id, SPS_id)
  - user with user_id request prefix of the video with content_id from SPS_id.

send_alert_message (user_id, content_id, cluster_id): user with user_id sends an alert message to all users in cluster with cluster_id for the content with content_id.

store_prefix (content_id)
  - user stores the prefix pushed by SPS and updates the stored prefixes list.
  - if cache is full eliminate the least recently arrived prefix

store_video (content_id)
  - user stores in his UD the video watched and updates the stored videos list.
  - if cache is full eliminate the least recently arrived video
  - and then quit_swarm for the eliminated video

join_swarm (swarm_id, SPS_id): user joins the swarm with swarm_id

quit_swarm (user_id, swarm_id, SPS_id)
  - requests from SPS to delete him from the swarm for content_id

SPS_p2p_participation (user_id, swarm_id, mode)
  - user requests from SPS to participate in swarm when mode=start

  - user requests from SPS to stop participating in swarm when mode=stop

QoE_monitoring (user_id, swarms_id): if the bit rate of received data is below the video’s bit rate SPS_p2p_participation with mode=start.

Periodically or continuously executed methods

update_clusters (user_id, SPS_id): request_clusters, clusters_formation

wait_alert_message(): the client waits to receive an alert message, to request a prefix.
13.6.8 Communication

User Messages

BindReq: {user_id}: The user with id user_id sends request to the local SPS for binding. The message contains the user_id of the requestor.

ReqCL: {user_id}: The user with id user_id sends request to the local SPS to send the up-to-date clusters belonging to him. The message contains the id of the user.

AlertMess: {content_id}: An alert message send from a user through a messaging cluster to his audience and the only information these message carry is the id of a content item. Then, the receiver should request the prefix of this content.

ReqPref: {content_id, user_id}: The user, who got an alert message, requests the prefix of the content_id from the SPS.

ReqVid: {content_id, user_id}: When a user demands to watch a video, this message is sent to the SPS.

QuitSwarm: {user_id, swarm_id}: When a user deletes a video from his UD sends this message to SPS to inform that he quit the respective swarm.

SPSPart: {user_id, swarm_id, mode}: This message is sent from a user to request support by SPS in video streaming when the user experiences QoE degradation. Also, the same message can be send to inform the SPS that the user does not need support any more. The mode parameter (start/stop) determines the kind of request.

SPS Messages

BindResp: This message is the response to the request of a user for binding. It does not contain anything, it is a confirmation.

RespCl: {clusters, table}: This message is the response to the request of a user for his up-to-date clusters. The SPS creates the new clusters and the table maps these clusters in user’s interest categories and sends the clusters and the table to the requestor.

RespPref: {content_id, prefix}: This is the response to a user’s request for a prefix. The SPS respond with a message containing the prefix and the id of the content.

RespVid: {swarm_id}: When a user requests a video, the SPS respond with this message containing the id of the local swarm formed for this content.

Algorithms

Initialization process
Figure 13-35: The sequence diagram demonstrates the communication between a user and the local SPS during initialization time.

**Socially-aware messaging overlay construction algorithm**

```
Socially-aware messaging overlay construction algorithm
{
    User: requests_clusters
    SPS: viewers_categorization
    SPS: clusters_formation
    SPS: response_clusters
    User: clusters_formation
}
```

**Socially-aware pull based prefetching algorithm**

```
Socially-aware pull based prefetching algorithm
{
    User AS1: send_alert_message
    User AS2: request_prefix
    SPS AS2: request_and_store_prefix
    SPS AS2: response_requested_prefix
    User AS2: store_prefix
}
```
Local P2P overlay construction algorithm

{  
  User: request_video  
  SPS: response_requested_video  
  User: store_video  
}
13.7 Detailed specification of vINCENT

We define a number of network operations to be implemented by the different entities. However, we leave the authentication of users with Facebook, as these procedures are well defined by Facebook (see terminology Access Token). For the sake of simplicity, we assume the Facebook login has happened on all entities and thus, all entities possess an Access Token to read the user’s social data. Moreover the Facebook API offers means to map access tokens to user IDs. We use Access Token* to signal an implicit mapping of an access token to a user ID.

13.7.1 PING

The uNaDas running the HORST-VINCENT software have to register with the Facebook APP, once they are online. Regular PING messages ensure that the Facebook APP always has an updated view of all uNaDas online, their location, and the social relations of their owners. Thus, the PING is coupled to the Access Token and GPS location of the uNaDa in order to map uNaDa IP and user. The GPS coordinates are configured by the user and are necessary to guide HORST-VINCENT users to nearby uNaDas as well as for the incentive mechanism (see Figure 13-37).

![Sequence Diagram](www.websequencediagrams.com)

Figure 13-37: PING messages ensure that the Facebook APP always has an updated view of the uNaDas online, their location, and their social relations.

The procedure is repeated every t seconds. If a uNaDa is not seen after this period, it is assumed to be offline. The uptime of the uNaDa is recorded in the History.

13.7.2 TUNNEL SETUP

The Mobile APP initiates the setup of a tunnel to a uNaDa from the trusted set of the mobile user, which is selected by the Facebook APP. The Facebook APP can select this user, as it knows user’s trusted sets and the respective IPs and can map these sets to a terminating uNaDa based on the information received through the PING procedure. For the sake of clarity, we use the following mapping:

- untrusted uNaDa(A) is the IP of the untrusted uNaDa providing the user of Mobile APP physical access.
- trusted uNaDa(B) is the selected trusted terminating uNaDa.
Messages for tunnel setup that can be overheard by the HORST-VINCENT(A) need to be transmitted over a secure connection (i.e., HTTPS). Every successful tunnel setup is recorded in the Facebook APP’s history of A.

**13.7.3 TRUSTED SET MANAGEMENT**

For the management of the trusted set from the mobile device, we use Facebook directly. For this purpose, the well-documented functions of the Facebook GRAPH API for friend list management are to be used [77], which then also defines the communication pattern and functions to be invoked. The API allows creating/deleting friend lists and updating the members of such a list, which fits our purposes. In the following, we assume a list called “Trusted Set” to exist.
13.7.4 Invoked Methods

13.7.4.1 AddToOnlineList()

The method `void : AddToOnlineList(Access Token*, timestamp, IP, GPScoordinates)` manages a List of uNaDas online, their IPs, GPS location, and their mapping to Facebook User IDs.

**Online list update Algorithm:**

- **Call:** On every received PING.
- **Purpose:** Manage the current view on uNaDas online and manage the history of uptimes of uNaDas.
- **Input Parameters:**
  - `Access Token*` is a Facebook access token, which is implicitly mapped to a Facebook user ID.
  - `timestamp` is the current system time.
  - `IP` is the source IP of the PING.
  - `GPScoordinates` the GPS coordinates of the uNaDa.
- **Algorithm:**
  - If the uNaDa is online for the first time, add a new entry for this uNaDa to the list of online uNaDas
    - Entry tuple:
      - `user=Access Token*`
      - `geoposition=GPScoordinates`
      - `firstonline=timestamp`
      - `currentIP=IP`
      - `online=true`
      - `lastseen=timestamp`
      - `timeonline=0`
      - `offloadedconnections=0`
  - Else update the list, where `user=Access Token*`
    - Update fields:
      - `lastseen=timestamp`
      - If (online==true)
        - set `timeonline=timestamp-timeold`
      - `online=true`
- **Return:** -

**Online list invalidation algorithm:**
• **Call:** In regular intervals (every \( t \) seconds).

• **Purpose:** Keep track of uNaDas going offline.

• **Input Parameters:** -

• **Algorithm:**
  - For each entry in list of devices online
    - If (\( \text{online}==\text{true} \) && \( \text{lastseen}-\text{timestamp} > t \)) set online status to false for this entry

• **Return:** -

### 13.7.4.2 SelectTerminatingUNaDa()

The method **IP Destination : SelectTerminatinguNaDa(IP Source, Access Token*)** selects a terminating uNaDa as an end point for the VPN tunnel. Internally, this method uses four different methods with decreasing priority, to determine the best terminating endpoint from the set of trusted users.

**Terminating UNaDa selection algorithm:**

• **Call:** Whenever a terminating uNaDa is to be selected.

• **Purpose:** This algorithm selects a terminating uNaDa from a user’s trusted set in a provider friendly way.

• **Input Parameters:**
  - IP Source is the source IP of the uNaDa providing physical access to the Mobile APP user.
  - Access Token* is the access token mapped to a Facebook User ID.

• **Algorithm:**
  - Candidates = Select those IPs from the OnlineList, that are part of the mobile user’s Trusted Set.
  - If IP Source is part of the set of candidates, return IP Source, i.e., the Mobile APP is offloading to a trusted uNaDa, which may terminate the connection directly.
  - Use one of the following methods to determine the terminating uNaDa with the smallest distance in terms of AS hops with a decreasing priority (i.e., AS Matching > ALTO > RB_TRACKER > GEODISTANCE). All of these methods sort the candidate list in decreasing order (i.e., the candidate with the smallest distance comes first).
    - **AS Matching:** Walk list for a candidate in the same AS is . If a candidate is present, take this user as the terminating uNaDa.
    - **ALTO:** If the ISP of the uNaDa used for offloading offers an ALTO interface, the set of candidates IPs is sent to the ISP to determine the best IP from the ISPs perspective. We assume the ISP to prefer candidates with a minimal hop count.
    - **RB-TRACKER:** The RB-Tracker mechanism maintains an overview of the distance of the uNaDas in its neighborhood. The set of candidate
IPs and asks the uNaDa used for offloading to determine the closest IP. This includes the possibility, that the uNaDa does not know any of the IPs.

- **GEODISTANCE**: If AS Matching ended without a result, ALTO is not available and RB-TRACKER cannot rank the list of candidates, the geographical position of all uNaDas is used to calculate the geographical distance to determine the closest IP from the set of candidates. This is the default fallback method, which we assume to perform still better than random selection.
  
  - Increase offloaded connections counter for uNaDa with IP source.
  - Return IP Destination.

**Returns:**

- **IP Destination** is the destination IP to which the tunnel is to be established.

### 13.7.4.3 GenerateSharedSecret()

This method generates and returns a sufficiently large random shared secret (String) for the L3TP VPN tunnel setup.

### 13.7.4.4 GetRate()

The method **bit rate : GetRate(Access Token*)** maps the history of the user to a granted service level, i.e., a bit rate granted to the user at untrusted uNaDas. The mapping is done using a function to be parameterized by the Facebook App. The function takes into account:

- **Trusted Set Size** (TSSAccessToken*): The size of the offloading user’s trusted set. This parameter is received from the user’s trusted set list via the Facebook Graph API.

- **Offloaded connections** (OCAccessToken*): This value is retrieved from the online list.

- **User Density at the GPS location of the uNaDa** (UDAccessToken*): This value is retrieved from a Geo database. As these services are usually not for free, an artificial dataset may be used in the first place. In a later deployment, more geographical data may be taken into account to increase precision.

- **Uptime of uNaDa** (UTAccessToken*): This value is retrieved from the online list.

**Notation:** the index AccessToken* always refers to the value of user AccessToken* whereas no index refers to an average of the whole population of users.

**History data preparation:**

- **Call**: In regular intervals, e.g., once per day before running the geoposition compensation algorithm.

- **Purpose**: Prepare history data for regression modelling (see next algorithm).

- **Input Parameters**:
  
  - For all uNaDas from the online list: TSSAccessToken*
  - For all uNaDas from the online list: OCAccessToken*
  - For all uNaDas from the online list: UDAccessToken*
Algorithm:
  - For each input parameter P ...
    - Calculate average avg(P) and standard deviation sd(P) of the whole set
    - For each uNaDa ... //Outlier elimination
      - If P\textsubscript{AccessToken},dt is greater or smaller than avg(P)+-2*sd(P), remove this uNaDa from all four input data sets
  - Returns: The four input parameter data sets, which are filtered for outliers.

Scoring weights estimation algorithm:
  - Call: In regular intervals, e.g., once per day.
  - Purpose: This algorithm determines the parameterization of the scoring function based on observed data. It determines what number of offloaded connections (OC) can be expected on average given the user’s parameters.

Regression model: \( OC'\text{AccessToken},dt = b_0 + b_1 * TSS'\text{AccessToken},dt + b_2 * UT'\text{AccessToken},dt + b_3 * UD'\text{AccessToken},dt + e \)

Input Parameters:
  - The filtered measurement from the history data preparation algorithm:
    - TSS'\text{AccessToken},dt
    - OC'\text{AccessToken},dt
    - UD'\text{AccessToken},dt
    - UT'\text{AccessToken},dt

Algorithm:
  - Calculate 3 x n matrix X, where \( M_i = TSS_i,dt, M_2i = UT_i,dt, M_3i = UD_i,dt \)
  - Calculate n vector y, where \( y_i = OC_i,dt \)
  - Calculate n+1 vector b using multivariate linear regression, e.g., Matlab’s robustfit function:
    - \( b = \text{robustfit}(X,y) \) with minimum squares criterion
  - Return b

Output Parameters:
A vector $b$ with four elements $(b_0, b_1, b_2, b_3)$ describing the weights of the regression model.

**Remark:** If the linear regression model turns out to be too simple, there might be a need to replace it with higher order models. An implementing team should keep that in mind. Moreover, there might be additional regressors added, in order to increase the precision of what number of offloaded connections can be expected from a certain geo location.

**Rate calculation:**

- **Call:** Whenever a rate to be granted for a certain user is requested.
- **Purpose:** This algorithm calculates a score for the user contributions using the $b$ vector returned by the previously described algorithm. Therefore, the residual for a certain user is mapped to an incentive function determining the reward or punishment in granted service for the respective user.

![Diagram](image-url)

Figure 13-40: $I(r,t)$ determines the height of the incentive when deviating from the average that can be expected from the regression model.

**Input Parameters:**

- $I(r,t) = (r^t) / \sqrt{1+(r^t)^2}$, an incentive function, which determines the reward or punishment of the user’s deviation from the average performance ($r$) that can be expected according to the regression model. The parameter $t$ is determined by simulation/market research and tunes the incentive’s aggressivity.
- A user id AccessToken*.
- The $b$ vector from the scoring weights estimation algorithm.
- An average service level $e$ in kBit/s satisfying a user that is contributing an average service to the system. This parameter is determined by simulation/market research. This parameter might also depend on the type of traffic, which may be taken into account in later versions of the mechanism.
- A maximum negative or positive incentive $c$ in kBit/s to be granted.

**Algorithm:**
- Calculate the residual of the user:
  \[ r = OC_{\text{AccessToken}^*} - (b_0 + b_1 \cdot TSS_{\text{AccessToken}^*} + b_2 \cdot UT_{\text{AccessToken}^*} + b_3 \cdot UD_{\text{AccessToken}^*}) \]
- Return \( e + I(r,t) \cdot c \) as \( r_{\text{AccessToken}^*} \)

**Output Parameters:**
- A data rate \( r_{\text{AccessToken}^*} \) to be granted to a certain user during offloading.

### 13.7.4.5 User Interface

The only entity in need of a user interface is the mobile App. We propose a design using 4 pages. The user starts with a login screen to Facebook to retrieve an access token. After successful login, the next screen shows the currently achieved service level including a rough mapping of the data rate to a mobile network access technology (e.g. HSDPA). A link is leading to a web page explaining the scoring mechanism and what a user can do to retrieve a higher service level. There are two options to be chosen from here: If the App is not connected to a uNaDa at the moment, the “Find Next WiFi” button leads to a navigation page allowing the user to find the next vINCENT access point. The “Friends” button allows the user to manage his trusted friend set on a separate screen.

![Mockup of page design](image)

**Figure 13-41:** Mockup of page design.
13.8 Detailed specification of MONA

The Mobile Network Assistant (MoNA) is a traffic management mechanism focused on the improvement of the energy efficiency of the mobile device, under consideration of the user perceived service quality (QoE). This is achieved by scheduling connections in space and time, based on a-priory network knowledge. Knowing the RTT and throughput of available networks allows to select the most energy efficient network connection, considering the QoE of the end-user. This includes all applications running on the phone. The traffic is managed for foreground and background traffic, although the optimization potential of background traffic is much larger.

![Diagram of MoNA](image)

**Figure 13-42: Overall structure of the Mobile Network Assistant (MoNA) and external interfaces.**

Figure 13-42 shows the general structure of the MoNA approach. In the basic version, MoNA intercepts and manages network requests of all applications on the mobile device. An extended version also supports pre-fetching content, which is triggered by the social content prediction and stored in a local cache. The mobility prediction is required to know future locations and dwell durations of the user. The components drawn in grey are external components, which are required by MoNA, but not core part of the mechanism.

MoNA itself consists of the connectivity manager, controlling the network access and traffic redirects to the local cache for temporary object storage, and the Network Optimizer, containing the logic of the mechanism. Here, blue components are already existing components, while the Network optimizer, marked in green, is the new component.

13.8.1 Mechanism Specification

The Network Optimizer, as the core component of the MoNA mechanism, consists itself of a number of components. Figure 13-43 shows the details of the Network Optimizer. Incoming requests of local applications are analysed by the Request Analyzer, which derives the service’s requirements from the type of service, the device state, and user preferences. The requirements in terms of bandwidth and delay are passed to the Network Model, deriving connectivity options. Using these, the Power Model estimates the cost in terms of used energy for each option. The derived service requirements, combined with
the availability of the network and the estimated energy usage, are used to estimate the QoE of the end user in the QoE model. Based on these estimates, the Network Action Recommender decides, which option to execute via the Connectivity Manager.

The following sections describe the components of the MoNA mechanism and decision criteria in detail.

**Network Optimizer**

Based on the incoming packet, the Network Optimizer determines the originating application and the device state. These are combined with the current network state and using the user mobility prediction to derive possible connection alternatives. These include the offloading from a cellular network to WiFi networks, but also deferring delay tolerant connections until e.g. WiFi is available. For these alternatives, the energy model of the device, determines the projected energy usage. The QoE model checks these connection options to estimate the QoE of each alternative. The Network Action Recommender then selects the option with the lowest energy consumption satisfying the minimum QoE standards as defined by the user and the circumstances.

The Connectivity Manager calls the Network Optimizer with:

- `analyzePacket(Byte[] packet, int processId, long timestamp)`,

passing in the first packet of each connection. This call is executed asynchronously for each incoming packet to not block parallel connections. Consequently, the packet is held back until the Request Analyzer has decided how to handle the connection. This buffer is expected to be quite small, as without server feedback no additional packets for a
connection are expected. A sequence diagram of the interface of the Network Optimizer is given in Figure 13-44.

The Network Optimizer contains a number of components specialized on context retrieval. These are

- Request Analyzer
- Network Model
- Mobility Prediction
- Energy Model
- QoE Model
- Network Action Recommender

Their functionality and calling is detailed in the next sections. A detailed sequence diagram detailing their interaction is given in Figure 13-45. These components are running in their own thread to periodically refresh the available data and register on system callbacks to collect up-to-date information of the system state. For each incoming network request, these augment the information connected with the network request to derive the optimal traffic control action. Hence, the processing done in the pipeline are basically a look-ups and simple calculations, introducing just a minimum delay before executing the derived action.

![Figure 13-44: Sequence diagram of the NetworkOptimizer.](image-url)
Request Analyzer

First, the Request Analyzer is queried to classify the incoming packet by calling

- `classifyPacket(Byte[] packet, Int port, IP destinationIP, DeviceState ds)`
  - Returns the Traffic Class (TC) and the Content Parameters (CP) containing the type of traffic (i.e. file transfer, streaming), and estimated size and duration of the transmission

The packet is classified based on the destination IP, port, and in the case of HTTP connections, the requested URL. The port and IP information helps filtering traffic of the same connection, which is assigned the same traffic class and content parameters as the first packet of the respective connection.

The traffic class is an identifier defining the type of service requested. This may be one of web, video, VoIP, and file transfer.

Network Model

The Network Model estimates possible connection options, based on the traffic class determined by the Request Analyzer and the predicted path as returned by the Mobility Prediction. Each possible predicted path is annotated with a network type, the RTT, and up- and downlink bandwidth. The detailed call is:

- `getNetworkOptions(PredictedPaths pp)`
  - Returns a list with NetworkOptions no, containing the networkType, RTT, up- and downlink bandwidth, and time to wait before the connection is predicted to be available
Within this class, the predicted path is mapped to the network availability as returned by the Network Coverage Server. Combined with the current velocity of the user, the network options are derived. The network option includes the network type, expected RTT, throughput, and time to connect.

**Energy Model**

The Energy Model calculates the corresponding power consumption of each NetworkOption based on the device’s known power model. This calculation is based on the power network type, RTT and available bandwidth, but also the dwelling time in each location. The corresponding call is:

- `getEnergyCost(NetworkOptions no, DeviceState ds)`
  
  Returns a list of PowerEstimates derived from the list of NetworkOptions.

The power calculation is detailed in Section 4.4.2. From the device state, the number of active connections is derived. From the network options, for each possible interface the cost is calculated. The resulting transmission cost is returned.

**QoE Model**

Based on the power estimates and the corresponding NetworkOptions, the QoE Model estimates the QoE for each option. The call is:

- `getQoE(NetworkOptions no, TrafficClass tc, PowerEstimates pe)`
  
  The resulting QoE is estimated as described in [78]. For this, the expected behaviour of the device must be defined for the respective traffic class `tc` and power estimate `pe`. This then is the reference case with a QoE of 4. Better performance increases the QoE up to 5, while poor performance degrades the resulting QoE, until it is unusable with a QoE of 1. This behaviour is to be defined for the network performance and the battery life time individually. By letting the user prioritise performance or energy consumption by setting a weight factor, a combined QoE can be calculated as the weighted sum of both QoE estimates.

**Network Action Recommender**

The QoE estimates are passed into the NetworkActionRecommender to select, which option to execute. As the estimated QoE already includes the power consumption, the list of QoE estimates is sorted by decreasing QoE and the highest option is selected.

- `getAction(NetworkOptions no, QoEEstimates qe)`
  
  Returns the NetworkOption to execute

The NetworkOptimizer takes the returned NetworkOption and executes it via the Connectivity Manager. The basic decision logic is pictured in Figure 13-46. It shows, how each incoming request is handled, based on the current connectivity and the derived QoE.
Connectivity Manager

The Connectivity Manager analyses and controls all traffic generated and received by the device. For this, a transparent proxy on the mobile device, a Linux kernel module, or a virtual interface in combination with *netfilter* rules (implementable as Android VPN service) are possible.

The Connectivity Manager extends the functionality of conventional proxies by allowing delaying incoming connections and hence, reducing the number of transitions of the interface to the high power state. The incoming packets are buffered within the proxy and copied to the Request Analyzer to decide, how to handle the corresponding connection. The parameters of the connection delay are configured by the Network Optimizer.

The Connectivity Manager holds back all packets of a connection, until it is called by the Network Optimizer with the definite decision on how to handle the packets. The function call is:

```
executeAction(action, delayTime)
```

- `action` is one of `{forward, discard}`
- `delayTime` the delay before the specified action is executed

As soon as the decision is received, the MoNA proxy schedules the command to execute the action after the specified time interval. The execution of scheduled connections is detailed in Figure 13-47.

![Diagram](image-url)
The Connectivity Manager also implements functionality to monitor the mobile network performance. This includes the minimum throughput observed when mobile networks are used, and the RTT, derived from processed TCP sessions. This information is fed back to the Network Quality Map, increasing the quality of the network estimates for the current location of the user.

**Local Cache**

The local cache stores the already retrieved items and allows the insertion of pre-fetched items and their later retrieval. For this, a simple interface is provided:

- `storeItem(ItemId id, Item it)`
- `Item it = retrieveItem(ItemId id)`

The ItemId is a unique reference to the stored content, which may be created using the target URL, or a hash of it. The cache is limited in size, as defined in the user preferences. A reasonable size is in the range of a few hundred MBs. For simplicity reasons, the preferred implementation is a LUR cache.

### 13.8.2 External Interfaces

The MoNA mechanism relies on the availability of a remote server, collecting and aggregating coverage samples, but also throughput and ping measurements. These samples are collected by the MoNA implementation either in a passive way by monitoring the network interfaces while connections are active, or by periodically probing the network from the mobile device. As this possibly generates cost and reduces the available traffic volume, active measurements must be limited to a minimum.

**Network Quality Map**

The MoNA mechanism accesses a server storing network quality data. The data can be retrieved from the server, but additional measurements are also to be transmitted to the server to improve the quality of the map.
The server uses a REST interface, which accepts and provides JSON messages containing the measurements. The API calls are:

- `getCoverage(BoundingBox bb, NetworkGeneration nt, Operator op, Resolution r)`
  - returns a 2D array of the signal strength with the requested resolution and bounding box
- `getRTTs(BoundingBox bb, NetworkGeneration nt, Operator op, Resolution r)`
  - returns a 2D array of the RTT with the requested resolution and bounding box
- `getUplinkRates(BoundingBox bb, NetworkGeneration nt, Operator op, Resolution r)`
  - returns a 2D array of the uplink rates with the requested resolution and bounding box
- `getDownlinkRates(BoundingBox bb, NetworkGeneration nt, Operator op, Resolution r)`
  - returns a 2D array of the downlink rates with the requested resolution and bounding box
- `getWiFiAps(BoundingBox bb)`
  - returns a list of WiFi APs contained in the bounding box. Information included is the location of the AP, maximum upload and download rates, and the RTT.
- `getWiFiCoverage(BoundingBox bb, Resolution r, Operator op)`
  - returns a 2D array of coverage of the given network operator. The operator may be one of `{Horst-AP, eduroam, ...}`
- `getWiFiRtts(BoundingBox bb, Resolution r, Operator op)`
  - returns a 2D array of the RTT of the given network operator within the bounding box. The operator may be one of `{Horst-AP, eduroam, ...}`
- `getWiFiUplinkRates(BoundingBox bb, Resolution r, Operator op)`
  - returns a 2D array of the uplink rates of the given network operator within the bounding box. The operator may be one of `{Horst-AP, eduroam, ...}`
- `getWiFiDownlinkRates(BoundingBox bb, Resolution r, Operator op)`
  - returns a 2D array of the downlink rates of the given network operator within the bounding box. The operator may be one of `{Horst-AP, eduroam, ...}`
- `addCoverageMeasurement(location, accuracy, altitude, speed, timestamp, lac, networktype, networkprovider, asc, signalstrengthDb)`
- `addPingMeasurement(location, accuracy, altitude, speed, timestamp, lac, networktype, networkprovider, asc, signalstrengthDb, pingmin, pingmax, pingavg, pingmeandev, pingcount)`
- `addThroughputMeasurement(location, accuracy, altitude, speed, timestamp, lac, networktype, networkprovider, asc)"
The measurements received by the server are the expected metrics at a given point. In the first implementation, these may be simple averages. If enough measurements for a single position are available on the server, the variance of the measurement can also be returned.

**Mobility Prediction**

The Mobility prediction is called with:

- `predictPath(TimeHorizon t)`
  - Returns a list of PredictedPaths pp, with a minimum of one entry containing a list of tuples of time, location, expected dwell time, and probability of visiting the location

The Mobility Prediction is running independently of the Network Optimizer to monitor the user’s location. The simplest, but still sufficiently accurate model uses a 2\textsuperscript{nd} order Markov process to estimate the future location based on past locations mapped to a 100m x 100m grid [79]. Such, a quite accurate prediction of the behavior can be expected after a two week learning period.

**Social Content Prediction**

Likely to be consumed social content is inserted into the download queue by calling

- `addSocialContent(URL url, deadline d)`

This is called by the RB-Horst content prediction module to insert multimedia content of high interest into the local cache. When retrieving the content, and connected to a RB-Horst AP, the content can be retrieved from the local uNaDa.
13.9 Detailed specification of MUCAPS

MUCAPS (MUlti-Criteria Application endPoint Selection) can actually be generalized to several deployment cases of the Multi-Access and Cost AltO (MACAO) handling service. This service is suited to any application having the choice among several candidate Application EndPoints (AEP). It:

1. Captures the initial AEP selection done by the acting AEP Gathering function (AEPG),
2. Identifies the AEP selection metrics and their weights given the application needs,
3. Requests the related transport network aware information via the ALTO protocol,
4. Uses it to rank the initially selected AEPs,
5. Sends the revised AEP selection to the AEPG.

Steps 2, 3, 4 are performed by a functional bloc called MUCAPS. Steps 1 and 5 are done via an interface between a MACAO handling Request Client (MARC) hooked to the AEPG and a MACAO handling Request Server (MARS) hooked to the MUCAPS block. The MARC and MARS communicate via a simple IP socket.

The MUCAPS specification describes an instantiation of the MACAO service where:

- The MARC is hooked an ISP managed AEPG deployed in its network,
- The AEPG is a DNS resolver that communicates with a DNS Client (user AEPG) located in the user device,
- The handling services that a MARC can request via the MARS interface are “AEP ranking” and “e2e path ranking”.

The functional blocs involved in MUCAPS are shown in Figure 13-48.

![Functional blocs involved in MUCAPS](image)

**Figure 13-48**: MACAO with MARC embedded in an AEPG located in the ISP network.

A central technology used in MUCAPS is the ALTO protocol. ALTO Stands for Application Layer Traffic Optimization) and the base ALTO protocol is being standardized by the IETF
as RFC 7285. ALTO provides guidance to applications w.r.t. network topology so that they choose their endpoints by providing applications with awareness on the underlying ISP topology and related costs. This awareness is materialized by information reflecting the ISP-centric view of the Internet expressed by an abstract human-readable description of the network topology partitioned in ISP defined network regions (PIDs), e2e costs among PIDs and possibly amongst Endpoints.

The base ALTO protocol in RFC7285 specifies:

- a number of ALTO services: Network Map, Cost Map, Endpoints Cost
- a transport means to convey this information between the ALTO Client and Server.

Application traffic needs to be guided w.r.t. several metrics that report not only ISP policy but also offering compromises and better QoE. The proposed “Multi-Cost ALTO” protocol extensions that support this are likely to be adopted by the ALTO WG. They allow to convey several metrics simultaneously in ALTO requests and responses, specify filtering constraints on several metrics and logical operators and propose a set of QoE impacting abstracted metrics, beyond the base protocol ‘routing cost’ and hop count so as to better support use cases such as Data Center or CDN.

In all the currently defined deployments, the MUCAPS includes the following blocks:

- An ALTO Client
- An Application Metrics and Weight Setting function (AMWS)
- A Multi-Criteria AEP evaluation function (MCEVAL)
- An ALTO Agent (AoA)
- An ALTO optimization information base (AOIB)

The MUCAPS functional block communicates with an ISP managed ALTO Server via its ALTO client.

13.9.1 Details on the MUCAPS related functional blocks

Figure 13-49 shows details on the functions involved in MUCAPS, while the operations sequencing will be explained in a further section.
Figure 13-49: Details on the functions involved in MUCAPS and operations sequencing.

The **MACAO Request handling Server (MARS) InterFace (IF)**, bridges the MUCAPS block with the MARC requesting evaluation and/or ranking of (S,D) pairs or application paths and providing the initial selection of (S,D) pairs or application paths together with the ID of the MACAO Service to request: typically “Endpoint Ranking and/or Evaluation” or “Path Ranking or Evaluation”. The MARS IF returns the new evaluation and/or ranking of (S,D) pairs or application paths done by MUCAPS. The MUCAPS block communicates with the ALTO Server via its embedded ALTO Client.

The MARS receives and services requests issued, via an IP Socket, by a MARC that is hooked to an AEPG function. For requests issued by “un-trusted” MARCs located in “public” devices, the services include at least “EP ranking” and “e2e Path ranking”. These services map respectively to the ALTO EP Cost Service and the ALTO Cost Map Service. The evaluation services providing “real-values” are preferably restricted to “trusted” MARCs since they provide numerical ISP scores or costs and are less transparent. It returns to the MARC, via the MARS IF, the list of (S,D) EPs with requested evaluation information: generically the rank of (S,D) pairs or e2e paths.

The **MACAO Request handling Server (MARC)** communicates with the MARS IF through an IP socket, and exchanges a light data structure containing at least: the MACAO Service ID, typically a Multi-Cost ranking or Evaluation Service, the list of application paths or (S,D) endpoint pairs to evaluate, and fields for their values and/or rank. The MARC is typically hooked to the function providing the initial selection of application endpoints or paths, called AEPG for Application Endpoints Gathering Function, that can itself be located in various places such as end user devices, the application network or its ISP managed part. The MARC may customize its response to the calling AEPG function so as to mimic the format of the values returned by the AEPG function so as to make the MARS operations transparent to the applications.

The **AMWS** function selects the ALTO metrics w.r.t. the application type and possibly time. It also sets the weight associated to the metrics, w.r.t. application needs or context information related to the UEP such as access type. The AMWM can **optionally** fine-tune
the metric weights w.r.t. the network access type and possibly time. The AMWM maps metric ID and weight values to the application and other information contained in the MARC currently by using a Look Up Table. The values of this LUT are periodically updated from an ISP managed AMWM configuration setting function.

The AMWS Config Settings function is a Graphical User Interface (GUI) integrated in an end system also to access the AMWS Function. This function is to be managed by the ISP. Via the GUI, the “ALTO ON” the “ALTO auto metric” and the “ALTO auto weight” modes are activated. As an option, it updates the values associated to the configurations mapped to groups of AEPs. For example: a group of AEP identified by their IP addresses is identified as Video Servers. AEPs of this group must therefore be selected w.r.t. a set of metrics including typically “routing cost” and “bandwidth score”. However, in some circumstances as low peak hours, the ISP may consider that “bandwidth score” does not need to be involved in the selection, or with a lower weight. Thus, the LUT always points to the same file but the latter may have a varying content. A simplifying option is however to set the values of the configuration files of the AMWS function as constant and avoid thus implementing complex interactions.

The Multi-Criteria AEP evaluation function (MCEVAL) ranks the EPs w.r.t. the metric values and their weights. It uses a vector-based multi-objective ranking method, based on the proximity of a candidate solution to some “ideal” solution.

The ALTO Server
The ALTO Server serves requests received by ALTO Clients according to the capabilities exposed in its Information Resources Directory and specified by the ALTO protocol (RFC7285)

The ALTO Client
The “ALTO Client” then sends a request to the ALTO Server to get cost values among the source and destination (S,D) endpoints (or among end to end paths for the Path Ranking MARS Service). Its behavior complies to the specifications of RFC7285.

The ALTO optimization information base (AOIB)
This function stores, according to their “freshness”, Multi-Cost ALTO responses as well as the evaluation and ranking results on paths and endpoints pairs, so as to faster serve frequently repeated queries on the same set of (S,D) pairs or paths.

The ALTO agent (AoA)
The AoA is a central function that is hooked with all the other functions for the sake of modularity and robustness: it knows the semantics of ALTO and of the functions using the ALTO information, passes information and requests among the entities it is hooked with. The AoA:

- Receives a Service request via the MARS Interface (IF) along with the MARS Service ID, the list of candidate paths or source and destination endpoints (S,D) EPs,
- Checks its “ALTO optimization information base” to see if the needed ranking is already available and therefore still valid,
- If no stored response is available, the AoA prepares the arguments for the Multi-Cost ALTO request to be sent by the ALTO Client: (ALTO Service ID, list of (S, D) EPs, cost metrics, cost mode by default numerical).
  - To identify the cost metrics, the AoA calls the AMWS function,
  - The ALTO Cost mode is by default “numerical”
  - The (S,D) pairs are included in the request sent by the MARC
The ALTO Service ID is mapped from the MACAO Service ID (path ranking/EP ranking)

- Once the ALTO Client requests and gets the information from the ALTO Server it hands it to the AoA that translates it into a form exploitable by the MCEVAL.
- The AoA sends the ALTO-based EP information to the MCEVAL, together with the applicable metric weights,
- Once the MCEVAL has performed its multi-criteria evaluation it hands the result to the AoA that translates it into a format exploitable by the MARS so as to be sent back to the MARC.

13.9.2 Network-based MUCAPS implementation and execution

In the current MUCAPS deployment, the MARC is located in the ISP network. This is the case for instance when the ISP wants to ensure that the DNS selection of the application network is compliant to its policy in resources usage. In that case, the ISP captures the response of its DNS Resolver, processes it with the MACAO module and sends the MACAO selection to the DNS Resolver that sends it to the DNS Client. All this is done in a transparent way for end users and applications.

The network-side deployment enables a lightweight transaction between MUCAPS and the ISP DNS resolver by: adding to the MUCAPS block a MARS service interface, becoming thus a MACAO Request Server (MARS). This deployment includes the following steps:

1. The application client queries the DNS server for the list of content servers to download/upload from.
2. The DNS Server,
   2.1. selects the MACAO Service “Endpoint Ranking”
   2.2. hands to its embedded MARC its query result, that is the list of the candidate Application Endpoint IP addresses, together with the IP address of the User Endpoint, that is the DNS Client associated to the user device,
3. sends this light information to the MARS IF via an established IP Socket, as a query to perform the multi-cost ranking of the AEPs,
4. The MUCAPS block performs the ranking and the MARS sends the reordered list of AEPs to the MARC.
5. The MARC passes it to the AEPG/DNS Server that sends it to the requesting AEPG/DNS Client.
Figure 13-50: MARC integrated in a network located AEPG, here a DNS server.

Figure 13-50 shows the sequence diagram between the entities involved in the MACAO AEP ranking. The acronym MSR stands for MUCAPS Service Request and represents a simple object structure sent by the MARC to the MARS and containing the following fields:

- int service_type; // presently Endpoint ranking
- char source_endpoint[1][]; // IP address of the User Endpoint
- char application_endpoint[][]; // IP address of the AEPs
- int num_of_application_endpoint;

Figure 13-51: Interaction between entities involved in the MACAO AEP ranking

**Detailed sequencing of MUCAPS operations on an example scenario**
The MUCAPS operation involves the following steps involving the functions depicted in Figure 13-51. As an example scenario, we may assume that the application is video streaming with 3 candidate AEPs and 2 metrics which are “routing cost” and “bandwidth score” in numerical modes. The AEPs gathered by the DNS will be ranked w.r.t. the values of:

- multiple metrics selected by looking up the AMWS function and their values provided by the ALTO Server.
- The metric weights adjusted w.r.t UE context parameters such as access.

1. For every downloaded chunk the DNS has returned the IP addresses of around 3 eligible AEPs which are video streaming servers using TCP as a transport protocol,
2. The MARC receives the DNS output, that is the IP address of the candidate AEPs and builds a MUCAPS Service Request (MSR),
3. The MARC then sends the MSR to the MARS, that calls the ALTO Agent (AoA),
4. The ALTO Agent gets the MSR and injects it in its own data structure. It then passes the set of AEPs to the AMWS,
5. The AMM LUT maps the 3 AEPs to Class "IP_1" pointing AMM-config_1.
   a. AMM-config_1 points at this time to the value stored in the AMM active ALTO config DB and equal to: “{(routingcost, numerical), (bandwidth cost, numerical)}”, which both are cost metrics supported by the Serving ALTO Server.
      i. In this scenario, during low peak hours, the pointed value is equal to “{(routingcost, numerical)}”, because the AMM ALTO metric config DB stores a schedule for the pointed values.
      ii. The schedule for the pointed values, the ALTO metric configuration updates are uploaded to the AMM ALTO metric config DB via the AMWS config settings – GUI
6. The AMWS function maps the address of the UE to “LAN/WiFi” as the ISP recognizes the IP address of UE as the one of a WiFi access point and this mapping is reported in the AMW LUT. The weight factor for metric “bandwidth cost” at this time is set to double the influence of bandwidth which is weaker during low traffic hours.
7. The AOA converts the metric IDs obtained from the AM LUT into ALTO semantics and is passed them the ALTO Client as one of the ALTO request parameters for the ALTO Endpoint Cost Service
8. The ALTO Client gets the AEP cost values from the ALTO Server and hands them to the AOA,
9. The AOA passes the weight information from the AMW LUT to the Multi-Criteria AEP evaluation function (MCEVAL) as one of its input parameters together with the metric values obtained via the ALTO Client,
   a. Note that the weight is not needed for the ALTO transaction
10. The ALTO Agent gets the ranking result and hands back to the MARS a list of IP addresses ordered w.r.t. the setting of the AMWS function.
11. The MARS sends this selection in its response to the MARC.
12. The DNS gets the updated choice and sends it to the UE and the MUCAPS operations performed in the meantime are totally transparent.

Implementation
The MUCAPS deployment has led to the following implementations:
- the MARC, MARS and MUCAPS block have been implemented in C
- the DNS Server code is the open source “djbdns” suite implemented by D.J. Bernstein, that allowed thus to integrate the MARC and its interaction with it.
- the ALTO Server is a reference implementation in Java.