Objective FP7-ICT-2007-1-216041/D-5.4

The Network of the Future

Project 216041

“4WARD – Architecture and Design for the Future Internet”

D-5.4
In Network Management for Generic Path

Date of preparation: **10-06-14**
Start date of Project: **08-01-01**
Project Coordinator: **Henrik Abramowicz**
Ericsson AB

Revision: **1.0**
Duration: **10-06-30**

Deliverable D5.4:
In Network Management for Generic Path

Document Properties:

Document Number:	FP7-ICT-2007-1-216041-4WARD / D-5.4
Document Title:	D-5.4: In Network Management for Generic Path (Final report of TC45)
Document responsible:	Lúcio Studer Ferreira (IST)
Authors:	Sabine Randriamasy (ALUF), Gehard Hasslinger (DT), Michael Kleis, Fabian Wolff (Fraunhofer), Augusto Neto, Sergio Figueiredo (IT), Lucio Studer Ferreira, Luis M. Correia (IST-TUL), Rebecca Steinert, Daniel Gillblad, Bjoern Levin, Anders Gunnar (SICS), Leonard Pitu, Carmen Blendea, Iulian Iliescu (SROM), Slawomir Kuklinski (TPSA), Virgil Dobrota, Andrei Bogdan Rus, Zsolt Polgar (TUCN), Avi Miron, Danny Raz, Reuven Cohen (Technion), Ning Wang (UniS), Thorsten Biermann, Holger Karl (UPB), Christopher Foley (TSSG)
Target Dissemination Level:	Public
Status of the Document:	Final
Version	1.0

Abstract:

This is the final report of TC45, reporting on the synergies and integration of WP4 “In Network Management” (INM) and WP5 “Forwarding and Multiplexing for Generic Paths” (GP) concepts. An architectural perspective of INM-GP is given, showing how both architectures fit together, easing integration and enhancing usability. Nine collaborative clusters are presented, showing synergies and integration of WP4 and WP5 research activities, and being discussed under the light of the INM-GP architectural perspective.

Document History:

Revision	Date	Issued by	Description
0.1	14/01/2010	Lúcio Studer Ferreira	First draft of the final report, based on R-TC45.1. It includes a section entitled “Achieved cooperation” where the cooperative work shall be reported.
0.2	07/04/2010	Lúcio Studer Ferreira	Inclusion of contribution of chapter 2, “INM GP – An Architectural Perspective”, contributed by Christopher Foley (TSSG) and Holger Karl (UPB)
0.3	20/04/2010	Lúcio Studer Ferreira	Restructuring of the deliverable, with one section per cooperative topic that shall describe the outcome of the cooperation. Individual researchs shall be shortly highlighted, referring to deliverable R-TC45.1 for more details
0.4	25/05/2010	Lúcio Studer Ferreira	First draft, congregating the results of the nine cooperative clusters. The document is ready for review within WP4 and WP5.
0.5	31/05/2010	Lúcio Studer Ferreira	Final version of the deliverable, addressing review comments from WP4 and WP5 partners. Ready for review by the POETS.
1.0	08/06/2010	Lúcio Studer Ferreira	Final version of the deliverable.

Executive Summary

TC45 WP4/WP5 Joint Task is a cooperative platform that aims to identify synergies and establish cooperative activities between WP4 “In Network Management” (INM) and WP5 “Forwarding and Multiplexing for Generic Paths” (GP). It addresses the use of INM monitoring capabilities and external services within GP, the investigation of advanced distributed congestion control schemes, and self-adaptive and resource optimisation mechanisms based on real-time monitoring. A first deliverable, R-TC45.1, identified and congregated individual contributions from WP4 and WP5 that are under this umbrella. TC45 has also contributed to WP5 deliverable D-5.2 and WP4 deliverables D-4.3 and D-4.5. The current deliverable is the final report on the activities carried within TC45. First, an architectural perspective of INM GP is given, showing how both architectures fit together, easing integration and enhancing usability. Then, the outcome of nine collaborative clusters involving various WP4 and WP5 research activities is presented.

An outline of the cooperative clusters is presented next. A section addressing “INM supported synchronisation mechanisms for distributed adaptive routing updates” presents how the distributed Multi-Path Routing kernel (the main outcome of WP5 Task T5.3 “Realising a generic path by routing”) is supported by several INM management capabilities specified in WP4. A section on “Wireless Monitoring and Self-Adaptive Routing in WMNs supported by GPMR” evidences how the INM and GP architecture support the transparent articulation of independent aspects, such as monitoring and routing. A section entitled “INM support of the Cooperation and Coding Framework” (CCFW), discusses how the CCFW (the main outcome of WP5 task T5.2 “Generic paths by cooperation and coding mechanisms”) may use remote information available in the INM system. A section entitled ”INM Cross-Layer QoS Used in Network Coding-Based GP” provides solutions for controlling the congestion employing network coding techniques based on the network state information, including anomaly related information, provided by the INM. A section entitled “Adaptive resource optimisation”, presents a resource optimisation algorithm that is adaptive to long-term changes in the network. A section entitled “INM-enabled multicast GP for fast failure recovery” improves the re-routing process and utilisation of backup paths in multicast-trees, based on the availability information provided by the distributed anomaly detection mechanism. Section “INM & GP joint applications in real network environments” presents the virtual cognitive radio switch that combines several strategies from both WP4 and WP5. Section “Traffic management support by INM: Demands, progress & business value” describes the role of INM operations in Future Internet traffic engineering.

Table of Contents

List of Figures

List of Tables

List of Acronyms

(S)NTP	(Simple) Network Time Protocol
AD	Anomaly Detection
A-HRP	Adaptive Hybrid Routing Protocol
AM	Adaptive Monitoring
AODV	Ad hoc On-Demand Distance Vector
AOMDV	Ad hoc on demand multipath distance vector routing
BER	Bit Error Rate
CAPEX	CApital EXpenditure
CAVB	Currently Available Bandwidth
CCFW	Cooperation and Coding Framework
CLQ	Cross-Layer QoS
CT	Compartment
DANSERS	Distributed Adaptive Network SEnsitive Routing Synchronisation
DAR	Distributed Adaptive Routing
DARP	DAR Protocol
Dl	Delays
DoS	Denial of Service
DSDV	Destination-Sequenced Distance-Vector routing
Dt	Dates
ETT	Expected Transmission Time
ETX	Expected Transmission Count
FRR	Fast ReRoute
GP	Generic Path
GPMR	Generic Path Management Records
HRP	Hybrid Routing Protocol
INM	In Network Management
LTD	Link Transit Delay
MC	Management Capability
MGD GP	Managed GP
MPLS	Multiprotocol Label Switching
MPR	Multi-Path Routing
MR	Master Record
NATO!	Not All aT Once!

NC	Network Coding
NH	Next-Hop
NNH	Next-next-hop
NPL	Node Packet Loss
NSM	Network state messages
NwE	Networking Entity
OLSR	Optimised Link State Routing Proto
OM	Observation Module
OP	Optimal Path
OPEX	OPerational EXpenditure
P2P	Peer-to-Peer
PPTT	Path Predicted Transmission Time
QoS	Quality of Service
RE	Routing Entity
RTT	Round Trip Time
SDH	Synchronous Digital Hierarchy
TC45	WP4/WP5 Joint Task
TI	Time Interval
TIC	Time Interval Counters
WCETT	Weighted Cumulative Expected Transmission Time
WMN	Wireless Mesh Networks
ZRP	Zone Routing Protocol

1 Introduction

TC45 WP4/WP5 Joint Task is a cooperative platform that aims to identify synergies and establish cooperative activities between WP4 “In Network Management” (INM) and WP5 “Forwarding and Multiplexing for Generic Paths” (ForMux). TC45 addresses similarities, complementarities, interdependencies, and interactions between technical works developed in the two work packages.

WP4 defines strategies and protocols for collaborative monitoring, self-optimising, and self-healing. WP5 develops a Generic Path (GP) abstraction that can incorporate recent advances on techniques to enhance the efficiency of data transport (multi-path routing, cooperation and coding mechanisms, enhanced resource sharing mechanisms). TC45 is structured around three topics where a cross-WP coherence of technical contributions is evidenced:

- Monitoring: To use INM monitoring capabilities and external services within WP5.
- Congestion control: To investigate advanced distributed congestion control schemes.
- Resource optimisation and adaptation mechanisms: To investigate self-adaptive and resource optimisation mechanisms based on real-time monitoring.

A first deliverable, R-TC45.1 [Ferr10], identified and congregated individual contributions from WP4 and WP5 that are under this umbrella, being organised around the identified topics; in it an overview of each technical contribution is presented. Synergies with other contributions have also been shortly identified, as presented in Table 1.

TC45 has also contributed to WP5 deliverable D-5.2 [Rand09], with Section 1.4 entitled “The relation of Generic Paths and In-network Management” and Appendix F entitled “Additional Material on GP and In-network Management”, to WP4 deliverable D-4.3 [Gonz10], with Section 11.4 entitled “INM support for Routing, Forwarding and Generic Paths”, and to WP4 deliverable D-4.5 with Section 4.4 “INM Support for GPs”.

The current deliverable is the final report on the activities carried within TC45, presenting the outcome of several collaborations resulting from some of the identified synergies. The remaining document is organised as follows. Chapter 2 presents an architectural perspective of INM GP, showing how both architectures fit together, easing integration and enhancing usability. Chapter 3 addresses the resulting cooperative activities between WP4 and WP5 topics. Conclusions are drawn in Chapter 4.

Table 1. TC45 topics and identified synergies.

				Identified synergies															
				monitoring capabilities							congestion control				Self-adapt. & resource opt. mech.				
		WP	Contribution	Distributed Anomaly Detection	Wireless Network Monitoring Supporting Routing	Adaptive Monitoring	INM operations specified by the routing NwE	Synchronising routing updates and measurements	Fast failure recovery	CCFW	Network Coding for Congestion Control	Transport Paths and Throughput Optimisation	Congestion Control Based on Emergent Behaviour	Control in DTNs	Self-Adaptve Routing in wirlss multi-hop networks	Resource management in WMNs	Point-to-Point communication in wireless networks	Routing Enhanced by GPMR	Robust resource allocation
				1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Monitoring	1.	WP4	Distributed Anomaly Detection	▨					x		x			x			x	x	x
	2.	WP4	Wireless Network Monitoring Supporting Routing		▨		x	x							x	x		x	
	3.	WP4	Adaptive Monitoring			▨					x							x	
	4.	WP5	INM operations specified by the routing NwE		x	x	▨					x						x	
	5.	WP5	Synchronising routing updates and measurements	x				▨						x				x	
	6.	WP5	Fast failure recovery			x			▨			x	x					x	
	7.	WP5	CCFW			x				▨								x	
Congestion	8.	WP4	Network Coding for Congestion Control								▨							x	
	9.	WP4	Transport Paths and Throughput Optimisation		x							▨						x	
	10.	WP4	Congestion Control Based on Emergent Behaviour						x				▨						
	11.	WP5	Control in DTNs			x								▨					
Self-adaptive	12.	WP4	Self-Adaptve Routing in wirlss multi-hop networks		x										▨				
	13.	WP5	Resource management in wireless mesh networks		x										x	▨	x		
	14.	WP4/5	Point-to-Point communication in wireless networks													x	▨		
	15.	WP5	Routing Enhanced by GPMR		x	x										x		▨	
	16.	WP5	Robust resource allocation	x		x						x						x	▨

2 INM GP – An Architectural Perspective

Both INM and GP have produced architectures within the scope of their work. This looks at how best to dovetail the architectures that will ease integration and enhance usability.

2.1 INM and GP Architectures

The INM architecture is based around the coupling of the management logic with the element which needs management. This element can potentially be anything within the network that needs management, e.g., service, device, path, and link. The architecture defines two components that help to realise this goal:

1. Management Capability (MC) - a piece of management logic with an organisational (north/south bound) and collaboration (east/west bound) interface.
2. Self Managing Entity - an encapsulation of the management capabilities and the element that they manage.

The goal within INM is to position the Management Capabilities as close as is possible/appropriate to the element that is being managed. When the management is tightly coupled with the element, it is said to be inherent to that element.

The GP architecture looks to the modelling, design, and use of data flows and entities for their implementation. This architecture should be applicable to a wide range of technology levels and encompass a wide range of data processing and forwarding functions. A number of key components are defined within the architecture:

1. Entity - an entity is a process which is uniquely identifiable and has the ability to communicate.
2. Compartment - a set of entities with an associated namespace where the entities can communicate (i.e., speak the same protocol).
3. Generic Path - a basic abstraction to model data flows between entities inside a compartment.

The GP architecture attempts to develop a wide range of different path types and produce a common way of interacting with these potentially very different path types. The paradigm used to achieve this comes from the software engineering discipline, using inheritance between classes of objects. So the generic path is the base class of this class hierarchy upon which multitudes of specialised path types can be developed. Moreover, the GP architecture mandates a certain design structure, how entities relate to each within and between compartments and how basic functions like routing, forwarding, neighbourhood, name resolution, flow/congestion control should be structured.

2.2 Connection Points between Architectures

One distinctly clear connection point exists between both architectures. The Generic Path defines different path types that will need to be managed, hence, there is an element that needs management from GP and the actual management logic coming from INM. The mechanism necessary for consolidating this connection is described in the following sections.

The way in which GPs are defined is by extending the base GP class with a new GP type. Integration of INM with GP occurs when defining this GP Class Hierarchy, Figure 1. All the GP types inherit from the base GP (a very limited sample is given in Figure 1). With the addition of INM, an additional high-level classification is necessary, which differentiates GPs containing inherent management and GP types that do not contain inherent management. The GP types containing management are indicated by the green class types in Figure 1. With this structure, it allows communicating Entities to request the creation of a GP that contains its own embedded management, or one with no management dependent on the Entity's transport requirements. Note that this structure is not intended as a separate class hierarchy of management paths (on the left) that manage the unmanaged paths (on the right).

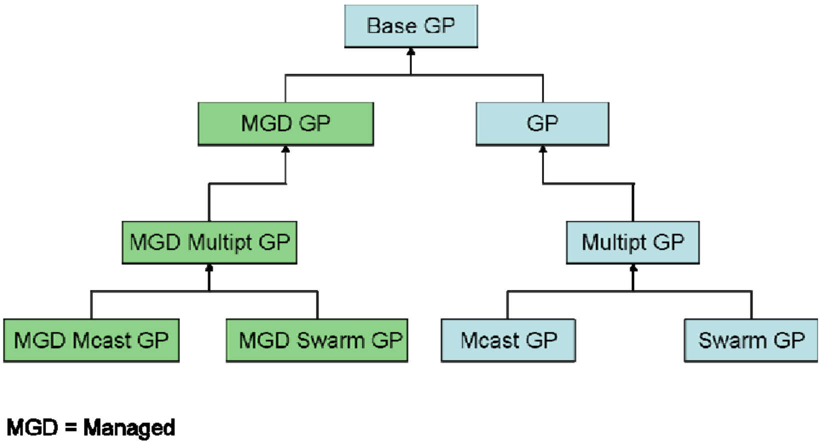


Figure 1. GP Class Hierarchy.

The Managed GP (MGD GP) contains both the GP logic and its management; the management in the form of Management Capabilities. The entities (more specifically: the endpoints with these entities) realising such a Managed GP and its specialisations are in essence Self Managing Entities, as they encapsulate both the element which needs managing and its management logic.

The number of Management Capabilities used to realise the management of the GP is entirely up to the GP designer. What does become visible is the INM Organisation and INM Collaboration interface that a Management Capability exposes. The Organisation interface is similar to a north-bound interface where configuration changes can be requested. The Entity that requested the creation of this GP will most likely be the main user of this interface. The collaboration interface is for the Management Capability to communicate with its peers, sharing and distributing management information. This information flow through this interface will use the GP itself for its transport. The components used to realise the Managed GP types are shown in Figure 2.

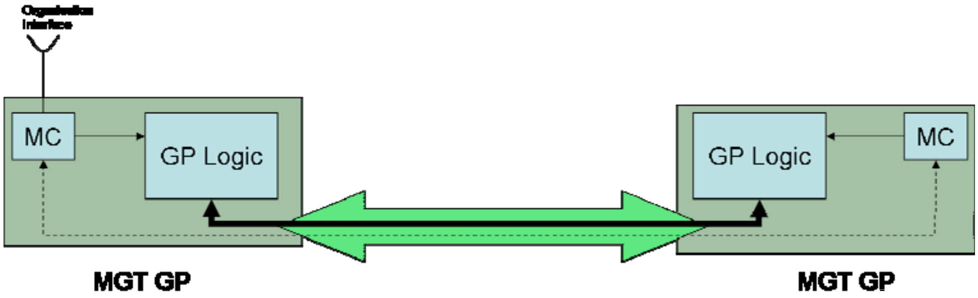


Figure 2. Inside the Managed GP.

The concept of co-design coming from INM should be a guiding principle how managed GPs should be created. Co-design states that when designing a service (in our current context this will be a new GP type), the designer should consider what management that service needs. So the management should be incorporated into the service design, not as an afterthought. As part of this, any management knobs (i.e., parameters that can be modified from outside the GP) that a GP exposes should become part of the Organisation interface of Management Capability residing inside the GP.

The internal interaction between the MC(s) and the GP logic within the GP (Figure 2) must be resolved during the GP design process.

3 Cooperative Clusters

In deliverable R-TC45.1 [Ferr10] several contributions from WP4 and WP5 were identified, within the scope of TC45 – monitoring, congestion control and resource optimisation and adaptation mechanisms – and synergies with each other. This chapter presents the resulting work of a set of defined cooperative clusters. These cooperative clusters are discussed under the light of the INM-GP architectural perspective presented in the former chapter.

3.1 INM Supported Synchronisation Mechanisms for Distributed Adaptive Routing Updates

This section is an embodiment of the distributed Multi-Path Routing (MPR) kernel – the main outcome of WP5 Task T5.3 "Realising a generic path by routing" [Rand09], combined with several INM management capabilities specified in WP4 that support synchronisation, in the whole routing control zone, of routing updates and network state information. The addressed routing protocols are distributed, link state-based and network state-sensitive. The routing updates, as well as the network state information used as input, need to be synchronised in the whole routing control zone, and there is at present no support for this in the current Internet routing protocols. The presented mechanism is referred to as: Distributed Adaptive Network SEnsitive Routing Synchronisation (DANSERS). DANSERS is based on a set of dynamically configurable timers and parameters with values specified by the routing protocol and with management provided by the corresponding Management Capability. Chapter 4 "Routing" of D-5.2 [Rand09] and Section 4.1 present the applicable GP based routing framework. DANSERS can also be integrated with standard distributed link state routing protocols, provided their hosting domain implements given time related features.

3.1.1 Qualifying Network State Changes in a Routing CT

The following working assumptions are made:

- The routing protocol: is distributed, dynamic, network state sensitive and uses network state reporting. DANSERS is presented in an intra-compartment (CT) context, that is, the routing entities (RE) coordinated by DANSERS belong to the same routing CT.
- The INM distributes network state information by means of Network State Messages (NSM) on behalf of routers. The RE specifies request parameters describing the time and spatial scope of the desired information.

- NSMs can report on links, nodes, or can contain commands and schedules to trigger an action. The values and parameters reported in a NSM are specified by the REs of the Routing CT, and are inspected and configured by means of the GP Resource Management Database.

The problem to solve, in the context of distributed routing is to avoid incoherent routing decisions between routers of the same control zone or routing CT, which are due to a different view of the network state and can cause routing loops. The problem occurs rarely when routing is based on almost static metrics such as hop number. However, it may become relevant when routing involves dynamic metrics such as current bandwidth (available, used, load, etc.) or packet loss. When dynamic metrics are involved in link state description, it is not possible for all the router databases of a routing CT to be up to date and synchronised. To ensure coherency and avoid loops, database synchronisation must be given priority over updating. So DANSERS attempts to minimise information obsolescence under the constraint that the databases are synchronised in the case when dynamic data are used in the network description. The routing protocol evolutions allowed by the GP architecture are described in Chapter 4 of D-5.2. In particular, the routing CT is the zone within the network that has a common namespace, in which all routers share the same view of the routing CT and follow the policy rules that are global to the routing CT.

This proposal addresses synchronisation of distributed adaptive routing, upon significant value changes in network state metrics, given two classes of change dynamicity:

- Sporadic changes: occur at random locations and times, after a long period of stability and are unexpected. They usually cause a spread of network changes in their time and spatial neighbourhood. Typical examples are: change in link availability, dramatic change in Current Available Link Bandwidth (CAVB), Node Packet Loss (NPL) or Link Transit Delay (LTD).
- Progressive changes: for values that are monitored at predictable times and locations. They often belong to the series following a sporadic change. Progressive change in CAVB, NPL and LTD are typical cases.

The CAVB metric is typically used by Distributed Adaptive Routing Protocols (DARP). DANSERS is applicable to DARP meeting the following assumptions, all feasible with standard features:

- Triggering actions such as route computation, updating of the Network State Data Base and Routing Information Base is supported by Timers firing every time interval (TI), time interval counters (TIC) and by dates (Dt) and Delays (Dl).

- Some input metrics to routing are measured at a predictable frequency and updated upon the policy of the routing protocol ran in the Routing CT,
- Routing Information refreshing is triggered by the firing of an Optimal Paths (OP) timer (OP-RefreshTimer), refreshing is done if OP is changed.
- Optimal Paths are computed when the OP-ComputationTimer fires.
- OP-RefreshTimer and OP-ComputationTimer fire at periodic dates that are separated by TIs, of constant length OP-HoldTime.
- The changes in the network state classified as "significant" are advertised in the relevant routing CT by the means of NSMs.
- NSMs apply to a link or a node and report their state and attribute values at a given date provided by a timestamp.

DANSERS adds the following capabilities to a DARP:

- In all routing Entities of a Routing CT, the same timers fire at the same date, which is synchronised by a standard time protocol such as (Simple) Network Time Protocol ((S)NTP) running in the Routing CT.
- OP-ComputationDate = OP-RefreshDate – OP-ComputationDuration, where OP-ComputationDuration is a parameter configured with respect to the path computation algorithm used.
- DANSERS ensures that all NSMs taken as input for routing are "old enough" for having already been flooded in the whole routing CT.
    - For this, DANSERS introduces *LSFloodingDelay* a configurable parameter representing an extended propagation delay for a NSM.
- The firing date of OP-ComputationTimer can be re-scheduled by DANSERS whenever a change in the topology occurs, that is when a "sporadic" change occurs.
    - In that case, the impacted Routing Entity (RE) floods the new OP-ComputationDate in the Routing CT and all the OP-ComputationTimer can re-synchronise.
    - For this, DANSERS introduces *NS-Delay,* a configurable parameter representing the propagation delay, in the routing CT, of sporadic changes of the network state itself.

The NSMs formed and sent by the INM are supported by the GP Resource Management DB, in particular by the GP Management Record (GPMR) associated to the RE, which provides the NSMs with the relevant updated metric values. A GPMR backing up DANSERS is attached to the Routing GPs. Routing GPs have Routing GP *Sections* GP_K, corresponding links and Routing GP Hops, corresponding to hops. A GPMR reports the i^{th} change recorded on a link GP_K by feeding NSM(GP_K, *i*). DANSERS activates and schedules the INM measurements and their processing, that will feed the relevant GPMR and initiate NSM circulation when needed.

3.1.2 Routing Entity Centred Decision Making

The DANSER protocol is distributed together with the REs. It initiates decisions that are centred on the RE: the decisions are taken by each RE, upon the NSMs that it receives. This allows fully distributed synchronisation. Each RE takes actions that comply to a policy and synchronisation global to the CT.

Like the network changes, a NSM that a RE receives can belong to two classes.

- Sporadic NSM: corresponds to a sporadic change. Therefore, it can be followed by a series of other NSM. Two consecutive ones are typically separated by a delay equal to OP-HoldTime×K_{OP}, where K_{OP} is configurable. It typically reports: a change in link availability (binary value), a dramatic change in, e.g., CAVB, NPL, LTD.
- Progressive NSM: reports values monitored at predictable times and locations. Progressive NSMs often belong to the series of NSMs following a sporadic one. CAVB is a typical case.

Figure 3 illustrates how DANSERS initiates the re-scheduling of the OP-RefreshTimer when the co-located RE has received a sporadic NSM.

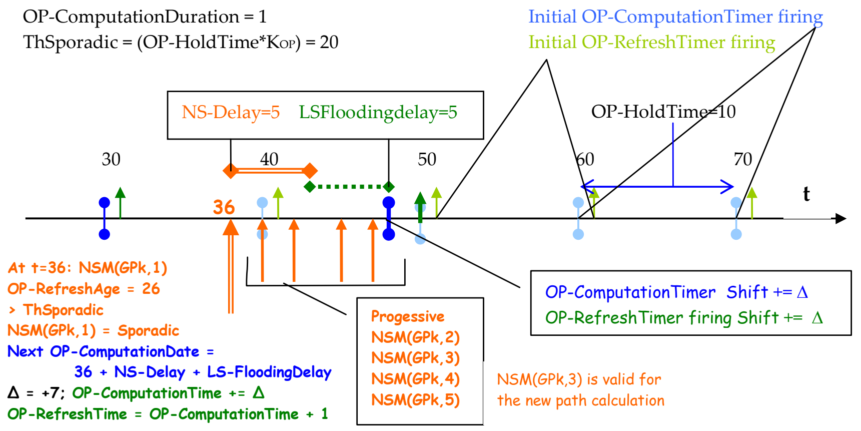

OP-ComputationDuration = 1
ThSporadic = (OP-HoldTime*KOP) = 20
Initial OP-ComputationTimer firing
Initial OP-RefreshTimer firing
NS-Delay=5
LSFloodingdelay=5
OP-HoldTime=10
30
40
50
60
70
t
36
At t=36: NSM(GPk,1)
OP-RefreshAge = 26
> ThSporadic
NSM(GPk,1) = Sporadic
Next OP-ComputationDate =
36 + NS-Delay + LS-FloodingDelay
Δ = +7; OP-ComputationTime += Δ
OP-RefreshTime = OP-ComputationTime + 1
Progessive
NSM(GPk,2)
NSM(GPk,3)
NSM(GPk,4)
NSM(GPk,5)
OP-ComputationTimer Shift += Δ
OP-RefreshTimer firing Shift += Δ
NSM(GPk,3) is valid for
the new path calculation

Figure 3. DANSERS supported synchronisation of Routing Update (at OP-RefreshDate) upon reception of a “sporadic” NSM.

In this example: OP-HoldTime = 10, $K_{OP} = 2$, the last OP-Refresh was done at *t*=20 and NSM(GP_k,1) was received at *t*=36. NSM(GPk,1) is classified as “sporadic” because when it arrives, OP-RefreshTimer has been fired more than OP-HoldTime×K seconds ago. Additional NSMs are received during NS-Delay. OP-Computation is done after additional LS-FloodingDelay. The DARP takes in the last instance of NSM(GP_k) that is at least LS-FloodingDelay time units old. In this example, it is NSM(GP_k,3). The periodic firing of OP-RefreshTimer and OP-ComputationTimer are subject to a time shift equal to Δ. Measurement timers are shifted accordingly. The shifting instruction is flooded in the Routing CT.

3.1.3 INM Support

The use of information about current network state combined with load adaptive routing provides better usage of existing network resources and fast reaction to failures. The mentioned use case shows the importance of real-time monitoring operations that are provided by INM. These operations include link/path ‘quality’ parameters and timer synchronisation of all network nodes in order to obtain synchronised update of routing tables. There are many different link metrics and algorithms for their computations. It is also worth to mention that the set of metrics is quite limited, and the same metric can be used by many different protocols. This justifies calculation of link or path metric in a manner independent on routing protocols. The network monitoring can also support routing operations by anomaly detection, fault detection (route error), and the denial of service attack awareness. All these functions are provided by INM and in general are routing protocol independent, because they in fact monitor the state of network resources. Such measurements are often made on a per link basis; however, in some cases a direct approach of end-to-end routing path quality can be applied. Per link operations are advantageous, for example in case of local re-routing or adaptive, loose source routing.

On the other hand, in order to measure path quality based on the quality of the links that compose it, additional operations are required. These operations are dependent on a parameter that is used for path metric computation. For example, in case of delay the addition operator can be used, but in case of throughput the link with minimum value determines path throughput. In general, the network process that supports routing should be able to monitor parameters of all links, and the routing tables should be used for path metrics calculation. Such cooperation can be provided in a classical style, i.e., via routing tables. INM can also provide some other information about the network state that is the result of ‘raw’ monitoring data. These operations can include time or spatial averaging of data or anomaly detection. They can be used for estimation of some long term parameters of links like averaged delay, link reliability (reputation), etc. Of course, the routing protocol should be aware of these parameters and it should be able to use them efficiently. The cooperation between the routing protocol and INM monitoring procedures has to include also the ability of the routing entity to control the monitoring procedures, in order to obtain monitoring data with appropriate accuracy. Such control should include the ability to change the sampling interval value, number of samples that are averaged, resolution of samples, spatial scope (typically which CT). Also, the way in which the raw monitoring data are further processed (statistics, transformation functions on the values, such as log, exp, absolute value etc.) should be controlled by the routing entity, because every routing protocol may have different requirements in regards to the network state monitoring. Accordingly, the requirements described in this chapter all measurement reports need time stamps.

3.1.4 Architecture

Figure 4 depicts the INM-GP architectural support of DANSERS. As specified in D-5.2 [Rand09], Section 4.1.2.3, routing specifies how data is forwarded and calculates the Route GPs between the Forwarding Entities inside a routing CT. The routing decisions (including the Route GPs, the flow distribution and their scheduling) are signalled across the Routing CT using GPs that interconnect REs and that are called *Control GPs*. As basic requirements, all REs should either be assisted by MCs capable of routing in different forms, acting as a routing broker between the GP and INM architectures, or backed up by auxiliary special Entities, which host themselves the MCs. DANSERS sticks to the latter, taking decisions such as timer synchronisation and updates based on the values retrieved from the multiple Routing GPMRs and signalled with NSMs.

The routing protocol specifies the desired measures, their shape, scheduling as well as thresholds and other values triggering routing and timer updates and then sends these values to the INM MCs that integrate them in their function calls. Convenient MCs must be embedded to the RE/auxiliary entity, according to the routing protocol ran in the CT, and a same RE may be enriched with multiple routing-related MCs. For example, mobile environments such as WMNs and VANETs should have more dynamic anomaly detection capabilities than RE at Core networks. Adding to that, the GP resource management DB is of key importance as the place for organising the parameters to be used (inspected / configured) by INM MCs, and finally updating the routing tables.

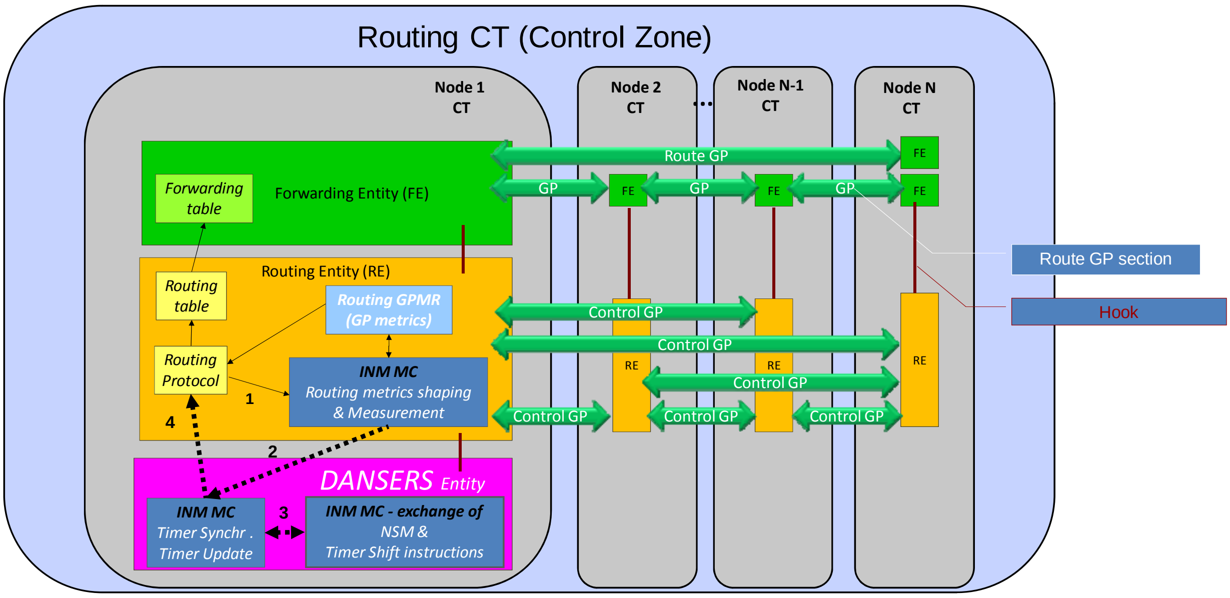


Figure 4. INM-GP architectural support of DANSERS. The thick dotted arrows depict the functional flows when the DE is used by the Routing Protocol.

3.2 INM Wireless Monitoring and Self-Adaptive Routing Supported by GPMR in Wireless Mesh Networks

This section discusses a cooperative cluster established between two contributions from WP4, one on wireless network monitoring supporting routing (presented in Section 2.3 of R-TC45.1) and another on self-adaptive routing in wireless multi-hop networks (described in Section 4.2 of R-TC45.1), and one from WP5 on routing enhanced by GPMR (detailed in Section 4.5 of R-TC45.1). It evidences how the INM and GP architecture support the transparent articulation of independent aspects, such as monitoring and routing.

3.2.1 Multiprotocol Routing Schemes and Link Metrics for WMNs

A plethora of routing protocols for Wireless Mesh Networks (WMNs) has been proposed. The existence of so many protocols is justified by different requirements and possible usage scenarios [Ram03], [Awe04], [Cou03]. Some evaluations show an advantage of a particular protocol in comparison to other protocols, but in a specific situation only. The performance of the existing WMN routing protocols is typically dependent on many factors like network density, load, nodes mobility, etc. It is worth to emphasise that in some comparative studies of WMN routing protocols several simplifications have been made (static topology, regular grid topology, and non-adaptive link bit rate) and the real time behaviour of these protocols was completely different than the simulated one. Simulations of several WMN routing protocols were presented in Annex D from Deliverable D-4.3 [Gonz10]. These simulations concerned the performance evaluation of the following routing protocols: Destination-Sequenced Distance-Vector routing (DSDV) [Per94], Ad hoc On-Demand Distance Vector (AODV) [RFC3561], and Optimised Link State Routing Protocol (OLSR) [RFC3626] based on different routing metrics in different scenarios (density, mobility, level of congestion). Here are the main conclusions derived from the simulations:

- When the nodes density is low or medium, OLSR based on the ETX metric performs well, but in dense networks AODV supported by the Less Hop Count metric provides the best performance
- When the network load is low, ETX-base OLSR performs better, whereas for a higher network loads Hop-count-based AODV has the best performance.
- For low mobility patterns, OLSR in general outperforms any metric combination with AODV - the performance gap between OLSR and AODV gets smaller if the node density grows. In case of high mobility, AODV outperforms OLSR even for low node densities; moreover, AODV based on ETX seems to be preferable for very low number of nodes, while in dense networks AODV based on a Less Hop Count performs the best.

Due to the conclusions, it seems obvious that using a single routing protocol/metric approach does not seem to be appropriate in general. Based on the presented evaluation, one proposes a combination of pro-active (OLSR), with re-active (AODV) routing schemes, with Less Hop Count and ETX metrics. The Hybrid Routing Protocol (HRP) D-4.3 [Gonz10] was created, enabling the usage of a pro-active and reactive routing protocol, and being based on the Zone Routing Protocol (ZRP) [Haa97]. Simulations have shown that in most cases HRP outperforms OLSR and AODV, showing that a combination of these approaches is indeed useful. However, even though HRP can outperform OLSR and AODV in various scenarios, a problem remains: it is required to pre-define a fixed zone - which obviously should be adapted according to the given network conditions. Moreover, one concludes that either Less Hop Count or ETX is more useful as a metric according to a given network situation. To overcome these problems, one proposes a new approach called Adaptive Hybrid Routing Protocol (A-HRP). In A-HRP each node, regardless of whether it is a sending, receiving or relaying node, continuously adapts its parameters according to the network situation it experiences. That means that data flow might get forwarded along a path based on different protocols and metrics, namely OLSR, AODV, Less Hop Count and ETX. As simulation results show, A-HRP generally performs pretty well and often outperforms all other, simulated approaches. The optimal scenario for A-HRP is a moderate scenario that does not reflect extreme cases, such as completely static or highly mobile nodes. However, if the number of nodes gets too high when using ETX as a routing metric, severe scalability problems occur. Hence, in this case a routing protocol that is solely based on the Less Hop Count routing metric should be preferred (e.g., OLSR).

The A-HRP case shows that the multiprotocol approach is a viable solution for the routing problems in WMNs. As it has also been shown, in opposition to the common opinion, there is no best metric for WMNs. In fact, the best approach is to select the metric accordingly to nodes density, load and nodes mobility. In present packet-switched networks the operations related to the path evaluation are relatively simple and the metric is typically a part of the routing protocol (only one metric is calculated, for example ETX). The most popular metrics for WMNs are: Less Hop Count, Round Trip Time (RTT), Expected Transmission Count (ETX), Expected Transmission Time (ETT), Weighted Cumulative Expected Transmission Time (WCETT) and IEEE 802.11s Airtime [Bau07]. These metrics estimate paths quality, but without complete analysis of the complex physical layer phenomena, such as the resource sharing among all nodes in the interfering range. The simulations results of the usage of modified AODV with different metrics (ETX, delay, jitter, Less Hop Count) [Kuk10] have shown no significant benefits obtained by any of the metrics against others used in quite realistic simulations (random topologies of different density, adaptive link rate). In Section 9.4.1 of D-4.3 [Gonz10], a cross-layer approach named Path Predicted Transmission Time (PPTT) [Yin06] is evaluated in terms of proper estimation of paths delay. PPTT enables the prediction of path delay based on link measurements and intrinsic mechanisms modelling using the value of the traffic which will be injected into a path. PPTT has been used with a modified version of AOMDV, the multipath routing protocol, which was called Load Adaptive MultipAth routing protocol (LAMA). In LAMA, PPTT has been used to select the best path for traffic forwarding in terms of delay; LAMA has been compared to AODV (single path with the Less Hop Count metric) [RFC3561] and Ad hoc On Demand Multipath Distance Vector (AOMDV) – multi-path with the hop-count metric [Mar01], in a static topology. It has been observed that LAMA establishes longer paths than AODV and AOMDV as expected, but these paths presented lower delay. The simulations have shown that this way of more advanced (cross-layer based) estimation of paths density provides generally better results, though when the load is low or extremely high the Less Hop Count approach performs better. These simulations have shown one more time that the existing approaches to path quality estimation are still imperfect and that the selection of the metric by the routing protocol can be dependent on some factors, what is in-line with the other simulations described earlier.

In general, the path metric calculation procedure has not only to deal with the selected quality parameter estimation, but also with the minimisation of the measurement overhead, the reduction of the traffic volume related to the dissemination of metrics and has to provide adequate accuracy and time resolution of the measurements.

3.2.2 INM and GP support

The simulations described above and novel approaches led us to the conclusion that the future WMN routing approach should be based on a multiple, cooperating routing protocols and that the network state monitoring should be independent of the routing protocols. The first observation is in-line with the Generic Path Architecture proposed by WP5 and the second observation justifies the important role of the In-Network-Management concept developed by the WP4. It is worth to notice that this separation provides the ability for optimising metrics calculation in an independent manner, and also the usage of the monitoring data for other purposes (Denial of Service (DoS) attacks avoidance, failure detection, anomaly detection). The separation of monitoring and routing is relatively simple, requiring only a small change in the existing routing protocols. The typical common parts that can be used for information exchange between the routing part and the monitoring part can be the routing table. The routing procedures have to be enriched by the routing management elements, which will be responsible for the routing protocol selection and configuration. This operation is in-line with the concept of Managed Generic Paths. A representation of this interaction between GP and INM frameworks is given in *Figure 5*. In the figure, only the relationship between transport and routing CTs is depicted. Notice that the underlying (Routing) Entities realize the Transport CT, and due to their management capabilities they end up realising the Transport CT itself.

The GP Resource Management Database (GPMR) [Rand09] is an opaque resource management database that includes relevant parameters and associated signalling. It aims to provide inherent GP support by adapting to the respective communication context, and by collecting, storing and organising in a Master Record (MR) the GP attributes and metrics (QoS parameters, hop counts, delay, load, reliability, bandwidth, etc). The GPMR allows the interaction between the GP and INM planes. INM's MCs interact with Generic Path management entities, namely CT Records (CTRs, for CT-related information) and GP Management Record (GPMRs), in order to execute necessary operations to activate management capabilities for the correct delivery of network services.

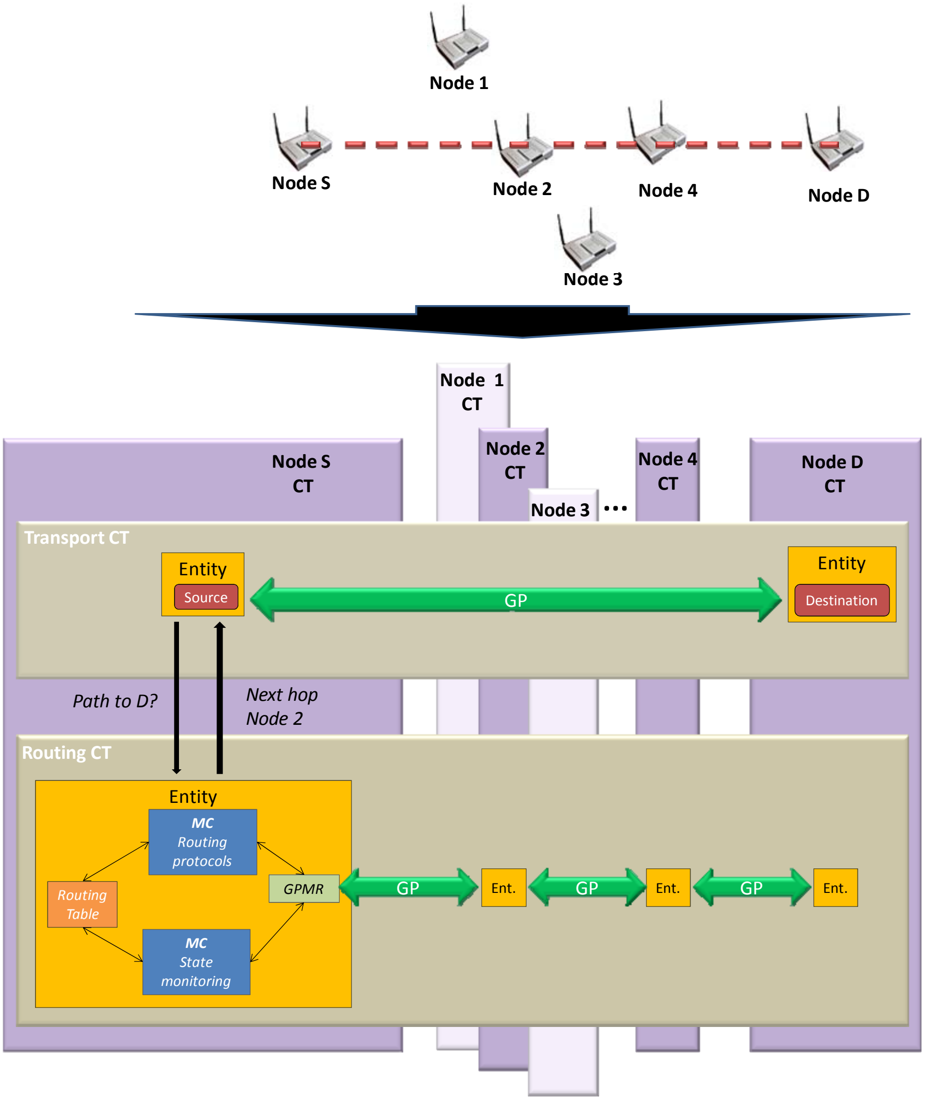


Figure 5. WMN scenario and INM GP support of routing and state monitoring capabilities.

In this specific case, the GPMR supports the routing decisions. The collected information is the one defined at the CT and by GP design: regarding WMNs, a proposal can be found in deliverable D-5.2 [Rand09], under Section 7, where the necessary classes and respective information are listed. In order to assure information reliability, MC such as monitoring and threshold detection must be provided and built-in at the RMAC and derived MGD GPs. Thus, the signalling for accessing/configuring each GP management record is defined by the GP type and its corresponding CT, providing the necessary adaptation to the context (i.e., communication level, access technology, etc). Network-wide metrics have associated thresholds which, when reached, trigger mechanisms such as changes in the refresh time of some parameters, inspection of other GPMR's (e.g., check another channel's Tx power) and CTR attributes inspection (e.g., intermodal distance). Such mechanisms have direct impact by increasing or decreasing the signalling and consequent overhead, and also depend on the considered MCs. This is an INM self-management mechanism.

The more complex management capabilities will have an underlying GP resource ontology, thus a MC representing the ontology proposed in Figure 7.7 of the previously referred Section 7 of deliverable D-5.2 [Rand09] is a must, describing the association between different QoS and other parameters (geographical position, load, channel utilisation, etc.) inspected from and triggering modifications at the resource database. Ontologies are of key importance to the previously described routing decisions, as they execute the desired inherent cross-layer design.

This work assumes basic control over the GP's universe, where GPs from different CTs and of different types exhibit different capabilities, but similar basic management capabilities. The essential MC is the creation of the GP itself, as the GP abstraction assumes some base control; capabilities specific to the GP class are, e.g., the configuration of transmitted power, RTT or throughput. As such, different management capabilities may be associated to multiple different GP types, and/or from different CTs.

The relational model of the GP resource management database is shown in *Figure 6*, which assumes Attributes associated to a specific GP type (a same Attribute may exist in distinct GP types) and also Attributes associated to a CT (this may be used for storing relevant information such as the list of stable channels, etc).

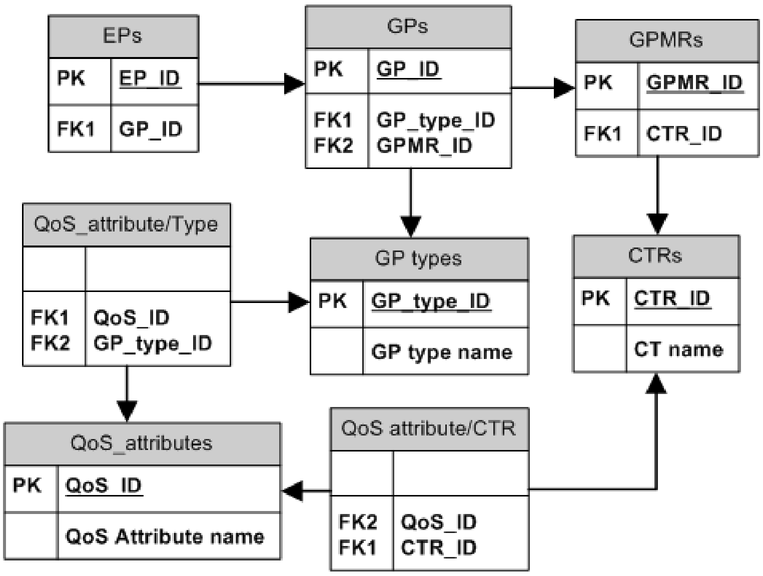


Figure 6. GP Resource Management Database Relational Model.

3.3 INM Support of the Cooperation and Coding Framework

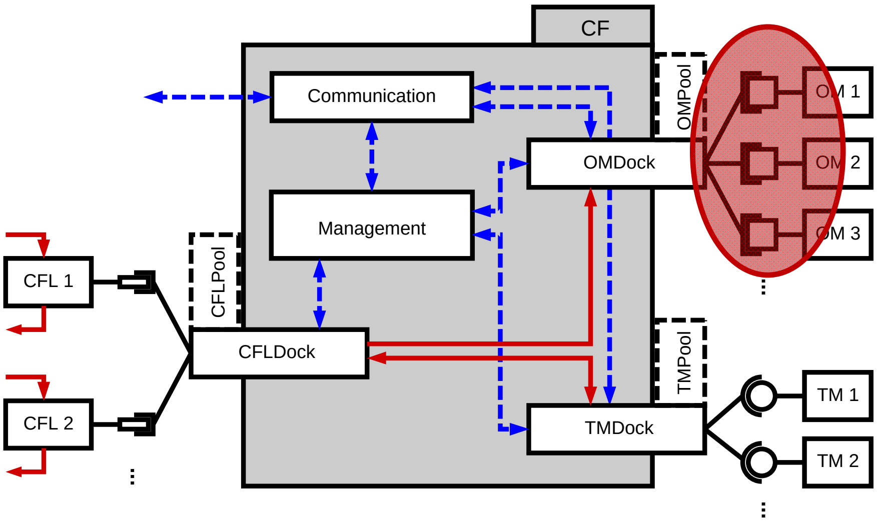
This section discusses how the Cooperation and Coding Framework (CCFW) – the main outcome of WP5 task T5.2 “Generic paths by cooperation and coding mechanisms” (detailed in Deliverable D-5.2 [Rand09]) – may use remote information available in the INM system.

3.3.1 CCFW and INM interaction

The Cooperation and Coding Framework (CCFW) [BiZH09] has been developed in the course of Task 5.2 to dynamically apply cooperation and coding techniques in scenarios where they are beneficial - not in all cases the usage of this technique provides evident profits. To decide whether a technique is currently beneficial or not, a node’s environment has to be monitored for certain properties, which are dependent on the actual cooperation or coding technique which is applied. These parameters include link usage level, available bit rate and the neighbourhood topology.

Monitoring of the environment is made in order to decide (1) whether all requirements to apply a certain technique are fulfilled and (2) whether the system can benefit from this technique. This operation is done by so called Observation Modules (OMs) of the CCFW architecture. OMs are modules that monitor some properties of the node or the node’s environment described above. The way how OMs are implemented is not defined by the architecture. Possibilities range from simple scripts that gather system information, e.g., by reading from the /proc file system in Linux, to sophisticated modules that, e.g., perform a distributed topology discovery. That is the point where INM will support the CCFW, Figure 7. As the CCFW does not care *how* an OM is implemented, it is easy to transparently query INM for information instead of collecting it by itself. The monitoring capabilities of INM can fully support CCFW operations, however, the interface between INM and CCFW must be defined. In fact, OMs require a lot of information that is already available to INM. Hence, reusing this information is much more efficient in terms of implementation and performance overhead instead of implementing own algorithms in the OMs for gathering the same information one more time.

On the other hand, properties/parameters about a certain data transfer, which depend on the used transmission technique (controlled by the CCFW), might be of interest for other purposes. For example by applying some forward error correction techniques, the robustness of the communication can be increased. This might allow to transfer data flows that have high QoS requirements over the improved by FEC link. These flows could not be transferred via this link before as the QoS requirements could not be fulfilled, in this case, the “other purpose” would be rerouting. Areas where this information is useful are resource allocation or even the CCFW itself again, e.g., to be able to activate or deactivate other techniques as well. Therefore, information available at an Entity that hosts a CCFW-controlled GP can be simply fed back to INM. This is done using the Organisation/Collaboration interface to the Management Capabilities in the appropriate Entity hosting the GP (Section 2.2).



(Solid red lines: data connections; dashed blue lines: control connections)

Figure 7. CCFW overview, where Observation Modules (OM) may use remote information available in the INM system.

3.3.2 Specific techniques

A network coding scheme is called deterministic if coding functions installed at a particular node for a particular set of flows always produce the same output packet for a given set of input packets. This is in contrast to random network coding, where coding functions have random components. The advantage of deterministic operation is that whenever the coding functions are in place and the decoding node knows how to decode, no further communication overhead is required. The drawback, however, is that a significant amount of coordination is required to establish a coding scheme, calculate the codes, and inform all participating nodes. In random mode, the coordination overhead is lower; however, each packet’s header must carry additional information about applied coding operations, so that the decoding node can revert them. To enable deterministic or random network coding, information from local as well as remote OMs is required. This information includes next and/or previous hops of particular data flows, available resources (CPU time and buffering capabilities) at nodes that are passed by these data flows, and topology information about possible paths that could be used by generated coded flows. The algorithms for gathering this information should not be implemented directly within an OM. They should be implemented within INM to guarantee efficient information dissemination and to increase reusability.

Forward error correction and fountain codes are techniques that add redundancy to a data flow to protect it against packet erasure. This is especially useful in wireless scenarios where such packet erasure can occur frequently. To enable such techniques, basically two kind of information is required. First, one has the current packet error rate; it is required to adjust the coding parameters to exactly compensate the occurring packet erasures without creating unnecessary overhead. Second, congestion must be detected on the path to not mistakenly send additional redundancy in case of buffer overflows, which would worsen the situation. For both operations the monitoring properties of INM can and should be used.

3.4 INM Cross-Layer QoS Used in Network Coding-Based GP

This section provides results from the cooperative work between WP4's adaptive monitoring (given in Section 2.4 of R-TC45.1) and anomaly detection (given in Section 2.2 of R-TC45.1) contributions with the combined work from WP4 and WP5 on network coding for congestion control (presented in Section 3.2 of R-TC45.1). The aim is the to provide solutions for controlling the congestion in fixed or low mobility networks employing network coding techniques operating based on the network state information, including anomaly related information, provided by the INM.

3.4.1 INM CLQ and GP Collaboration

One defines parameters of a node by monitoring its available resources (bottom-up approach), and imposing parameters to the substrate network (top-down approach) can be considered as well. A tight collaboration between INM CLQ and GP is essential, in order to monitor, allocate, or guarantee the usage of resources. Despite the GP inherent basic management logic, the INM CLQ supervises the MGD GP, informing it about the global situation in the network.

As it is depicted in Figure 8, the CLQ management capability exchange information with both MGD GPs (through collaboration interface) and GPs that does not contain management capabilities (through organisational interface). The MGD GP 1 is actually a GP that is specialised in congestion control by employing Network Coding (NC) techniques in the network. The coding process is done by the GP Logic (NC-based GP Logic), whilst the activation/deactivation or adaptation of the coding type utilised is done by the MC block.

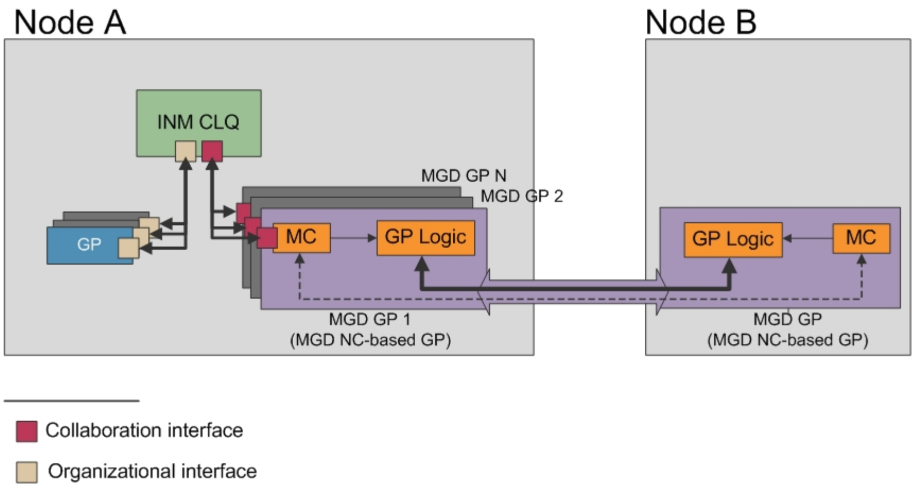


Figure 8. Exchange of information with two MGD GPs by the CLQ management capability.

3.4.2 Algorithms for Cross-Layer QoS for

The following example algorithms are employed by the Cross-Layer QoS for situation awareness and self-adaptation: Anomaly Detection (AD) [StGi09], Adaptive Monitoring (AM) [ElRa08] and "Not All aT Once!" (NATO!) [CoLa09].

As a service, the measurements could be done by request. Thus the AD module sends a request to obtain current link latency/round-trip-time via one hop-to-hop transaction. The MC receives the request and immediately sends the probe. If a probe reply is received, the measured latency is sent back to the AD-module. The anomaly-detection model autonomously adapts to the new observation. In case a probe reply is not received, the CLQ does nothing, but the failed probe response will be accounted for in the AD-module.

A strategic node that implement the CLQ needs to get metrics from all network elements within its management domain (i.e., available transfer rate, one-way delay, Bit Error Rate (BER), etc.). This could be provided by the CLQ being present in each node. However, this approach is not always realistic, because some of the network elements are too simple to integrate sophisticated mechanisms. It may lead to either too much traffic for information exchanges, or to limitations of scalability. Two INM concepts can be deployed in order to address this problem: NATO! and Adaptive Monitoring (AM).

NATO! is a probability scheme for estimating the size of a group of nodes affected by the same event without explicit notification from each node, thereby avoiding feedback implosion. Using the NATO! algorithm, the CLQ strategic node query all the devices within its domain with a specific request (i.e., which of them are able to accept flows with a given transfer rate and/or one-way delay). NATO! ensures that only a small sample of those devices that can accommodate the request are responding, while the strategic node can accurately estimate the total number of nodes. If statistics show that the number of nodes fulfilling the requirements is large enough, CLQ could involve QoS-aware routing only. If the statistics are bad, then network coding is an alternative mechanism. Thus, NATO! helps INM to dynamically decide whether QoS-aware routing or network coding-based techniques should be deployed [PoKi09], [RuBa10].

Alternatively, the AM scheme can effectively collect the timely metrics needed for the CLQ operation. The AM scheme implements a distributed algorithm that dynamically allocates the various monitoring resources that are available within devices in the managed domain, based on current needs (threats). Thus, AM is able to effectively provide the needed metrics for the CLQ to determine the necessity for network coding-based techniques.

As described in [Go10], INM CLQ has 2 types of netlets (protocol stacks), as defined in D-2.3.1 [ArZi10]. A Measurement Netlet has actually no protocols included for communicating directly with the hardware. This is required by three major beneficiaries: a) the bottom-up approach while performing periodical measurements on top of the MAC Sub-Layer; b) the top-down approach, in case of emergencies, performs global resource allocation by replacing the GP management that failed; c) a hop-by-hop transport between nodes. The Publishing Netlet may include legacy/ future protocols and it is needed for exchange of information with other MCs located in different nodes. On the other hand, for NATO! algorithm, the server interacts with clients through a NATO! Netlet, as depicted in Figure 9. The server is co-located with CLQ in the strategic node, i.e., the node implementing major management capabilities such as discovery, event handling, security, etc. The NATO! aware domain could interact with non-NATO! aware domain through Publishing Netlet and collaboration interface.

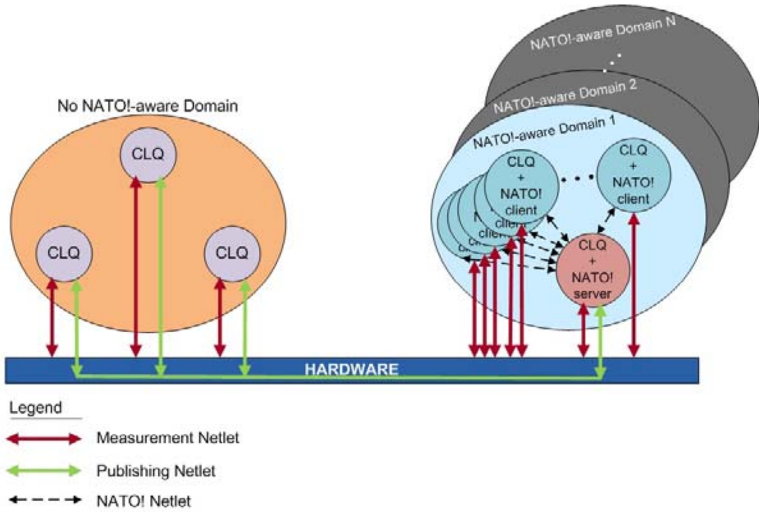


Figure 9. Interaction of the server with clients through a NATO! Netlet.

3.5 Adaptive Resource Optimisation

This section is the outcome of a cooperative cluster between WP4 technical work on anomaly detection (detailed in Section 2.2 of R-TC45.1) and WP5's work on robust resource allocation (described in Section 4.6 of R-TC45.1). One combines parts of the anomaly detection mechanism with resource optimisation techniques. Specifically, the adaptive modelling approach applied in the case of anomaly detection can also be used in a similar way for the case of resource optimisation, in order to trigger re-optimisation in a timely manner. The aim is to obtain a resource optimisation algorithm that is adaptive to long-term changes in the network.

3.5.1 Architectural Mapping between INM and GP

Since only a part of the learning mechanism from the distributed anomaly detection is used for re-optimisation purposes, there is no direct mapping between the two methods in a joint INM GP architecture.

However, let us assume that an adaptive monitoring module exists and that it is designed to measure and statistically model the traffic situation between nodes (not necessarily adjacent) as described. Then, the optimisation module and the monitoring module would be related in two separate management capabilities, residing in dedicated nodes in different regions in the network, Figure 10. For the purpose of triggering re-optimisation properly, the monitoring module would upon convergence of recently observed traffic behaviour, provide relevant data (via its collaboration interface) to the optimisation module, which manages a GP based on the outcome.

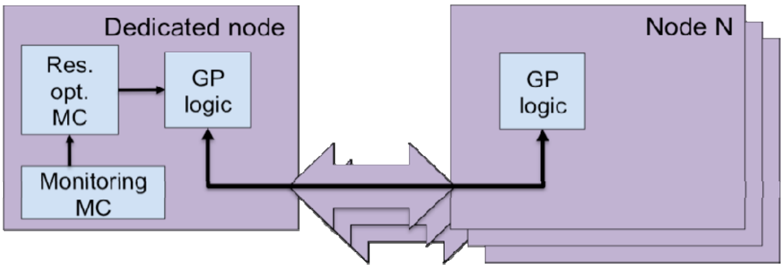


Figure 10. Architecture mapping between a resource optimisation module and a monitoring module.

3.5.2 Technical Specification

To configure the routing, some knowledge of traffic demands in the network is needed, particularly if the routing profile is set to meet service level agreements made with customers. The traditional way to do this is to calculate a routing setting for an anticipated traffic situation obtained, e.g., from traffic measurements. When the traffic situation deviates from the anticipated traffic situation, performance degrades and the routing setting needs to be recalculated. To avoid frequent recalculations, one has developed in WP5 methods to calculate routing settings that are able to absorb fluctuations in the traffic demands.

The calculation (or optimisation) of the routing parameters is performed for a set of different traffic situations describing uncertainty or variability of the traffic demands. To formulate a tractable optimisation problem the set of traffic situations need to be captured by a mathematical model. The focus in WP5 has been on models where the traffic variability is described by an ellipse. This kind of model arises when traffic demands are assumed to follow a Gaussian distribution and one wishes to optimise the routing setting for the most likely outcomes of the assumed stochastic model. Thus, the routing setting is determined not only for the mean value of the assumed Gaussian distribution, but also for wide variety of samples from the distribution. The gain is that the routing setting does not need to be calculated for new samples even though they deviate from the mean value of the distribution. Hence, one needs to estimate the parameters of the underlying Gaussian distribution, assuming that one is able to observe the end-to-end traffic demands.

Even though the routing setting does not need to be recalculated for new traffic situations, they can be regarded to be samples from the assumed Gaussian model. However, if the parameters of the model change, one needs to trigger a recalculation of the routing setting for a new set of parameters of the distribution. For this purpose, one adopts a similar statistical learning model used for distributed anomaly detection (based on measurements of link latencies and packet loss) developed in WP4. This will augment the robust routing approach with adaptive functionality. The routing setting will be able to autonomously adapt to changing conditions in the network while still benefit from robust routing.

The outcome of the effort in TC45 is the model architecture as well as the decision logic in the first module where deviations of the assumed Gaussian model are detected. Recalculation of the routing setting is only triggered when parameters of the Gaussian distribution change instead of for each sample from the distribution leading to a more stable routing setting. To perform this, one needs to perform the following steps:

- Detect a change in the assumed distribution.

- Obtain a new set of parameters of the Gaussian distribution.
- Determine if the change in traffic situation requires a new setting of routing parameters.
- Recalculate the routing setting and disseminate the result in the network.

These actions are performed in different modules and our focus in TC45 is on the first step above, detection of change in the assumed distribution. To this end, one models traffic data using partially overlapping statistical distributions as indicated in Figure 11. The long term development in traffic behaviour is accounted for by usage of the overlapping statistical distributions. The observed data is summarised partially in the previous model and used as prior input to the next model; old data successively decay while sensitivity to new data points is reduced. The average of the T first samples of the observed data in the current model is used as prior input to a new model. The impact of earlier samples is gradually forgotten as new samples become available. The sensitivity to new data and the decay rate can be tuned by adjusting the model size N and decay rate T.

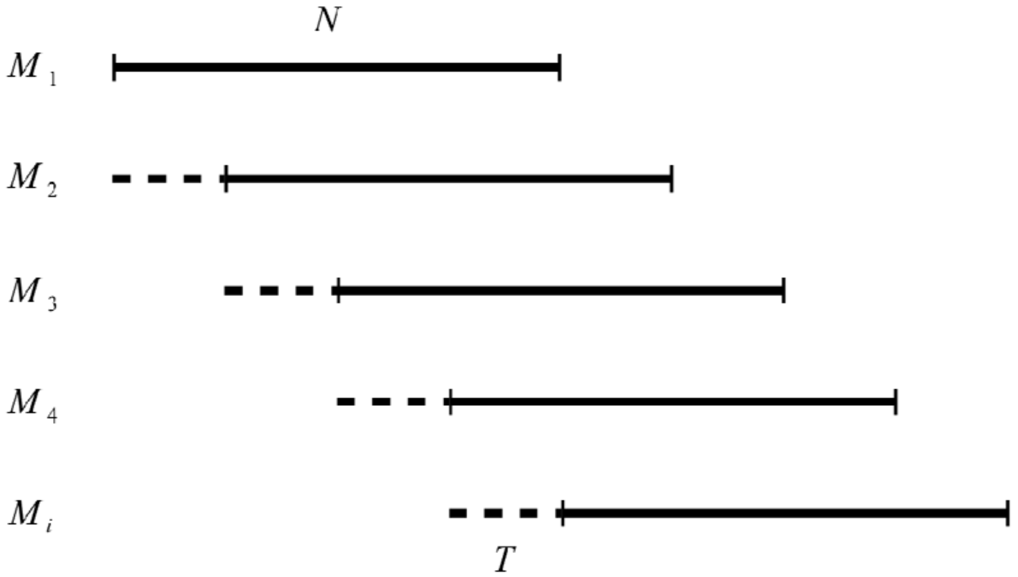


Figure 11. Traffic parameter estimation using overlapping models M with model size N and decay factor T.

In order to detect a change in the model distribution, the symmetric Kullback-Leibler divergence is calculated between the current and previous model estimates. When the Kullback-Leibler metric is above a certain divergence threshold, a triggering signal is set to prepare for a re-optimisation. When the models have converged to the new ‘regime’ (i.e., reaches below a certain convergence threshold), a new routing setting is calculated in the optimisation module.

3.6 INM-enabled Multicast GP for Fast Failure Recovery

This section is a cooperative cluster that combines a contribution from WP5 on multicast fast failure recovery (presented in Section 4.6 of R-TC45.1) and another from WP4 on anomaly detection (introduced in Section 2.2 of R-TC45.1). The aim is to improve the re-routing process and utilisation of backup paths in multicast-trees, based on the availability information provided by the distributed anomaly detection mechanism.

3.6.1 Mapping with the INM-GP Architecture

This piece of work is an instantiation of the generic INM-GP architecture for supporting graceful fast reroute functions that aims to protect GP-based multicast routing infrastructure. In WP5, a GP aware multicast fast reroute technique was designed based on the NotVia-like tunnels. It should be noted that Fast ReRoute (FRR) protection approach should be specific to different types of failures in multicast routing. For instance, in case a (in tree) link failure, the original multicast tree can be healed by reconnecting to the Next-Hop (NH) neighbouring node via an alternate path that does not involve the failed link. However, in case of a node failure, the situation will become much more complicated. This is because the failure of an intermediate node will result in the breaking of multiple downstream tree branches that are constructed dynamically upon the group join requests. As such, each potential repairing node cannot pre-establish alternative paths a priori before they have the knowledge on the physical location of remote receivers (i.e., have received any group joins). The solution is to pre-provision node-protection GP that terminate at Next-next-hop (NNH), and upon receiving individual group join requests, these pre-provisioned node-protection GPs are dynamically bound onto the multicast delivery tree.

One has proposed to establish dedicated backup NotVia-like Multicast GPs for link and node failures respectively. However, a key issue is how the potential repair router attached to the failure knows the exact failure pattern in order to activate the appropriate multicast protection GP. Towards this end, that AD function is applied as a complementary technique that is responsible for reporting the actual pattern of failure (e.g., link or node failure). As shown in Figure 12, a GP endpoint pre-provisions two types of multicast protection GP: link protection path, which terminates at NH, and node protection path, which terminates at NNH. When the multicast delivery service is activated, the AD function located in each endpoint keeps monitoring the working condition of the locally attached elements. In case a failure is detected, the AD component is responsible for analysing its specific pattern, and localise it to a link failure or a node failure. The result is then provided as input to the multicast GP logic which is responsible for activating the corresponding protection GPs for failure recovery.

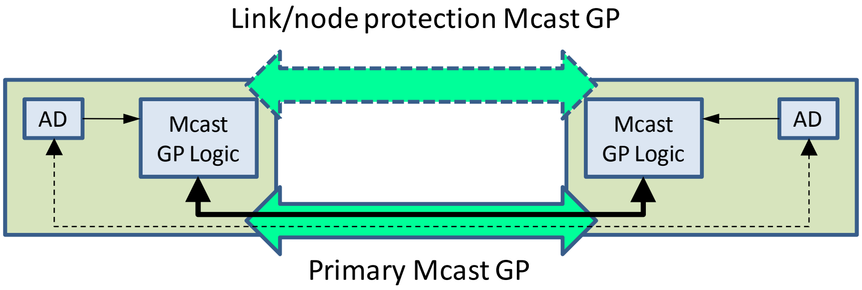


Figure 12. INM-enabled Mcast GP for fast failure recovery

3.6.2 Detailed Specifications

One now illustrates in details how NotVia backup paths are constructed and activated (with the necessary input from AD) according to different failure patterns.

As mentioned previously, to protect multicast delivery trees against link failures is relatively easy. A static NotVia backup path is constructed, which originates from each repairing router and terminates at the NH node of the router. Such a backup path effectively is the second shortest path from the repairing router to its NH which excludes the link to be protected. Once the link is detected to have failed, the local repairing router (head node of the failed link) immediately encapsulates the affected multicast traffic and diverts it onto the NotVia backup path in order to reach its NH that is still alive. In this case, all the downstream group members (i.e. receivers as leaf nodes) can be reconnected by one single backup path. Effectively, such a single backup path can be used for recovering both unicast and multicast traffic at the same time upon single link failures.

Compared to the link failure scenario, to protect multicast infrastructure upon node failures is a much more complex task. Specifically:

1. Lack of knowledge on “re-grafting” points in case of node failures – according to the standard NotVia scheme for unicast FRR, NNH is used as the tunnel endpoint for automatically bypassing the failed next-hop node towards the protected destination. As far as multicast routing is concerned, the location of in-tree NNHs is not automatically known by the repairing router. As a result, the repairing router is not aware of the Not-via tunnel endpoints for local protections on each group upon a node failure.
2. Point-to-multipoint requirement – existing NotVia tunnels are all point-to-point for unicast routing protections. How these tunnels can be used for point-to-multipoint

multicast routing needs to be considered. For instance, is it necessary to extend standard NotVia tunnels to be point-to-multipoint in order to support multicast FRR?

To tackle the above issues, our proposed technique for node protection can be summarised as follows. NotVia backup paths, which terminate at the NNH of each repairing router, are pre-provisioned before hand, and these protection paths can be directly used for unicast routing. Such a backup path effectively is the second shortest path from the repairing router to its NNH, which excludes the NH neighbouring node to be protected. In order to support multicast protection, individual NotVia paths need to be dynamically bound onto individual multicast trees as they are constructed by specific incoming multicast group member joins. The purpose is for each in-tree repairing router to know its own in-tree NNHs, leading towards remote downstream group members in case its own NH neighbouring node is unavailable.

The overall failure recovery operation in multicast routing is illustrated in Figure 13. The AD function is embedded inside each repairing router and keeps monitoring the status of local nodes and links and is responsible for reasoning specific types of network failures in order to provide necessary feedback to the fast recovery engine. If any node detects a communication failure on a connection, it requests assistance from the neighbours of the suspected node in order to localise the failure to a node or link:

- In case the communication fails between the suspected node and all of its neighbours, the failure is classified as a node fault. The fault is locally reported to all neighbours of the faulty node.
- In case the communication fails between the suspected nodes and at least one of its neighbours successfully can communicate to the suspected node, the failure is classified as a link fault. The failure is locally reported to both nodes on the faulty link.

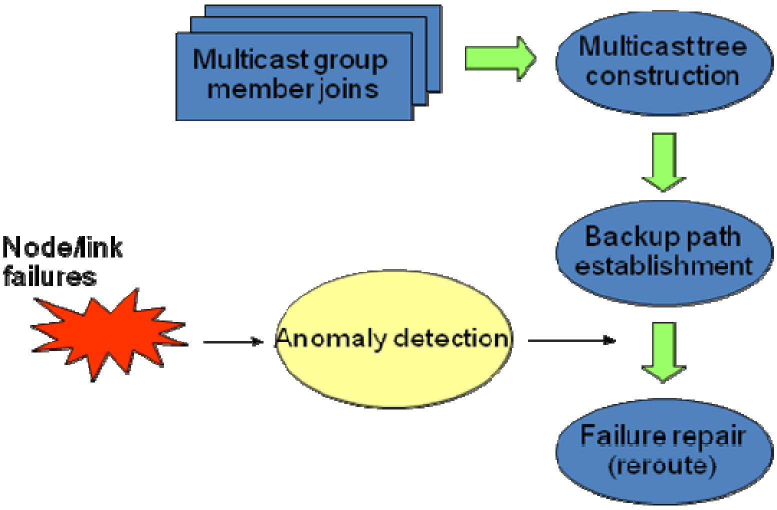

Multicast group member joins
Multicast tree construction
Backup path establishment
Node/link failures
Anomaly detection
Failure repair (reroute)

Figure 13. Node protection operations using AD-based multicast failure recovery.

According to the feedback from the neighbours of the suspected node, the repairing router will be responsible for activating the corresponding NotVia backup path to recover the multicast delivery tree.

3.7 Virtual Cognitive Radio Switch: INM & GP Joint Application in Real Network Environments

This section presents results from the cooperative cluster between two technical contributions from WP4, one on point-to-point radio communication in wireless networks (described in Section 4.4 of R-TC45.1) and another on anomaly detection, from WP4 (presented in Section 2.2 of R-TC45.1), with a contribution on radio resource management for WMNs, from WP5 (introduced in Section 4.3 of R-TC45.1). This contribution presents a solution, the virtual cognitive radio switch, combining several strategies from both WP4 and WP5. It uses already available hardware infrastructures to obtain the best possible result for all network participants, while using a modern, decentralised approach to tackle network and traffic issues.

3.7.1 Introduction

Bigger growing networks pose a huge load on nodes. Our goal is to use as much as possible out of today's existing hardware infrastructure in order to allow a better QoS for all network partners. Today's picture is fair simple: a client receives the requested data from a server. The QoS depends on the quality of the connection and the network load. If the data is distributed across more nodes and through one ore more clients (devices connected to the network), even in a busy network environment or in case of a poor network connection the speed at which the data is transferred reaches fair values for all users/clients.

A mobile network, consisting of a server, *n* nodes and *x* clients is considered. Each user (client) has an ongoing data transfer with a node or the server itself. One considers a user that moves inside the network. In a classic environment, the user receives the data from one single point that is its access point. In order to obtain a better connection, one proposes using a different model, which is somehow similar to the Peer-to-Peer (P2P) system. In case the user requests data from a node or a server, the server asks the user if it wants to use helpers. A helper is a normal user that "reroutes" data from the server to the client. Helpers are the actual users that, as helpers, do not have any knowledge about the actual data they process. The resulting environment is an ad-hoc network in which nodes and clients (the client itself disconnects from the node and relies only on the helpers for data exchange) communicate with each other wirelessly using a virtual radio cognitive switch, as depicted in Figure 14. In this environment the clients and nodes, all include an anomaly detection module (for situation awareness), a resource management system (to provide a reliable and fault tolerant service) and a configuration/prediction module (that ensures a smooth running system even in a busy network environment), as represented in Figure 15.

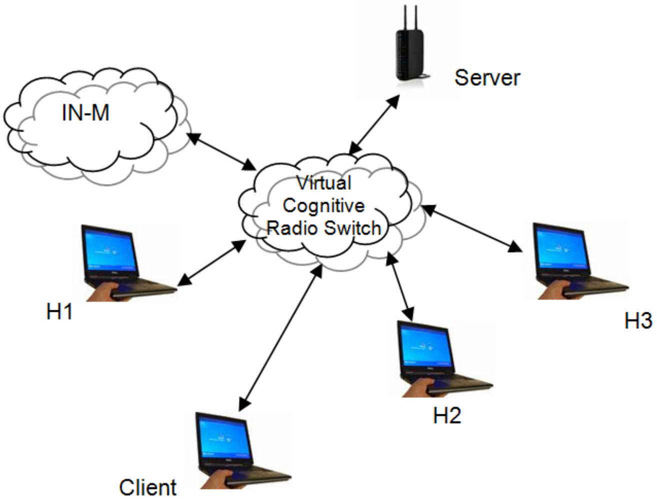


Figure 14. Virtual cognitive radio switch.

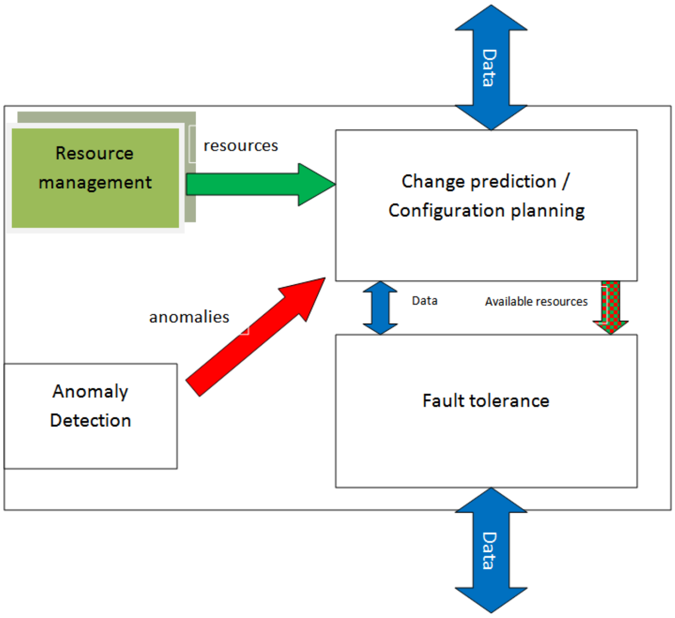


Figure 15. All nodes include an anomaly detection and situation awareness module as well as a resource management system.

3.7.2 Anomaly Detection for Situation Awareness

Anomaly detection is a fully distributed algorithm, capable of detecting communication faults and shifts in the expected link delay. The method is based on overlapping statistical models of observed link delays, which allows for adaptation to network variations as well as detection of short-term and long-term latency shifts.

Observations of the link delay are obtained via probing performed in each node. All nodes probe adjacent nodes (collecting observations of packet loss and link delay or RTT), statistically modelling each connection. The probing interval is autonomously adjusted on individual links, based on the expected link delay for the specific connection.

Communication faults and latency shifts are detected based on the statistical models. Detected faults are localised to a certain node or link, possibly via node collaboration if the situation requires it. The concluded availability status about the faulty node or link is reported to neighbouring nodes and other relevant recipients.

3.7.3 Resource Management System

A distributed strategy for management of radio resources of self-organised multi-radio WMNs is proposed, described in detail in Deliverable D-5.2 [Rand09]. Nodes use a channel management strategy that guarantees connectivity with mesh neighbours. Each node announces a listening channel where it is always listening with one of the radio interfaces. A mesh node also has a switchable radio interface that can be tuned to the listening channel of any neighbour, in order to forward data.

The mesh communication infrastructure is built and optimised using a mechanism of announcement of resources of each node (periodically sent hello packets) announcing the node’s resources and state information (forwarded traffic load, distance to gateway, number of neighbours forwarding traffic to it, average delay, etc). This infrastructure incorporates extra QoS statistics that are used to turn neighbour mesh nodes into helpers. A “helpers” mechanism is activated whenever a more intense traffic flow (anomaly detection reports slow links for the given transfers) is detected by a mesh node, providing a local mechanism to distribute a traffic flow whose intensity is not bearable by the mesh node alone. Using neighbours as helpers, and by diverging the traffic that is only to be forwarded by this “mesh client” flow into them, could locally help for a more efficient flow of information.

Mesh networks deal with the forwarding of traffic flows aggregated from/to Access Points. Being this, it is not easy to identify a chunk of information to be received (as is done for the case of a single user, where he/she requests the download of a certain file).

3.7.4 Change prediction/Configuration Planning

The autonomous system that triggers the use of helpers is configured manually. The configuration planning allows for manual configuration via a configuration component, supported by a prediction component for the purpose of preventing configuration mistakes that subsequently may lead to misconfiguration based faults in the network. Configuration planning is done by an administrator. Changes that are set up run through the configuration component, which in turn runs a simulation of the proposed change via the prediction module. The prediction module runs based on a Markov chain, where a future state of the system depends only on the current state, as described in Deliverable D-4.4 [Nunz10].

For each system variable a transition matrix is computed. The transition matrix reflects any changes the system variable can undergo. Each “live” node computes the probabilities using data received from its neighbourhood; the transition probabilities are local. This provides an understanding of local conditions as having a “global view” does not always reflect all possible issues. A global view of the system is only created by the system administrator who can access all nodes. This reduces the computing effort in each node and also minimises the security risks.

3.7.5 QoS Assurance

Each helper has its own area of effectiveness in which it can ensure the best speed and reliability for a given number of users (mobile entities). This area is computed based on routed data packets (e.g., VoIP or video packets are given higher priority then others), data volume, number of users, the necessity of rerouting data packets from other helpers and the need of receiving data packets from the server (in case the server is the only entity that has the requested packets). Each helper computes the QoS for its given workload. Using this information it draws a „tile”- an area in which it can assure a good QoS for a certain number of clients.

Based on the “tiles” drawn by the helpers a virtual network map is created. The data chunks are collected in a virtual storage space for the mobile device. By measuring the client’s needs and movement, each “tile” should be able to describe the path which the client follows. The client has the possibility to accept or to reject a certain helper.

Based on simulation results both helpers and clients will be aware of the network transferred data and its effects. If the effect will be one of overload or overstress on a helper or network node then data packets get rerouted.

Generalised processor sharing algorithms are used as a measurement unit for the network state. As the different clients inside the network request a multitude of data packets, different performance levels are required. Using a round robin like method helpers can be shared between more clients without QoS decreases.

3.7.6 Approach to Improved Fault Tolerance

A combination of the described methods is proposed for the purpose of achieving a point-to-point communication service with improved fault tolerance and reliability. One assumes that the system is up and running and has an initial set of running rules. All network components run an instance of the above mentioned modules. The communication between network components is unicast and any detected issues are to be shared among neighbouring entities. Also neighbouring components are aware of their vicinity for data sharing and anomaly detection. An overseeing entity (administrator, server) is used for collective data gathering and availability information. Multi radio routers are considered available for all users. Link/delays (RTT) are considered to be statistical modelled.

3.7.7 Specification

One considers a normal data transaction in which all users are connected to a node within the given network, and a user asks for a data download. The server asks the user if it wants to use helpers. In case of a positive response, the server asks the surrounding resources to become helpers. The wireless resource management system uses a channel allocation algorithm, which also allows for the radio connection to take place, to collect information of the neighbours (connection channel, load throughput, distance to the gateway and delay). This creates a local, distributed picture, which is used to selecting the best resources for the given task.

By this the requesting user becomes a client and the surrounding ones become helpers, as depicted in Figure 13. The requested download is split into chunks which in turn consist of small pieces of data. The client receives a list of helpers, available chunks and QoS information from the server. Each helper receives a chunk based on the QoS that it can provide. If a multi-radio node is used, connections can take place at any moment the client chooses.

Based on the received list the client connects directly (without communication via nodes or server) to some of the helpers. The helpers can be connected to any other node than the one of the client. As the communication is established, the client and the helpers monitor neighbouring nodes, in order to detect faults and latency shifts via the anomaly detection module. Helpers and clients are aware of their neighbourhood in order to be able to reroute pending chunks or lost ones (chunks are rerouted based on information provided by the resource management system), and to perform collaborative fault localisation. Changes in the network topology trigger the generation of a new list of which the client and helpers become aware of immediately.

Using the radio channel management strategy provided by the wireless resource management system the client is always reachable on a channel so helpers can send him the data chunks. This method allows for a very fast data download even in a busy network environment (there is no interaction from the nodes or the server in this transfers).

The channel management strategy for wireless mesh networks discovers and catalogues the resources, by type and speed class, for each node and its neighbours. Each node announces its resources periodically such that neighbourhood info is stored locally, keeping it updated. Resources are considered to be any infrastructure component willing and able to help in the given scenario. Because it is a wireless network, a history is saved such that mobile users can be evaluated also based on their previous behaviour. Users that “come and go” are to be marked as unreliable even if their resources are suited for the given model. Each entity inside the network will be labelled. Due to this approach, the time devoted to some operations is reduced. For example, mobile phones that have sparse resources, are not scanned by the resource management system. Extremely mobile users are served first and their resources are not used within the network.

The resource information contains the quantity, quality and security information of the users and nodes and INM contained settings and restrictions that apply the system taken into consideration.

Based on the exchange of anomaly detection control messages and availability status between participating network equipment, necessary actions are taken to reroute data chunks and switch between available helpers upon detection and localisation of failures and latency shifts. For example, in case the client detects a communication failure to one of its helpers, it asks the neighbours of the suspected helper if they experience the same problem. In case the communication failure is proved to be caused by a link fault, the helper is informed about the link fault by the client, subsequently rerouting its chunks to another helper. In case the communication failure is proved to be caused by a node fault in the helper, the client updates its list of helpers and informs adjacent network equipment (and other relevant recipients) about the faulty helper.

In all cases resources are detected and "proposed" to the system by the resource management system. Based on the outcome of the anomaly detection, resources are rearranged and provided to the client.

3.7.8 Scenario

Two types of users are taken into account. The first one is a cell phone. This is a highly mobile user, travelling around a lot and using some network resources for its data transfers. If the cell phone only needs little transfer the classic approach (client-server) is used. In case of data streaming the cell phone becomes the client and the surrounding resources become helpers. The phone starts, as above stated, the data transfer but due to the fact that it is moving around helpers need to be exchanged very fast. As the client moves some links to some helpers will loose their strengths. The anomaly detection will report this and available resources are offered to the client. As the network gathers some history regarding the current user it will realise that it has to allocate more resources to it so chunks will not go lost. This will stress out the network but the configuration management reports the issue and an outside administrator reconfigures the network. After the settings are checked and applied the behaviour of fast moving clients becomes known to the system. The resource management system allocates more resources and the fault tolerant algorithm redraws the tiles based on the new setup (tiles are smaller in order to provide a better QoS for the user). After the client finishes the streaming session the network returns to a classic state. The client reconnects to a node and disconnects from the helpers; resources become available again.

In case of a laptop the mobility is not such a big problem any more. Also a laptop has more resources to offer and might also be usable as a helper. This gets noticed by the resource management system which rebuilds the list of available resources. In case the laptop asks for a data download the same approach as above is taken only fewer resources are allocated. After the data download is finished the laptop reconnects to the node but also offers its resources to the system. If the node gets notified that a highly mobile user is in need of resources, the laptop quickly is reconfigured as a helper and routes data packets to it.

3.8 Traffic Management Support by INM: Demands, Progress & Business Value

This section is based on INM traffic engineering applications to enable optimised resource utilisation. The contribution is related to work on the WP4business value theme that shows the impact on CAPEX and OPEX due to savings in routing equipment and energy consumption. The focus is on enabling functions through several INM contributions for enhanced monitoring and failure reaction together with an optimised traffic path design for load balancing to be realised through WP5 GP concepts.

3.8.1 Traffic engineering in IP and MPLS Networks

Pure IP networks do not give much support for traffic engineering. Measurement of the link loads over 15 minutes intervals have been the only standard information on IP backbone traffic prior to the millennium, such that special measurement equipment or network tomography approaches were required to estimate the edge-to-edge traffic paths and volumes. The introduction of Multiprotocol Label Switching (MPLS) as a sub layer below IP backbones substantially improved the situation towards a flexible handling of traffic paths including standardised measurement of traffic matrices and pre-configuration of backup paths. Since 2001 a number of large network providers have underlain their IP backbones with MPLS and are still expanding seamless MPLS into access aggregation areas. MPLS networks offer precise knowledge of the edge-to-edge traffic flows as the basis for algorithms to compute an optimum path design for load balancing in normal operation as well as provisioning of backup paths for pre-computed failure scenarios.

Establishment of a full traffic engineering cycle includes:

1. measurement of the current edge-to-edge flow demands,
2. computation of an optimised network wide path design and
3. (re-)configuration of the computed design into the routers.

Traffic engineering tools [CoNa08], [HaSF05], [PiMe04] have been developed to support the full cycle, where a re-optimisation of traffic paths is recommended to adapt to any change in topology, upgrades of link bandwidth or major shifts being encountered in the traffic matrix. Those tools can also be used to investigate failure resilience and topology design for throughput optimisation.

The reducing of utilisation gaps after link upgrades is a main performance gain to be achieved by traffic engineering. Single link upgrades without further traffic redirection are subject to considerable inefficiency, since an upgrade to *k*-fold capacity subsequently diminishes the load on the link by a reciprocal factor 1/*k*. Figure 16 demonstrates utilisation gaps encountered in link upgrading processes for increasing traffic demand. Currently IP traffic is doubling within two years on fixed broadband access platforms and even within one year for mobile broadband access [MINT10]. MPLS traffic engineering enables to redirect transport paths from links in the surrounding to an upgraded link, such that newly installed bandwidth is instantly used to smooth down higher load on other links.

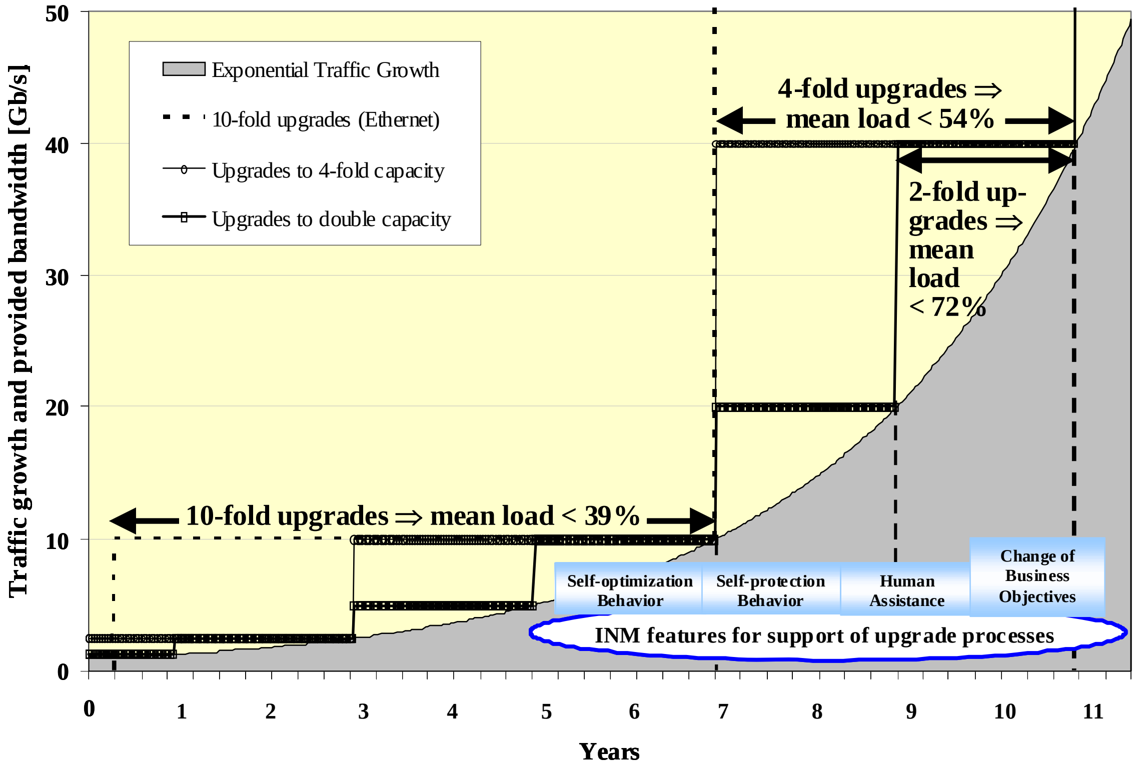


Figure 16. Link upgrades and resulting utilisation gaps for exponential traffic growth.

3.8.2 Approaches for Fast Failure Resilience

A major challenge for full deployment of the proposed path optimisation for load balancing mechanisms can be seen in the demand to integrate failure recovery with path optimisation and upgrading processes. A set of relevant failure cases including single link failures has to be taken care of by preparing automated switching to backup paths.

Resilience of a network with regard to a set of failure scenarios means that the network throughput covers all demands in the current traffic matrix in normal operation as well as in all considered failure cases, such that a worst case analysis of the network throughput has to be carried out over all scenarios in the failure set [HaSF05]. Traffic engineering tools can pre-compute optimised backup paths for each failure scenario. A single link failure usually triggers IP/MPLS rerouting and shifts the traffic from the failed link to backup paths. When network dimensioning in the backbone has to cover single link failures without optimisation tools, this often means a 50% reduction in allowable throughput with regard to a merge of traffic from two links on a provided backup path of the same capacity. Experience from [HaSF05] in meshed networks revealed a higher potential for throughput improvement by traffic engineering when single link failures are taken into account in addition to normal operation without failures.

For seamless continuation of real-time services, a delay below 50 ms is a crucial demand as achieved for path switching to bypass failed segments in Synchronous Digital Hierarchy (SDH) ring networks, which remains challenging for MPLS and other networking environments. Fast efficient monitoring algorithms have been studied as a central activity of Inherent Network Management and future Internet projects for a broad scope of networking technologies. Main contributions within the INM approach are addressing:

- Decentralised monitoring techniques based on gossiping and tree based distribution of information [PiSt09]; an alternative monitoring scheme is connecting self-organising entities in a structured peer-to-peer network [LiZa09] using distributed hash tables, where gossiping and peer-to-peer approaches can deal with dynamic networks subject to a high churn rate.
- Aggregation and evaluation of information in a management overlay, from simple functions, e.g., a sum or a minimum to more complex evaluations of network wide threshold crossings [WDSC07], [WuDS08].
- The estimation of size of the network or the size of groups with specific characteristics in the network [CoLa09] at minimum messaging overhead, as a basic function to be embedded in higher layer evaluations and applications.
- Efficient topology discovery using a new hide & seek principle for scalable information propagation and synchronisation [Guae09].
- Search methods for highly dynamic networks using combinations of flooding and random walks including information about the targeted path [HaKu09].

In addition, contributions in 4WARD and IMN include generic paths and path optimisation with regard to failure resilience [CoNa08], [HaSF05]. Together with standardisation for pre-configuration mechanisms [GON10], [SOCR10], the complete previously described traffic engineering cycle is supported, where the scope includes heterogeneous and dynamic networks via distributed schemes rather than static or slowly varying topologies. Even if the efficiency of traffic management differs in heterogeneous network environments, the trend to green IT puts pressure towards optimised resource utilisation, since savings in installed routing equipment also reduce energy consumption. The analysis of the efficiency of traffic engineering [HNMF10] shows a considerable potential for optimisation in terms of higher throughput which corresponds to 20% or more long term savings in capital expenditure (CAPEX) in resource deployment as well as reduced energy consumption on the operational expenditure (OPEX) side.

4 Conclusions

This document shows synergies and integration of WP4 “In Network Management” (INM) and WP5 “Forwarding and Multiplexing for Generic Paths” (GP) concepts. INM provides a new class of management functions, which are distributed and performed in real-time. These functions include all aspects related to network monitoring and resource adaptation. In order to implement these functions, the INM Framework has been also defined. Generic Path architecture is a new architectural concept that includes functions related to end-to-end communications in a multi-technology environment and includes nodes mobility handling, congestion control, network coding and efficient resource sharing mechanisms. Both mentioned approaches have some complementary functions but also competitive ones. In this document, several examples of the cooperation between INM and GPs have been shown. The first example shows an architecture integration possibility of INM and GPs; other examples show how INM can support:

- adaptive routing operations in wired and wireless environment;
- the Cooperation and Coding Framework (CCFW) developed in WP5;
- cross-layer congestion control which uses network coding techniques based on the network state information and anomaly related information;
- resource optimisation algorithm that is adaptive to long-term changes in the network;
- fast failure recovery that improves the re-routing process and utilisation of backup paths in multicast-trees, based on the availability information provided by the distributed anomaly detection mechanism;
- the virtual cognitive radio switch that combines several strategies from both WP4 and WP5;
- Traffic management operations.

The described integration of INM and the GP based architecture is – except one case, which was implemented – at the conceptual level only. Of course, the list of potential use cases that can show interactions of functions of INM and GPs is much longer. It has to be emphasised that in the current networks no real-time management operations exist. INM introduces new functionalities, which have to be taken into account and, in some cases, used by other blocks of the overall network architecture. Some of INM functions are reusable, for example network monitoring can be used for fault detection, path load monitoring, congestion or deny of service attacks detection. Due the separation of INM and GPs, the INM function can be accessed by different architecture components and their intrinsic algorithms can be optimised independently. The remark is also true for GPs related functions.

There is no doubt that more detailed analysis and proof-of-concept implementation of both architectures are required in order to evaluate the overall concept in more details (performance, signalling overhead, algorithms stability, etc.). In the framework of this project, such integration was not possible due to lack of time – the development of both concepts took significant part of the project time leaving no room for pretty complex validation of INM and GP concepts by implementation.

5 References

[ArZi10] P.A. Aranda Gutierrez, M. Zitterbart (eds.), *Final Architectural Framework,* 4WARD, Deliverable D-2.3.1, June 2010.

[AtZi08] A. Atlas, A. Zini, *Basic Specification for IP Fast Re-route: Loop-free-Alternates*, IETF RFC 5286, Sep. 2008.

[Awe04] B. Awerbuch, D. Holmer, H. Rubens, “High throughput route selection in multirate ad hoc wireless networks”, in *First IFIP TC6 Working Conference on Wireless On-Demand Network Systems* (WONS), Madonna di Campiglio, Italy, Jan. 2004.

[BiZH09] T. Biermann, Z. Polgar, K. Holger, “Cooperation and Coding Framework”, in *Proc. International Workshop on the Network of the Future (Future-Net)*, Dresden, Germany, June 2009.

[Bop01] R.V. Boppana, S. Konduru, “An adaptive distance vector routing algorithm for mobile, ad hoc networks”, in *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Anchorage, Alaska, USA, Apr. 2001.

[Bro98] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols”, in *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, USA, Oct. 1998.

[CoLa09] R. Cohen, A. Landau, "Not All At Once!, A generic Scheme for Estimating the Number of Affected Nodes While Avoiding Network Implosion”, in *Proc. of Infocom'2009 mini-conference*, Rio de Janeiro, Brazil, Sep. 2009.

[CoNa08] R.Cohen, G.Nakibly, “Maximising Restorable Throughput in MPLS Networks”, in *Proc. of IEEE Infocom*, Phoenix, AZ, USA, Apr. 2008.

[Cou03] D.De Couto, D. Aguayo, J. Bicket, R. Morris, “A High Throughput Path Metric for Multi-Hop Wireless Routing,” in *ACM Mobicom Conference*, San Diego, CA, USA, Sep. 2003.

[Das00] S. R. Das, R Castaneda, J. Yan, “Simulation-based performance evaluation of routing protocols for mobile ad hoc networks”, *Mobile Networks and Applications (MONET)*, Vol. 5, No. 3, Mar. 2000, pp. 179-.189.

[Dra04] R. Draves, J. Padhye, B. Zill, “Comparison of routing metrics for static multi-hop wireless networks”, in *ACM Special Interest Group on Data Communication (SIGCOMM)*, Portland, Oregon, USA, Feb. 2004.

[Ferr10] L.S. Ferreira (ed.), *First Report on TC45 WP4 (INM) / WP5 (ForMux) Joint Task*, 4WARD, Deliverable R-TC45.1, Jan. 2010.

[Gonz10] A. Gonzalez (ed.), *In-Network Management Design*, 4WARD, Deliverable D-4.3, Jan. 2010.

[Guae09] L. Guardalben et al., *HISKY: A Cooperative Hide and Seek Discovery over In Network Management*, Technical Report, Instituto de Telecomunicações, Portugal Telecom Inovação, www.av.it.pt/ssargento/pub_report.html, 2009.

[HaKu09] G. Haßlinger, T. Kunz, "Challenges for Routing and Search in Dynamic and Self-organising Networks", in *Proc. Ad Hoc Now Conference*, Murcia, Spain, May 2009.

[Has97] J. Haas, "A new routing protocol for the reconfigurable wireless networks", in *Proc. of IEEE 6th International Conference on Universal Personal Communications,* San Diego, CA, USA, Oct. 1997*.*

[HaSF05] G. Haßlinger, S. Schnitter, M. Franzke, "The efficiency of traffic engineering with regard to failure resilience", *Telecommunication Systems,* Vol. 29, No. 2, 2005, pp. 109-130.

[HNMF10] G. Haßlinger, G. Nunzi, C. Meirosu, C. Fan, F.-U. Andersen, "Inherent Network Management: Support for Traffic Engineering and CAPEX Analysis in Future Internet Scenarios, *accepted to Proc. EUROMICRO SEAA Conference*, Lille, France, Sep. 2010.

[LiZa09] A. Liakopoulos, A. Zafeiropoulos, "Autonomic Monitoring and Resource Management Using P2P Techniques", in *Proc. of Selected Papers, TERENA networking conference*, Malaga, Spain, June 2009.

[MINT10] MINTS, *Minnesota Internet traffic studies*, www.dtc.umn.edu/mints/home.php, 2010.

[Nunz10] G. Nunzi (ed.), *In-network management system demonstrator*, 4WARD, Deliverable D-4.4, May 2010.

[Per94] C.E. Perkins, P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers", *Computer Communication Review*, Vol. 24, No. 4, Oct. 1994.

[PiMe04] M. Pioro, D. Mehdi, *Routing, Flow & Capacity Design in Communication and Computer Networks*, Morgan Kaufmann, 2004.

[PiSt09] A. Gonzalez Pietro, R. Stadler, "Controlling Performance Trade-offs in Adaptive Network Monitoring", in *Proc. of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009)*, New York, USA, Sep. 2009.

[PoKi09] Z. Polgar, Z. Kiss, A.B. Rus, G. Boanea, M. Barabas, V. Dobrota, "Preliminary Implementation of Point-to-Multi-Point Multicast Transmission Based on Cross-Layer QoS and Network Coding", in *Proc. of 17th Int. Conf. on Software, Telecommunications & Computer Networks IEEE SOFTCOM 2009*, Split-Hvar, Croatia, Sep. 2009, pp.131-135

[Ram 03] V. Ramasubramanian, Z.J. Haas, E.G. Sirer, "SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks", in *Proc. ACM MOBIHOC*, Annapolis, Maryland, USA, June 2003.

[Rand09] S. Randriamasy (ed.), *Mechanisms for Generic Paths*, 4WARD, Deliverable D-5.2, Dec. 2009

[RFC3561] RFC 3561, *Ad hoc On-Demand Distance Vector (AODV) Routing*.

[RFC3626] RFC 3626, *Optimised Link State Routing Protocol (OLSR)*.

[RuBa10] A.B. Rus, M. Barabas, G. Boanea, Z. Kiss, Z. Polgar, V. Dobrota, “Cross-Layer QoS and Its Application in Congestion Control”, in *Proc. of 17th IEEE Workshop on Local and Metropolitan Area Networks LANMAN 2010*, Long Branch, NJ, USA, May 2010.

[SOCR10] SOCRATES, *Self-Optimisation and Self-configuration in Wireless Networks*, FP7 European Commission project, www.fp7-socrates.org, 2010.

[StGi09] R. Steinert, D. Gillblad, *An initial approach to distributed adaptive fault-handling in networked systems*, SICS, Kista, Sweden, Rep. T2009:07, 2009.

[Wah08] S. Waharte, B. Ishibashi, R. Boutaba, D. Meddour, "Performance study of wireless mesh networks routing metrics," in *Proc. of IEEE/ACS International Conference on Computer Systems and Applications*, Doha, Qatar, Apr. 2008.

[WDSC07] F. Wuhib, M. Dam, R. Stadler, A. Clemm, “Robust Monitoring of Network-wide Aggregates through Gossiping”, in *Proc. Integrated Management* (2007),and *accepted for IEEE Transactions on Network and Service Management*.

[WuDS08] F. Wuhib, M. Dam, R. Stadler, “Decentralised Detection of Global Threshold Crossings Using Aggregation Trees”, *Computer Networks*, Vol. 52, 2008, pp. 1745-1761.