Information and Communication Technologies (ICT)

Grant Agreement Number :	216366
Project Acronym :	Euro-NF
Project Title :	Anticipating the Network of the Future – From Theory to Design
Funding Scheme :	Network of Excellence
Start Date of Project :	01 January, 2008
Duration :	54 months (originally 36)

Deliverable Number :	D.SEA.7.2.7
Version Number :	1.0
Title of Deliverable :	Report on the Seventh Future Internet Cluster Workshop
Contractual Due Date :	May, 2012
Actual Date of Completion :	June, 2012
Workpackage contributing to the Deliverable :	SEA.7.2 – Concertation with European Projects and Other Organizations
Lead Contractor for this Deliverable :	BTH (partner 11)
Editor{s} :	Markus Fiedler, Kurt Tutschku

Nature of the Deliverable : (R/P/D/O)*	R
Dissemination Level : (PU, PP, RE, CO)**	PU

*Nature: R-report, P-prototype, D-demonstrator, O-other
**Dissemination: PU-public, PP-restricted to programme, RE- restricted to a group, CO-confidential

Project URL: http://www.euronf.org

CONTENTS

1 GENERAL INFORMATION

Workpackage Number: SEA 7.2
Full name of the event: Seventh Future Internet Cluster Workshop
Date: February 13th, 2012
Place: European Commission, Brussels
Web page for the event: http://euronf.enst.fr/p_en_Events_Concertati_INSTFP7_Clust12Feb_618.html
Partner in charge of the organization: BTH, Partner 11
Chairpersons: Markus Fiedler (BTH, Partner 11)
Kurt Tutschku (UNI WIEN, Partner 04)

2 MAIN OBJECTIVES

In the recent years, several initiatives have proposed views on what the networks of the future could be. Recent concertation meetings were devoted to introduce the projects contributing to this domain as well as their own visions and roadmaps. These meetings were fruitful as facilitators for information exchange and for promoting synergies among the projects.

The natural next step was to propose to join forces in facing the key challenges to transform these visions into reality, both from the scientific and technological as well as from the socio-economic points of view. Considering the very broad scope of the networks of the future, it appeared much more realistic and promising to address successively specific topics with a broad potential impact for Europe through dedicated workshops, which would enable focused exchange of information and views among experts with strong interest in the respective fields.

In close cooperation with the European Commission, Euro-NF promoted this idea and revisited the format of the Future Internet Cluster meetings held under the auspices of the Commission about every four months as part of the overall concertation process. This led to the concept of Future Internet Cluster Workshops (FICW). The first FICW was held at ETSI in Sophia-Antipolis on March 09, 2010, and was followed by six FICW. The most recent FICW, organised at the European Commission's premises in Brussels on February 13, 2012 and henceforth called FICW7, is reported. Apart from minor variations, the workshops kept their overall format with one or two half-day sessions with a 50:50 split between presentation and moderated discussion time.

3 SCIENTIFIC CONTENT AND PROGRAMME

This half-day Seventh Future Internet Cluster Workshop (FICW7) was devoted to the topic **Novel Networking Architectures**, with one session on **The Impact of Software Defined Networking, OpenFlow and new Flow Switching Technologies**.

As opposed to earlier instances and due to a short preparation timeline, this FICW built its programme upon invited talks during the presentation session. The invited speakers were invited to address a set of pre-defined questions, and they also served as panellists. Unfortunately, one speaker dropped off due to circumstances out of control.

Programme

14:00 Introduction to the topic

European Commission and Markus Fiedler, Future Internet Cluster Co-Chair / Euro-NF

14:15 Presentation session

Chair: Markus Fiedler, Blekinge Institute of Technology / Project Euro-NF

- Introduction to the topic
- Olivier Bonaventure, Université catholique de Louvain / Project CHANGE:
Flow Processing and the Rise of the Middle
- Andreas Gladisch, Deutsche Telekom / Project SPARC:
Applications of SDN in Carrier Networks
- Hagen Woesner, European Center for Information and Communication Technologies GmbH / Project OFELIA:
Is OpenFlow the Future Internet?
- *Michael Jarschel, University of Würzburg / Project G-LAB:*
Scalability Issues of SDN on the Example of OpenFlow (CANCELLED)
- Giuseppe Bianchi, University of Roma Tor Vergata / Project FLAVIA:
Wireless Access Programmability: New Architectural Insights and Proof-of-Concept Validation on Commodity Cards

15:30 Coffee break

16:00 Discussion session

Chair: Kurt Tutschku, University of Vienna / Project Euro-NF

- Panellists as above

17:30 End

Rapporteurs: Markus Fiedler and Kurt Tutschku

4 SPONSORS

The sponsorship of the European Commission, providing topical, organizational and promotion support, as well as venue and catering, is gratefully acknowledged.

5 SCIENTIFIC COMMITTEE

Partner Number	Partner Acronym	Contributor Name	Contributor e-mail address
04	UNI WIEN	Kurt Tutschku	kurt.tutschku@univie.ac.at
11	BTH	Markus Fiedler	markus.fiedler@bth.se
External	EC	Pertti Jauhiainen	pertti.jauhiainen@ec.europa.eu
External	EC	Rüdiger Martin	ruediger.martin@ec.europa.eu

6 ORGANISING COMMITTEE

Partner Number	Partner Acronym	Contributor Name	Contributor e-mail address
01	GET	Jean-Paul Lefèvre	jean-paul.lefevre@arches-conseil-grenoble.eu
04	UNI WIEN	Kurt Tutschku	kurt.tutschku@univie.ac.at
11	BTH	Markus Fiedler	markus.fiedler@bth.se
External	EC	Pertti Jauhiainen	pertti.jauhiainen@ec.europa.eu
External	EC	Rüdiger Martin	ruediger.martin@ec.europa.eu

7 NUMBER AND LIST OF PARTICIPANTS

7.1 Number of Participants

- External Participants: 17
- Euro-NF Members: 4

7.2 List of Participants

Partner Number	Organization	Participant Name
External	Ericsson	Henrik Abramowicz
External	EC, INFSO F4	Jacques Babot
External+	CNIT / University of Roma Tor Vergata	Giuseppe Bianchi
External	Università di Genova	Raffaele Bolla
External+	UCL	Olivier Bonaventure
11*	BTH	Markus Fiedler
External	i2CAT Foundation	Sergi Figuerola
External	University College-London	Alex Galis
External+	Deutsche Telekom	Andreas Gladisch
External	France Telecom SA	Alexandre Gouraud
External	Eurescom	Ádám Kapovits
External	EC	Pertti Jauhiainen
External	EC	Rüdiger Martin
20	IT Aveiro, Portugal	Jonathan Rodriguez
External	Ericsson	Benoit Tremblay
4*	University of Vienna	Kurt Tutschku
36	Alcatel-Lucent	Martin Vigoureux
External	Eurecom	Michelle Wetterwald
External	Ericsson	Fiona Williams
External+	EICT GmbH	Hagen Woesner
External	Heidelberg-Germany	Frank Zdarsky

+ Speaker
* Organiser

8 COMPREHENSIVE SUMMARY FOCUSING ON THE MAJOR OUTCOMES

8.1 Invited Talks

For the preparation of the talks, the speakers were invited to address the following set of optional questions:

1) How do you define "Software Defined Networking", "Network Programmability" and/or "Flow Switching"? Where do you see the main advantages e.g. of "OpenFlow"?
2) Which are the key components of your novel architecture?
3) Which novel functionality is provided? In particular, what application is enabled by the technologies developed in your project beyond what is possible today?
4) To which extent do the technologies developed in your project improve application performance? Are there any benchmarking results available? What is the overall experience with your approach?
5) Which are the threats for not adopting the technologies developed in your project?
6) How can the technologies of your project inter-work with the other technologies presented in the cluster workshop?

8.1.1 Olivier Bonaventure, Université catholique de Louvain / Project CHANGE: Flow Processing and the Rise of the Middle

The talk brought forward the role of middleboxes as important entities for enabling (or disabling) good flow transfer properties in current and future Internet. Its first part described the situation of "TCP in the wild" in current internet and identified frequent violations of the end-to-end principle such as removal of TCP options, re-segmentation, modified sequence numbers and proxy acknowledging by middleboxes; those problems are not expected not disappear with IPv6. On the background that current Internet is a collection of IP network connected by hundreds of middleboxes, the second part advanced the need to advance the functionality of those boxes towards flow- and application-aware middleboxes that serve both the network provider (by offering efficient management functionality) and application developers (by offering possibilities to improve applications). Considering flow processing as first class primitive, it was claimed that a scalable extensible software platform is needed for featuring innovations instead of hampering them in ways as current middleboxes do.

8.1.2 Andreas Gladisch, Deutsche Telekom / Project SPARC: Applications of SDN in Carrier Networks

The talk started with a review of possibilities to program a switch or router, mentioning the plethora of operating systems, SDKs and APIs. Network processing was discussed, and the need of splitting control and hardware (the so-called SPlit ARChitecture at the core of SPARC) was motivated, thereby highlighting the major functions processing, generic forwarding and an open, standardised interface (such as OpenFlow). Use cases in the context of aggregation/access and datacenters were presented. Finally, all questions listed above were addressed in detail.

8.1.3 Hagen Woesner, European Center for Information and Communication Technologies GmbH / Project OFELIA: Is OpenFlow the Future Internet?

The talk with the subtitle "If not, will it at least help building it?" started by illustrating relationships between Software-Defined Networking, OpenFlow, Split Architecture and Future Internet. A set of deficiencies of the current Internet regarding naming, resolving and layering were identified, and the benefits by new approaches, including OpenFlow, were discussed. In particular, the separation of control, processing and data path as well as the need for a Network Operating System allowing for flexible matches and actions. Key components of the new architecture were described, such as pattern / flow recognition and network slicing facilities. In this context, the contents of a base class of an OpenFlow controller were presented, followed by an intensive discussion (including example) of virtualization as a must for successful splitting control, forwarding and processing. The presentation concluded that, if extended properly, OpenFlow might help Future Internet research and experimentation, and with a set of requirements on an OpenFlow-based Network Operating System (including support by a – potentially European – support community).

8.1.4 Michael Jarschel, University of Würzburg / Project G-LAB: Scalability Issues of SDN on the Example of OpenFlow

Cancelled due to circumstances out of control.

8.1.5 Giuseppe Bianchi, University of Roma Tor Vergata / Project FLAVIA: Wireless Access Programmability: New Architectural Insights and Proof-of-Concept Validation on Commodity Cards

The talk motivated the need to network (and in particular Wireless Access) programmability by the fact that in current networks, every time a new component comes in, significant changes have to be made, which hinders fast deployment. Wireless programmability as a means of abstraction and decoupling was described together with enabling concepts for opening the wireless stack such as "softMAC", open firmware and DSP/FPGA boards. The anatomy of MAC protocols was addressed and categorised into actions, events, conditions and logic. The proposal of a Wireless Processor engine as generic XFSM executor was advanced, and the example of a MAC program for this environment was discussed. Three functional validations were presented for (1) piggybacked acknowledgements; (2) pseudo-TDMA; and (3) randomised multi-channel access. The presentation concluded with status review of R&D work related to the proposed Wireless Processor.

8.2 Moderated Discussion

Moderator: Kurt Tutschku, University of Vienna / Project Euro-NF
Participants in the panel discussion: the above speakers and the audience.

In the following, a summary of the panel discussion is presented in a clearly laid-out way, sorted by topics related to Software-Defined Networking (SDN) and OpenFlow:

- **Why not just MPLS?** MPLS is included in OpenFlow, but OpenFlow does not need the full-blown control plane of MPLS (and is thus cheaper).
- **Differentiation to Active Networks (AN)**: SDN is an evolutionary approach on top of current Internet; as opposed to AN, routers can remain unchanged.

- **Rules and roles**: Who is setting the rules how to switch traffic streams? Operators want to remain in charge; it is considered dangerous to leave (full) control to the end users.
- **How about end-to-end capacity?** Will probably not be provided in the Internet context, and not either by OpenFlow. However, the user will be moved closer to the data centers and thus profit from better connectivity.
- **Need for a good access network** to move the user closer to the data centers
- **Use cases for OpenFlow**:
  (1) providing a common API;
  (2) managing flows;
  (3) new programming language.
- **How about APIs?** APIs should provide a compact (and elegant) set of primitives; vendors can distinguish themselves from each other. APIs are expected to replace Command Line Interfaces (CLI) for management purposes.
- **The role of software.** Software will be decoupled from hardware, which will help in terms of innovation. Besides of providing flexibility, software will require rigidity when it comes to requirements and validation (?)
- **Added values**? (1) cost reductions; (2) universal, standardized control interface; (3) decoupled hardware/software innovation that allows for new ideas; (4) economics of scale
- **What is missing in OpenFlow?** Resource information is considered orthogonal (?). More interaction between applications and networks is required ("from smart grids to smart networks")
- **Business cases:** In particular, strategic placement of the corresponding boxes helps to provided added value even for smaller players (such as SMEs that are usually excluded from large datacenters). Through its inherent flexibility and abstraction, SDN/OpenFlow will ease the way of smaller players into the market.
- **How likely is it to get there, and what has to be done?**
  (1) There is a global community with a large momentum to go into open networking.
  (2) Good progress is expected (OpenFlow 1.1/1.2), even for smaller players.
  (3) There is a market need for bringing service closer to the customer, and for standardizing.
  (4) It's a matter of price; in case the new equipment is one order of magnitude cheaper than today, OpenFlow is going to win.
  (5) The providers need to be a part of it.

8.3 Summary

The outcome of the 7th Future Internet Cluster workshop is that **Europe has an excellent chance for driving the novel networking architectures,** which based, for example, on Software Defined Networking. In particular **five recommendations can be given:**

1) Software Defined Network, Network Virtualization and Network Federation are upcoming new features for European operators;

2) Europe should play on the strengths of mobility management and smart access aggregation;
3) Novel operational techniques can be developed in Europe to sustain Europe's strength in operator business;
4) Europe needs to attract more people at universities and SMEs to advance the topic;
5) The promotion of test facilities and other playgrounds is important.

9 OVERALL QUALITATIVE ASSESSMENT

This seventh workshop was mainly attended by non-Euro-NF delegates, most of them involved in the Future Internet Cluster Meeting that took place on February 13 and 14, 2012.

The overall format appeared to be well-adapted to the goal of triggering a brainstorming of open (research) issues in the selected areas. Eventual questions after the talks were kept as fuel for the discussion session, which became lively and engaged. This means that the overall format worked well,

Suggestions for future workshops would be to

- Keep the overall format;
- Keep on co-locating with major events, as this is of key importance for participation;
- Focus on "hot topics";
- Consider a more flexible structure: i.e. a more variable limit (e.g. 40:60 or vice versa) between presentation part and discussion part (depending on the topic);
- Make sure that eventual statements do not consume too much discussion time;
- Provide a set of focal points for the brainstorming.

10 FINANCIAL ASPECTS

As already mentioned, the EC took entirely in charge the practical organisation of the workshop, including catering. Again, this is gratefully acknowledged.

Euro-NF supported only its key contributors through the following flat grants, covered from the respective partners' budget:

- Session chairs, acting also as moderators and rapporteurs: 1 200 € each (2 grants)

APPENDIX 1: CALL FOR PARTICIPATION

See also http://ec.europa.eu/information_society/events/cf/fnc9/item-display.cfm?id=7900

Future Internet (FI) Cluster Meeting

Brussels, Avenue de Beaulieu 25, Room S1

The Future Internet Cluster has both intensive contributions to the Future Internet Assembly (FIA) activities and also specific activities of research towards the internet of the future, including both evolutionary and clean-slate approaches. The projects look not only at the technical aspects, but also at the economic and social implications of the current and future networking architectures: several projects have large work packages on socio-economic issues, while few others address the Green Internet issues.

LINKS AND DOCUMENTS

- Agenda [v10/02/2012] (228 KB)

Contact: Pertti JAUHIAINEN (European Commission, Belgium)

PRESENTATIONS

5 Presentations

- 7th Future Internet Cluster Workshop
- Flow Processing and the Rise of the Middle
- Applications of SDN in Carrier Networks
- Is OpenFlow the Future Internet?
- Wireless Access Programmability: New Architectural Insights and Proof-of-Concept Validation on Commodity Cards

APPENDIX 2: MATERIAL FOR DISSEMINATION

Slides from the presentations are enclosed hereafter, and are also available on the Euro-NF Web site at: http://euronf.enst.fr/p_en_Events_Concertati_INSTFP7_Clust12Feb_618.html

Flow processing and the rise of the middle.

Mark Handley, UCL (UK)

Presented by Olivier Bonaventure, UCL (Belgium)

Part 1

Today's Internet

Protocol Layering

- Link layers (eg Ethernet) are local to a particular link
- Routers look at IP headers to decide how to route a packet.
- TCP provides reliability via retransmissio ow control, etc.
- lication using OS's API to d

The usual suspects

- NATs are ubiquitous
  - We've become pretty good at working around them.
- Firewalls are ubiquitous
  - Ability to communicate using one port does not imply that communication is possible on any other port.

What actually happens to TCP in the wild?

- We studied 142 access networks in 24 countries.
- Ran tests to measure what actually happened to TCP.
  - Are new options actually permitted?
  - Does re-segmentation occur in the network?
  - Are sequence numbers modified?
  - Do middleboxes proactively ack?

Middleboxes and new TCP Options in SYN

Observed	TCP Port		
Behavior	34343	80	443
Passed	129 (96%)	122 (86%)	133(94%)
Removed	6 (4%)	20 (14%)	9 (6%)
Changed	0 (0%)	0 (0%)	0 (0%)
Error	0 (0%)	0 (0%)	0 (0%)
Total	135 (100%)	142 (100%)	142 (100%)

- Middleboxes that remove unknown options are not so rare, especially on port 80

What actually happens to TCP in the wild?

- **Rewrote sequence numbers**:
  - 10% of paths (18% on port 80)

  - Two probable causes:
    - » TCP-level proxy behaviour
    - » Firewalls trying to improve initial sequence number randomization

What actually happens to TCP in the wild?

- Testing for TCP-level proxies:
  - **Resegmented data**: 3% of paths (13% on port 80)
  - **Proxy Ack**: 3% of paths (7% on port 80)

  - Note: all of these paths also removed new options from the SYN

What actually happens to TCP in the wild?

- **Ack data not sent:**
  - 26% of paths (33% on port 80) do strange things if you send an ack for data not yet sent.
    - » Drop the ack
    - » "correct" it.

What actually happens to TCP in the wild?

- **Rewrote sequence numbers**:
  - 10% of paths (18% on port 80)
- **Resegmented data**:
  - 3% of paths (13% on port 80)
- **Proxy Ack**:
  - 3% of paths (7% on port 80)
- **Ack data not sent:**
  - 26% of paths (33% on port 80) do strange things if you send an ack for data not yet sent.

Not to mention…

- NAT
  - Pretty nearly ubiquitous, but comparatively benign
- DPI-driven rate limiters
- Lawful intercept equipment
- Application optimizers
- Anything at the server end:
  - Firewalls
  - Reverse proxies
  - Server load balancers
  - Traffic scrubbers
  - Normalizers, etc

Our methodology will not detect most of these, but we're pretty sure they're out there too.

IPv6 will save us!

- No.

CHANGE

Part 2: Tomorrow's Internet

Option 1: Extrapolate the current Internet

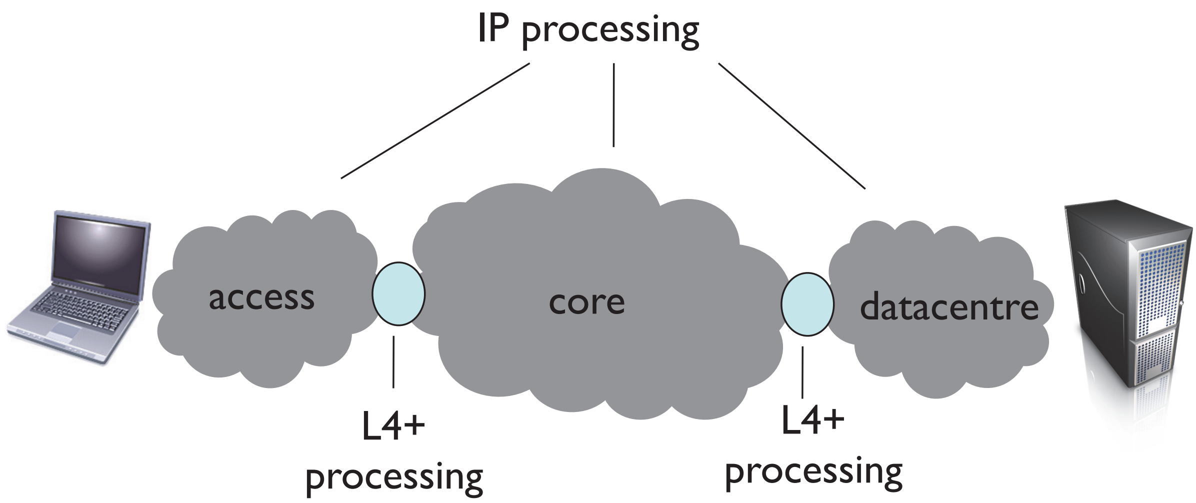
- Plenty of box vendors will sell you a solution.
  - Whatever you think your problem is.
- Current apps get optimized and set in silicon.
- Future apps tunnelled over HTTP
  - (but what do all those port 80 specialized middleboxes do?)
- Impossible to reason about the concatenation of middleboxes.
  - If you think STUN/TURN/ICE is hard to reason about, you've not seen anything yet,

Option 2: Devise a wonderful new Internet architecture that everyone will love and deploy.

Option 3: Reverse engineer a new Internet <u>architecture</u> from the current mess.

- Observation: The Internet is becoming a concatenation of IP networks interconnected by L4+ functionality.

A segmented Internet

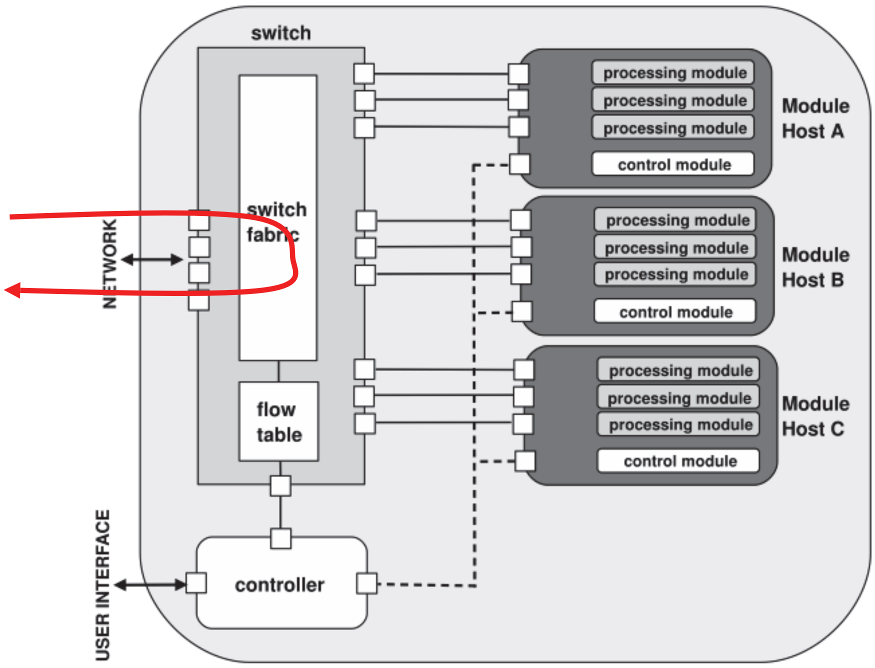


It already looks somewhat like this, but the L4+ processing is more distributed.

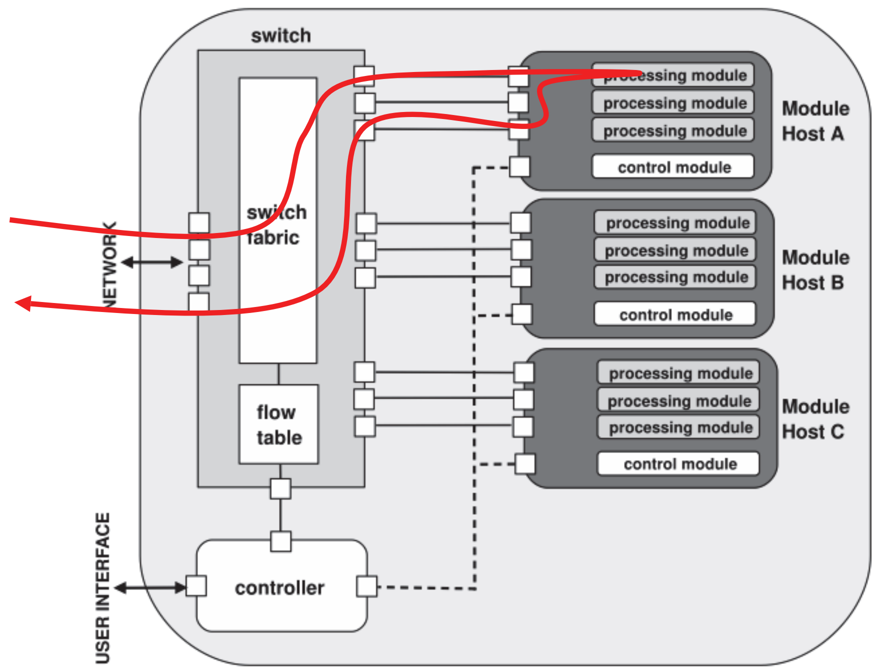
A platform for CHANGE

- Those L4+ platforms need to be more general than today's middleboxes.
  - More open.
  - More upgradable, as new apps arrive.
  - Aggregate functionality, so it is manageable.
  - Identifiable, so we can reason about them
  - Cheap and scalable.

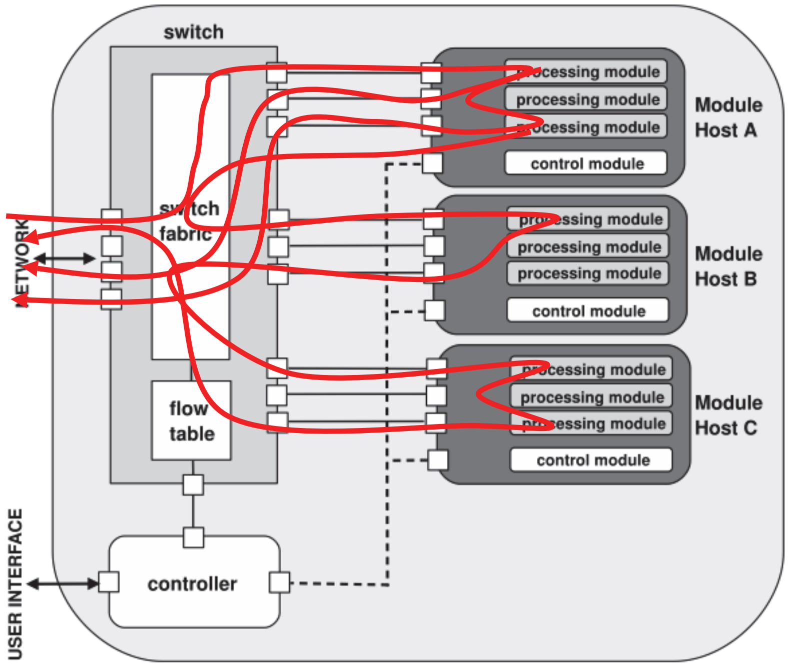
Flowstream



Flowstream



Flowstream



A better mousetrap

- Change is <u>not</u> primarily about building a better middlebox.
  - Though much of the effort goes on this.
- The observation is that the Internet has already embraced **flow processing**, albeit implicitly.
  - We believe we need to make flow processing a <u>**first-class citizen**</u> within the Internet architecture.

Application middleboxes

- Many applications are already built around middleboxes:
  - Skype supernodes
  - SMTP servers and IMAP servers for email.
  - CDNs for video streaming.
- Unlike ISP-imposed middleboxes, these are:
  - Application specific
  - Directly addressable

Unified Goal

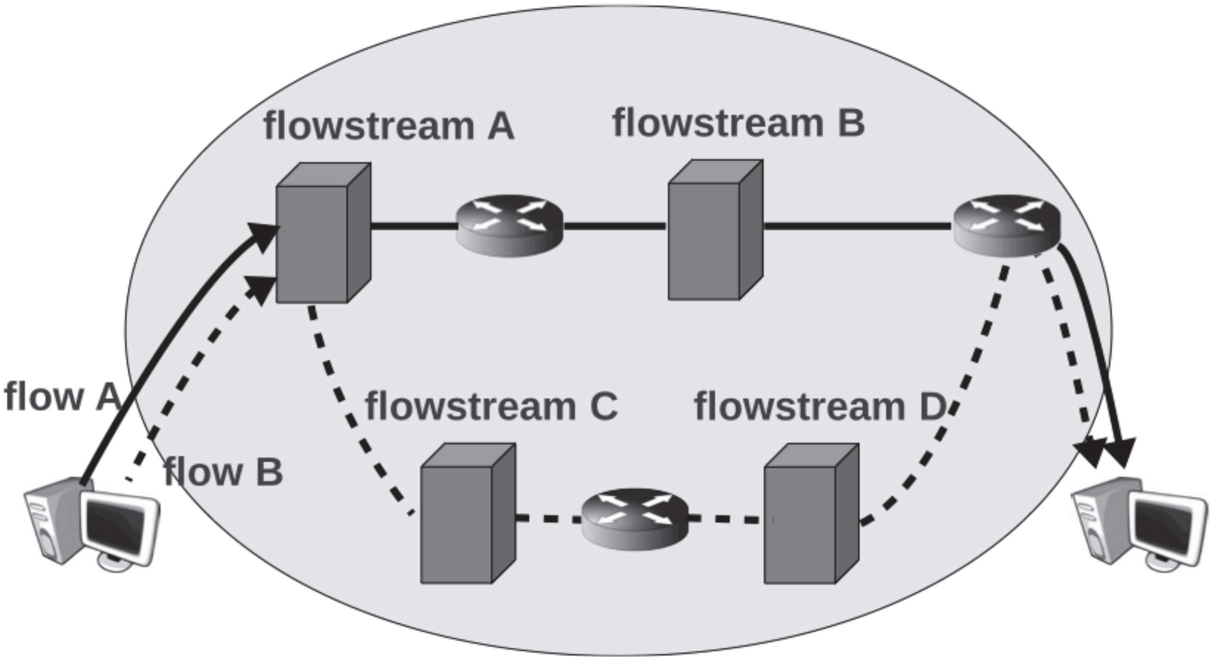
» *Application middleboxes*

» *ISP-imposed middleboxes*

Our goal is to provide a unifying framework that can perform both middlebox roles.

- Network operators can manage their net effectively
- App developers can enhance their applications.

Empowering both the ends and the middle



Architectural components

- A scalable general purpose flow processing platform.
- A categorization of flow processing into a few classes.
  - Allows reasoning about concatenation of processing without needing to know the details.
- A way to identify who can request processing.
- A way to name flows to be processed.
- A way for end-systems to discover platforms which they can enlist to perform processing.
- A way to attract flows to a flow processing platform.

http://www.change-project.eu/

- Flow processing as a first class primitive
- Scalable extensible software platform to enable it.
- Mechanisms to remotely authorize instantiation of processing and protocols to communicate with flow processing platforms.
- Architectural framework to reason about the emergent behaviour of the network.

Going with the flow…

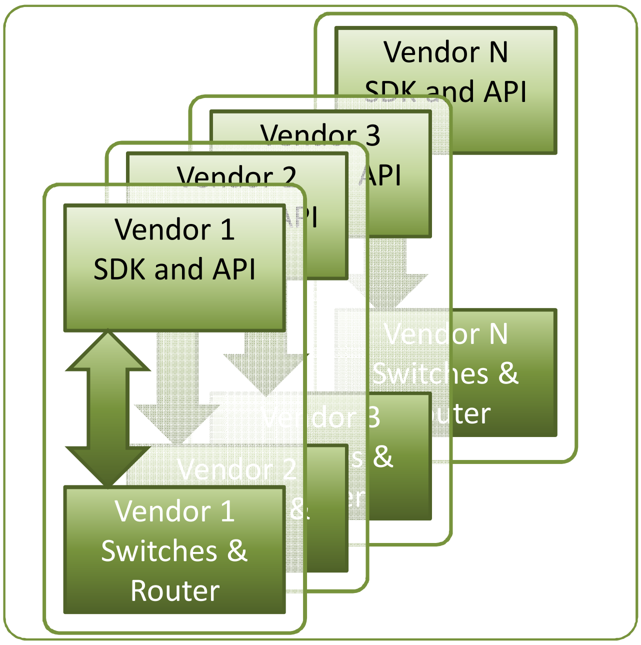
- Currently flow processing in middleboxes serves to inhibit new applications.
  - Optimization of the present
  - Inextensible inflexible network security
- **Key question: is it possible to re-claim the middlebox as a force for enabling end-to-end innovation?**

SPARC – **Split Architecture**

Applications of SDN in Carrier Networks

Andreas Gladisch
Deutsche Telekom AG, Innovation Laboratories

How do you program a switch / router today ?



- High variety of
  - different operation systems,
  - software development kits and
  - API
- Specific per system vendor and per system generation
- Closed solutions
  - Usually integration done by vendors
  - Most of software can not be used for other systems or other vendors

Hardware - vendor specific software “ECO-systems “ !

What do you want to program?
Forwarding and Addresses.

- A Data Packet today (PBB simplified)

PBB MAC Destination address	PBB MAC Source address	Q-Tag (B-VLAN)	Q-Tag (I-VLAN)	MAC Destination address	MAC Source address	Q-Tag (S-VLAN)	Q-Tag (C-VLAN)	Client Length Type

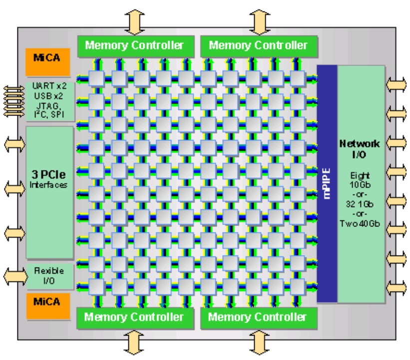
IP Header	IP Source	IP Destination	DATA

- Address pair defines a logical layer
- Highly complex hardware
- Specific hardware usually related to specific addressing schemes, limited set of rules related to hardware (add-dest. port out)
- Nearly the same functions with different address schemes ((G)MPLS, PBB etc.)

Can we benefit from generic match / forwarding?

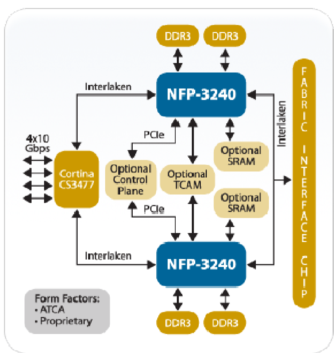
What do you want to program?
Network Processing.

Example 1: Tilera 100 core network processor



Source: http://www.tilera.com/products/processors/TILE-Gx_Family

Example 2: Netronome programmable line-card



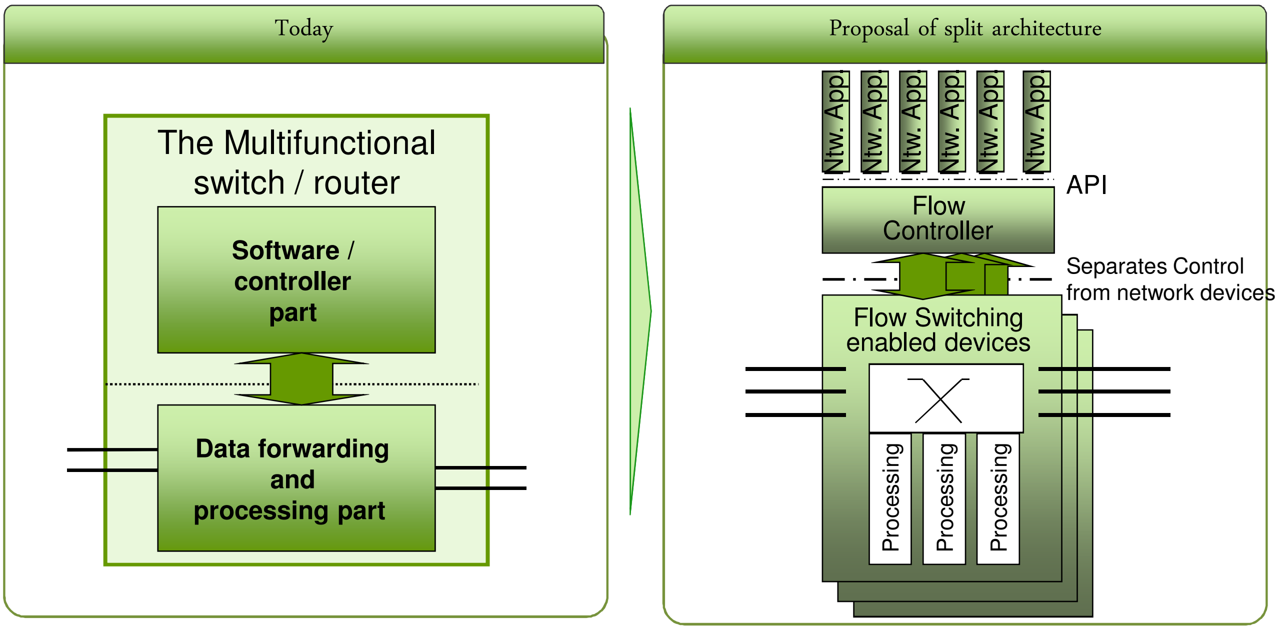
http://www.netronome.com/pages/switching-routingration

Example 3: Alcatel Lucent 400G network processor

http://www.alcatel-lucent.com/fp3/

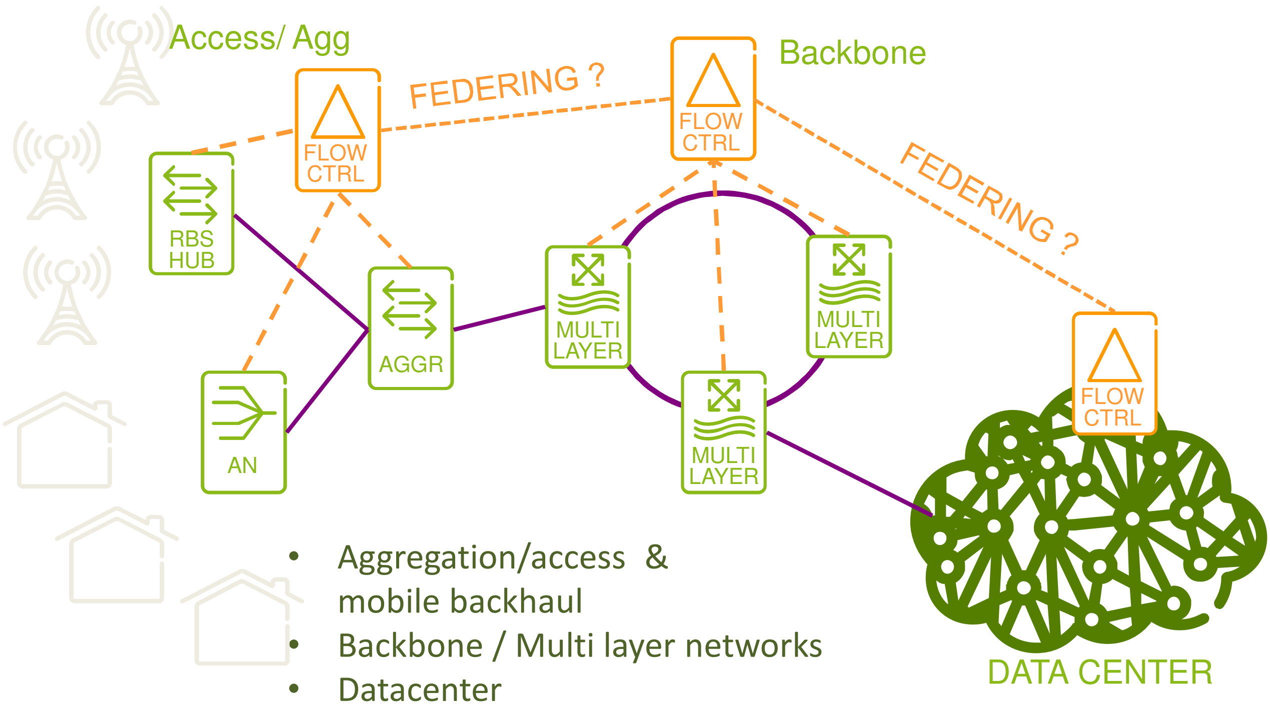
Enormous Progress in Chip technology enables highly programmable forwarding hardware

Essential of Split Architecture.
How to split today's switch / router.



Split of software centric and hardware centric part;
Split of application / control (software); split of forwarding / processing (hardware)

Use Cases.
SPARC concentrate on Aggregation/Access.



- Aggregation/access & mobile backhaul
- Backbone / Multi layer networks
- Datacenter

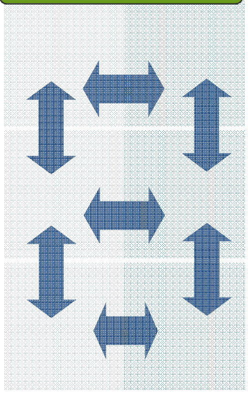
The challenges of future (datacenter) networks.
Seamless virtualization needed.

Technology

TC	IT
IP- VPN Router Virtual Router	VM- Ware Server Virtual Server
E-VPWS E-VPLS VLAN	V Sphere Virtual Storage Storage
Virtual Fiber Lambda Fiber	

The long term vision

ICT

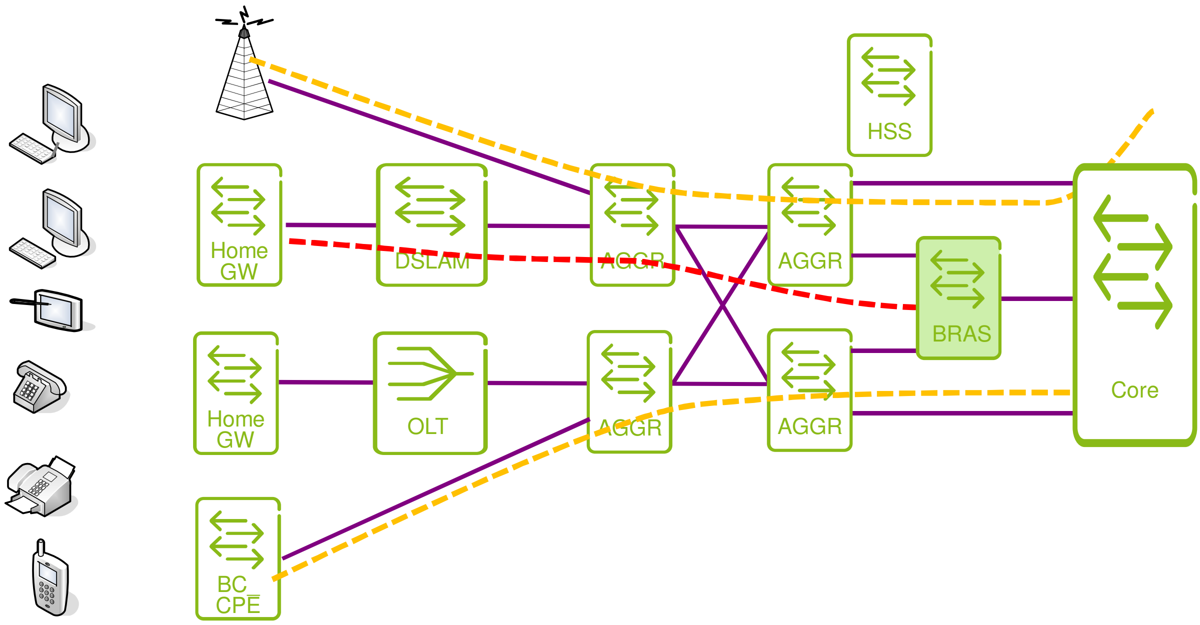


Seamless ICT virtualization

- infrastructure management
- business models
- business processes

Still a gap between TC and IT virtualization: Has to be closed

Access Aggregation Scenarios.
State of the Art.

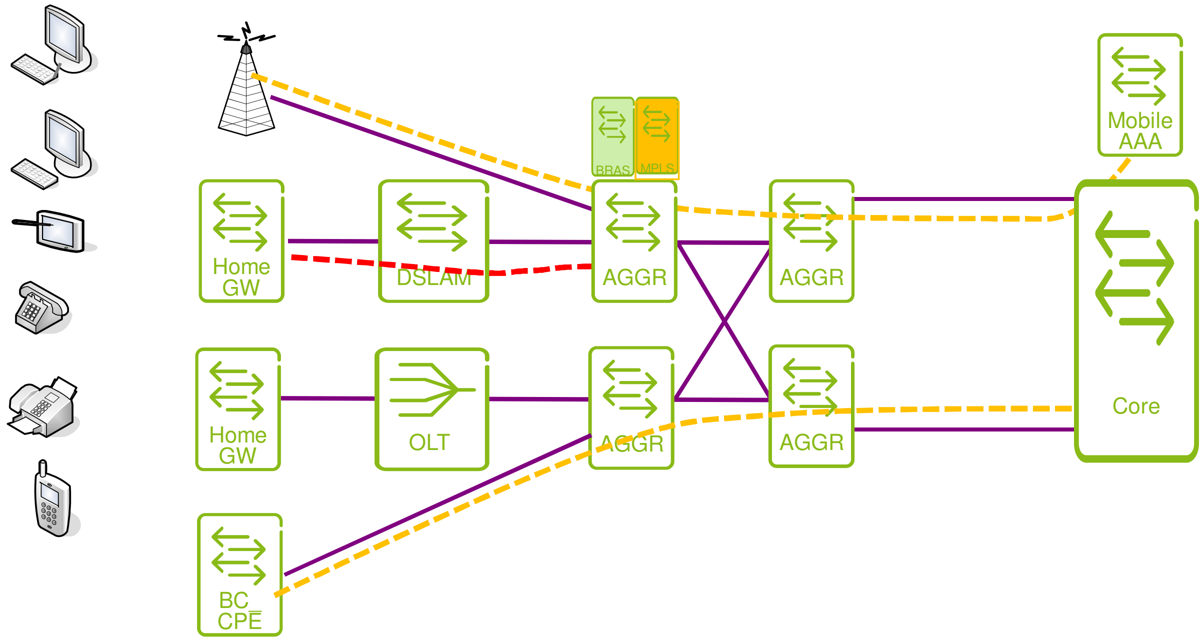


Can split architecture simplify the access / aggregation?
Tunnel config.for mobile backhaul and business customer. Customer management and AAA

SPARC

Access Aggregation Scenarios.

Split Architecture.



What if we could turn any AGGR switch to a "soft" BRAS

What if we could turn any AGGR switch to a „soft" MPLS-router

SPARC

Questions.

- How do you define "Software Defined Networking", "Network Programmability" and/or "Flow Switching"? Where do you see the main advantages e.g. of "OpenFlow"?
  - Universal
    - Vendor agnostic programming interfaces
    - Vendor agnostic network application integration
    - Hardware abstraction layer
  - Forwarding and processing
    - Flexible, i.e. programmable forwarding / filtering
    - Integration of (flow) processing function
    - Integration of seamless virtualization approaches
  - Decoupling of software and hardware innovation cycles
  - Maximum of open approaches

- Which are the key components of your novel architecture? (The presentation should be focused on the main items.)

  – Split of forwarding and control
  – Inclusion of flow processing functions and circuit switches (optics)
  – Hardware and vendor agnostic API
  – Recursive controller architecture

- Which novel functionality is provided? In particular, what application is enabled by the technologies developed in your project beyond what is possible today?

  – Carrier class functions on top of the „academic“ open flow
    - OAM
    - MPLS
    - BFD based link/tunnel monitoring
    - Resilience
  – Open flow has an inherent “packet inspection functionality”
    - Enables new approach for network AAA functions
    - Easy implantation of PPPoE and DHCP, easy migration
  – Proposals to expand the existing specification

Questions.

- To which extent do the technologies developed in your project improve application performance? Are there any benchmarking results available? What is the overall experience with your approach?
  - Success: less complex realization of software
    - MPLS
    - Multicast
    - PPPoE
  - Easy adaptation of network functions ,
  - Easy implementation of network functions
  - Simplified service migration
  - Larger software developer community for network applications

Questions.

- Which are the threats for not adopting the technologies developed in your project?
  - Network community goes for another network operation systems
  - Energy consumption might be higher than for specific devices
  - Fragmented approaches / standardization
  - Less important open software community

Questions.

- How can the technologies of your project inter-work with the other technologies presented in the cluster workshop?
  - International cooperation
    - Stanford
    - GENI
    - Japan
  - Already a lot of cooperation with
    - OFELIA
    - Change

SPARC

SPARC – **Split Architecture**

Applications of SDN in Carrier Networks

Andreas Gladisch
Deutsche Telekom AG, Innovation Laboratories

SPARC

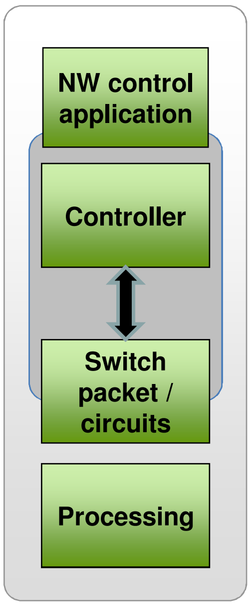
Agenda

- Overview
- Questions
  1. How do you define "Software Defined Networking", "Network Programmability" and/or "Flow Switching"? Where do you see the main advantages e.g. of "OpenFlow"?
  2. Which are the key components of your novel architecture? (The presentation should be focused on the main items.)
  3. Which novel functionality is provided? In particular, what application is enabled by the technologies developed in your project beyond what is possible today?
  4. To which extent do the technologies developed in your project improve application performance? Are there any benchmarking results available? What is the overall experience with your approach?
  5. Which are the threats for not adopting the technologies developed in your project?
  6. How can the technologies of your project inter-work with the other technologies presented in the cluster workshop?

SPARC

Essential of Split Architecture. OpenFlow.

- OpenFlow is an architectural platform for
  - Unified control plane for packet and circuit networks
  - Simplified network control and management
  - Fostering innovative change
- In the US first use for academic testbed facilities
- First commercial offering of OpenFlow based systems yet
- OpenFlow is not yet carrier-grade and needs extensions
  - Optical and wireless technologies
  - Extend filter format description to generic labels
    - MPLS, Carrier Ethernet, IPv6, Optical circuits
  - Resilience, redundancy
  - Network operation, OSS



How can carrier-specific formats and OAM be included?
What awareness of processing should be introduced in OF?

- iPOP 2011 in Kawasaki, Japan
  - MPLS Demo
- Future Network & Mobile Summit in Warsaw, Poland
  - MPLS Demo
- 10th GENI Engineering Conference in Puerto Rico, USA (2011)
  - MPLS Demo
- FIA in Budapest, Hungary (May 2011)
  - MPLS Demo (Ericsson booth)
- Open Networking Summit 2011 (Stanford)
  - Joint booth with OFELIA (PPPoE)
- FIA in Poznan, Poland (October 2011)
  - Joint booth with OFELIA (MPLS and PPPoE)
- Open Networking Summit

Is OpenFlow the Future Internet?

If not, will it at least help building it?

Hagen Woesner,
EICT GmbH, Berlin
Coordinator, OFELIA project

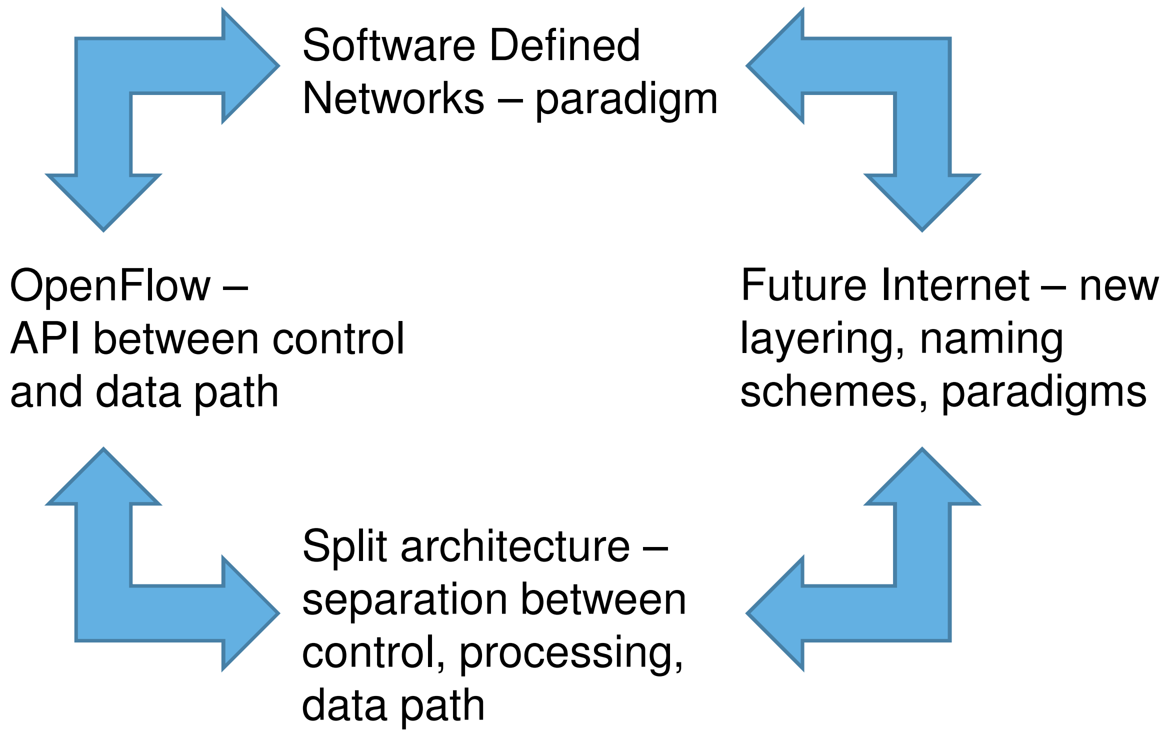
Contents

- SDN, OF, Split Architecture, Future Internet
- How can OpenFlow help FI research?
  – And what is missing in OF?
- New contributions:
  – Recursive control plane architecture
  – Name space reservation → virtualisation
- Relation to OFELIA

The list of questions.

- 1 How do you define "Software Defined Networking", "Network Programmability" and/or "Flow Switching"? Where do you see the main advantages e.g. of "OpenFlow"?
- 2 Which are the key components of your novel architecture? (The presentation should be focused on the main items.)
- 3 Which novel functionality is provided? In particular, what application is enabled by the technologies developed in your project beyond what is possible today?
- 4 To which extent do the technologies developed in your project improve application performance? Are there any benchmarking results available? What is the overall experience with the your approach?
- 5 Which are the threats for not adopting the technologies developed in your project?
- 6 How can the technologies of your project inter-work with the other technologies presented in the cluster workshop?

Software Defined Networks, OpenFlow, Split Architecture, Flow Switching, and Future Internet



What is Future Internet research aiming at?

- Overcome the deficiencies of the current Internet:
  - "naming the wrong things": we *mean* a node when we *address* its network attachment point
    - multi-link operation and mobility becomes very difficult
    - Shouldn't we name something else, like: information?
  - "resolving the wrong names at the wrong place"
    - DNS should not be named with an IP address (this is its address, not name), v4-v6 transition becomes difficult
  - "wrong layers in our model"
    - The multi-layer that we see are not ISO/OSI!
- New naming/resolution schemes (e.g., CCN), new layerings (e.g., RNA)
- FP7 4WARD (and FIND, AKARI) addressed pretty well the problems, even delivered some solutions

…and how does OpenFlow help?

- OpenFlow defines matches and actions
  - Extensibility is key for real 'future Internet' naming schemes
    - OpenFlow is supposed to add extensible match in v1.2
  - Extensible action still to come / under discussion
    - Requires hardware abstraction layers (HAL) below OpenFlow
- OpenFlow lets us experiment with new technologies on real hardware
  - New routing protocols
  - New matches/actions (eventually)
- Still open how stateful actions can materialize
  - Per-packet state in the hardware is generally considered bad
    - but it's there! C.f. tunnels or QoS (PPPoE, GTP, LSP)

Key components of the new architecture

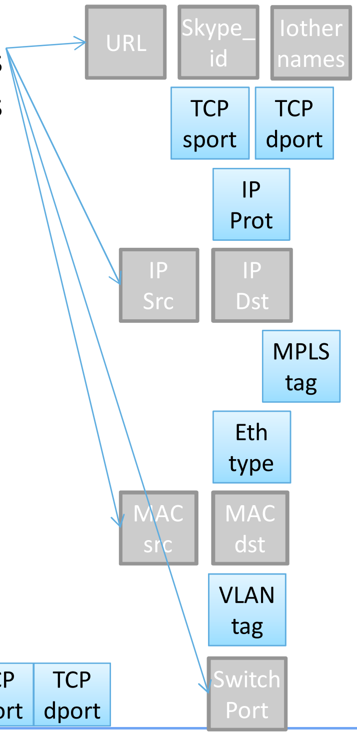
- There are several projects in the OpenFlow area, our contribution comes from OFELIA, SPARC, local Berlin projects.
  – Recursive network architecture (c.f., RNA or RINA, J. Touch, J. Day) – translated to Split architecture, building on OpenFlow
    - Recognizing *Patterns in network architecture* - FI
    - Recognizing *flows* – OF
    - Slicing the network – FlowVisor integration into controller

Base class of an OpenFlow controller containing

- Access control protocol (e.g., naming/addressing, endpoint creation)
- Resource information exchange protocol (e.g., Routing)
- Error and Flow control protocol (i.e. endpoint control, e.g., ioctl)
- Match/action control protocol (i.e. OpenFlow)
  – Including reservation of name spaces (slicing)

Names and ports

- The packet header contains
  – Names
  – Ports



- Names identify entities in a domain defined by
  - Name space
  - Administrative boundaries
- Ports identify nothing.
  – They *point* from an entity in one compartment to another entity in another compartment
- Flows are identifiable by the combination of <name, d_port>, where d_ port is the one leading *to* the named entity (from “below”)
  – Example: LSP_id:=<dest_IP, label>
  – Any OpenFlow flowspace

Switch Port	MAC dst	MAC src	Eth type	VLAN ID	VLAN prio	MPLS label	MPLS Tr_cls	IP src	IP dst	IP Prot	TCP sport	TCP dport

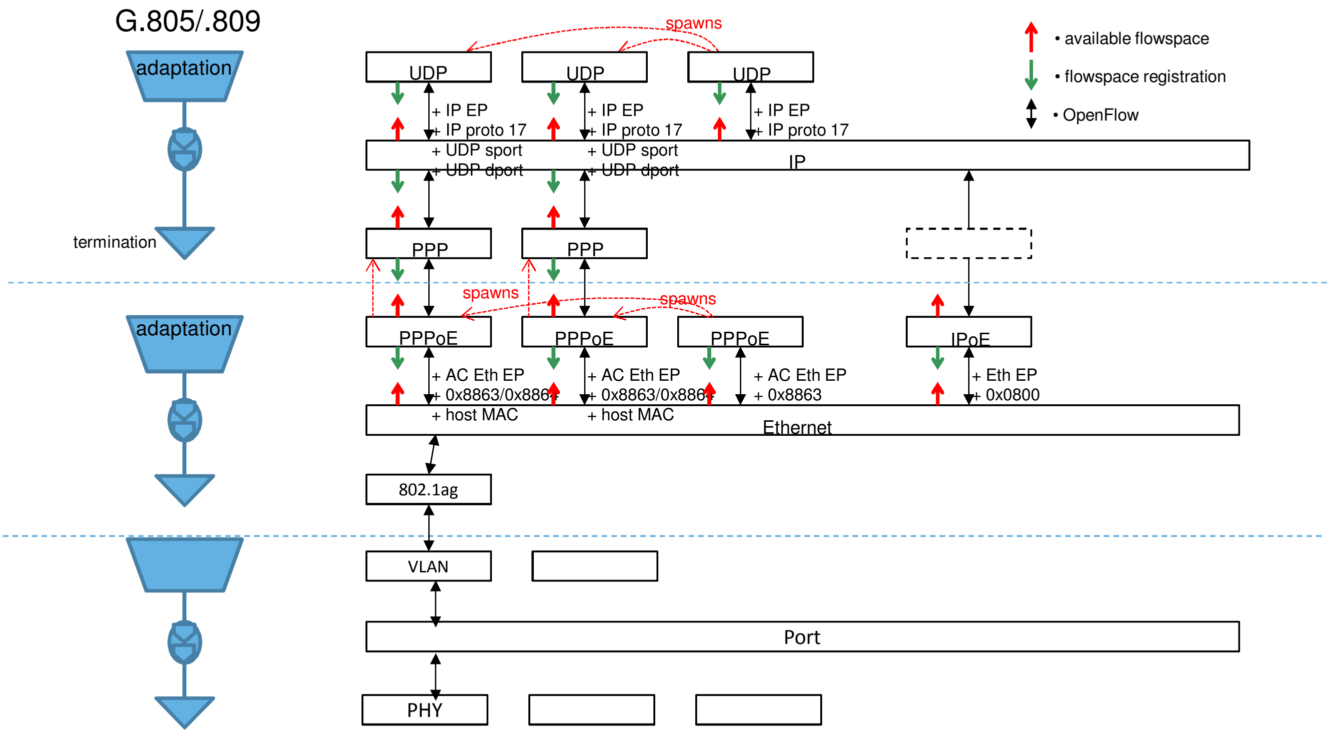
OpenFlow 1.1 match fields

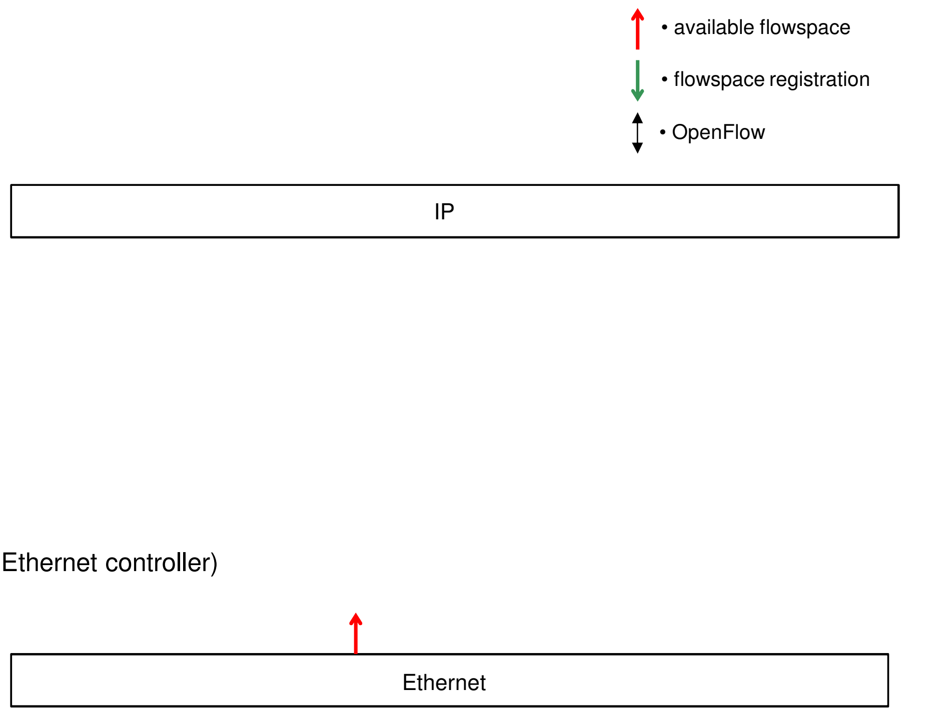
Virtualisation of networks

- In a split architecture, all components have to be virtualised (or partitioned?)
  – Control
    - FlowVisor virtualises the control path
      – The bandwidth between switch and controller(s)
      – The processing capacity of the controller(s)
      – The flow space
  – Forwarding
    - Slices may virtualise the backplane
      – TCAMs, counters, action tables, flow tables
  – Processing
    - The bandwidth at the switch ports, i.e. the link capacity
    - The network sources and sinks

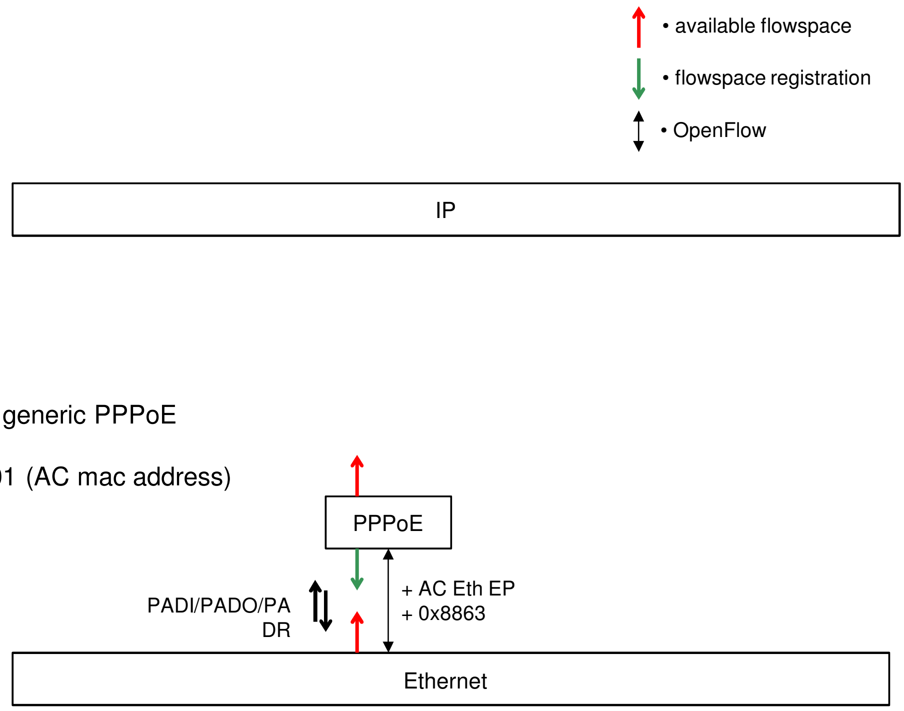
Should flow space be *virtualized* or *partitioned*?

Example of what is possible

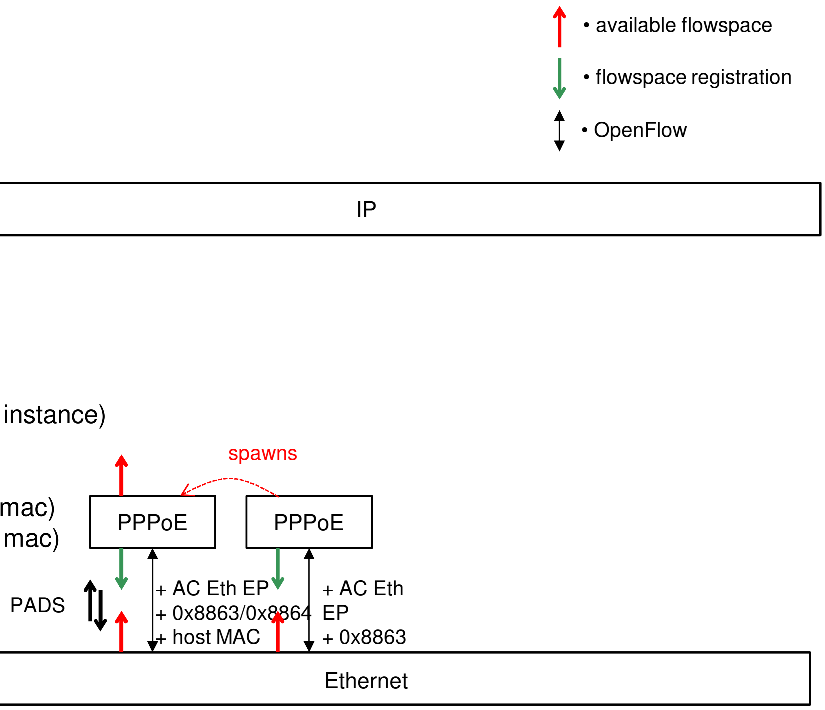




1 **Flowspace** (available by Ethernet controller)
➔
- all wildcard

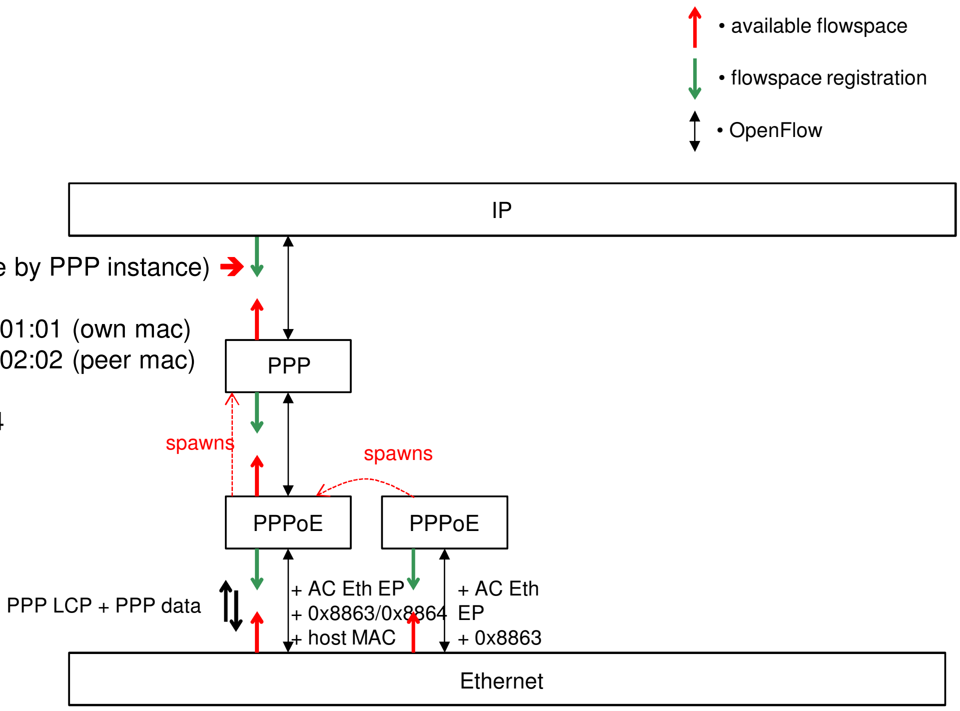


2 **Flowspace** (available by generic PPPoE instance) ➔
- dl_src: 01:01:01:01:01:01 (AC mac address)
- dl_type: 0x8863



3 **Flowspace** (available by PPPoE instance) ➔
- in_port: 5
- dl_src: 01:01:01:01:01:01 (own mac)
- dl_dst: 02:02:02:02:02:02 (peer mac)
- dl_type: 0x8864
- pppoe_sessid: 1234

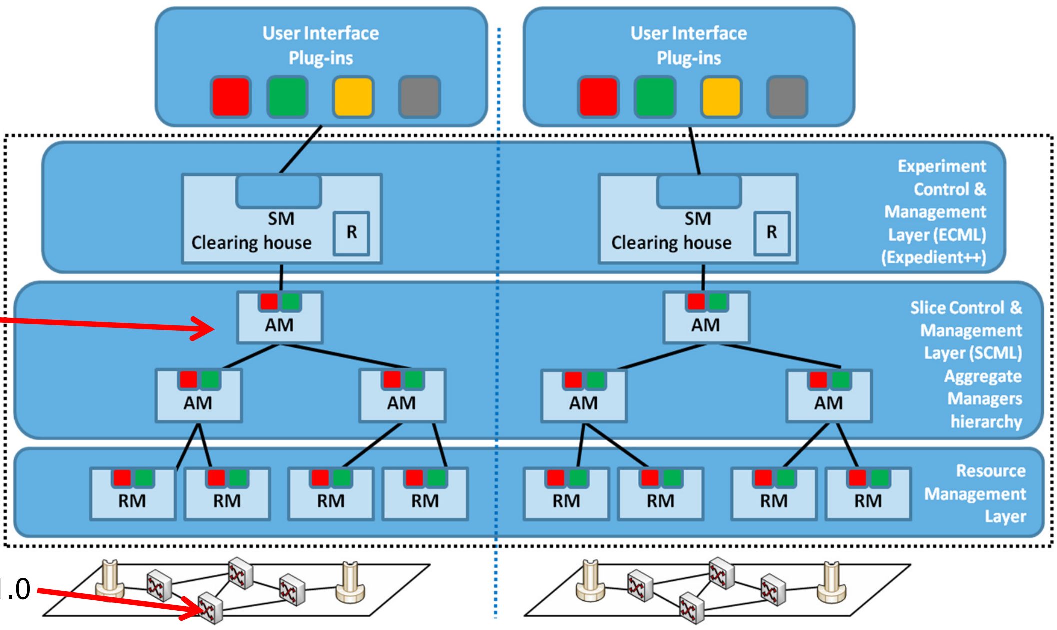
4 Flowspace (available by PPP instance)
- in_port: 5
- dl_src: 01:01:01:01:01:01 (own mac)
- dl_dst: 02:02:02:02:02:02 (peer mac)
- dl_type: 0x8864
- pppoe_sessid: 1234
- ip src: 1.2.3.4



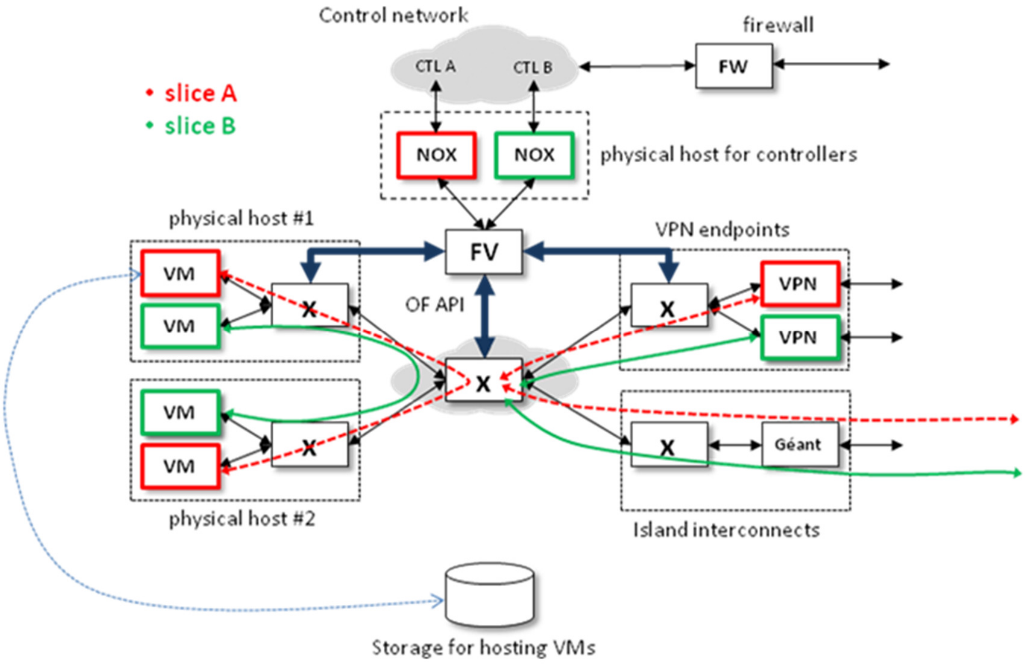
Any relation to OFELIA?

Recursive Layering of AMs
- Need to slice and reserve

Deploy functionality beyond OpenFlow 1.0



Virtualisation in OFELIA – is a VLAN enough?



Conclusions

- OpenFlow may help FI research and experimentation if extended properly
  – Flexible match and flexible actions
  – Hardware abstraction layer
  – Carrier extensions for OAM
- Network Operating System based on OF requires
  – Modularity
  – Open source kernel
  – Virtual machine definition (abstraction layer) for the datapath
  – A supporting community, that could well be European

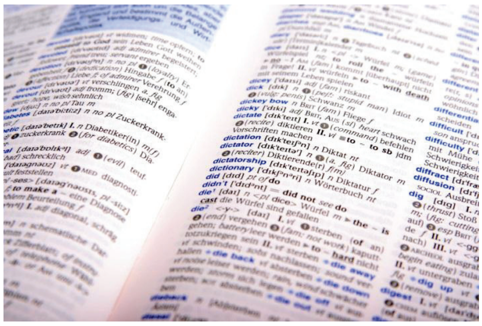
Scalability Issues of SDN on the Example of OpenFlow

Phuoc Tran-Gia, Michael Jarschel, Rastin Pries
www3.informatik.uni-wuerzburg.de

Agenda

- Definition of SDN
- OpenFlow in G-Lab
- OpenFlow Scalability
- Summary

Definition of “SDN”



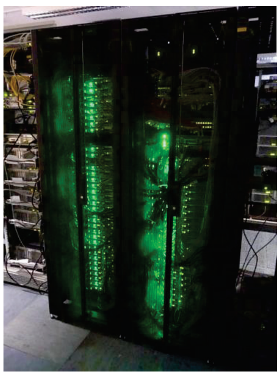
- What does “SDN” stand for ?
  - Software-**Defined** Networking (ONF)
  - Software-**Driven** Networks (IETF SDN)
- Definition SDN (IETF SDN)
  - Software **Driven** Networks (SDN) is an approach to networks that enables applications to converse with and manipulate the control software of network devices and resources.
- Our definition attempt for SDN
  - Software-**Defined** Networking is a networking paradigm in which the forwarding behavior of a network element is flexibly determined by a software control plane decoupled from the data plane.

General Information about G-Lab

www.german-lab.de

SPONSORED BY THE Federal Ministry of Education and Research

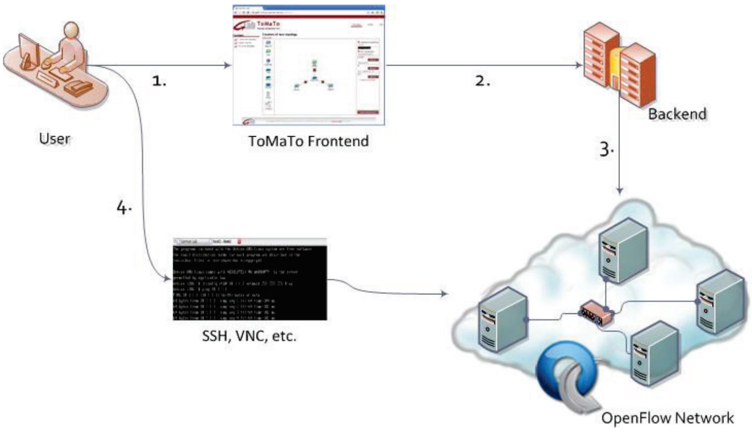
- National Future Internet project in Germany
- Funded by the Federal Ministry of Education and Research
- Currently 32 partners from
  - Academia
  - Research institutes
  - Industry



- Typical site consists of:
  - 22 “normal” nodes (2x4 CPU, 16 GB RAM, 4x146 GB HDD)
  - 2 Netnodes (additional NICs)
  - 1 Headnode (Faster CPU, 12x146 GB HDD)

Current Status of OpenFlow in G-Lab

- Separate OpenFlow Network
- Topology and machines are provisioned through G-Lab “ToMaTo”
- Control of the machines is available through SSH connections and/or VNC



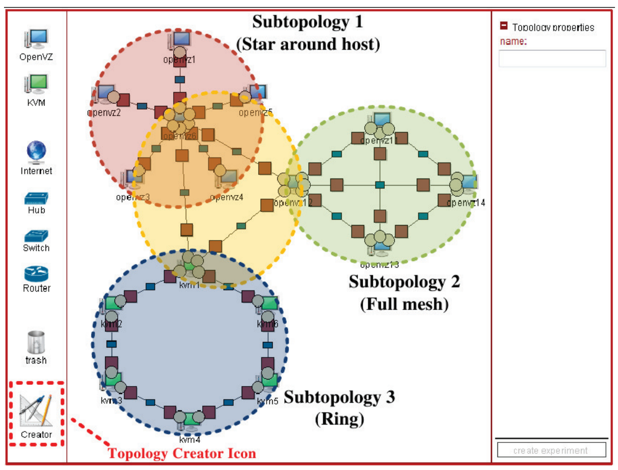
- OpenFlow Network is sliced using a (manually configured) FlowVisor
- OpenFlow controllers are not provided

G-Lab “ToMaTo”

ToMaTo: „Topology Management Tool“

https://tomato.german-lab.de

- Developed and maintained by G-Lab SIG-Facility (D. Schwerdel, TU Kaiserslautern)
- Allows researchers to design own topologies for network experiments

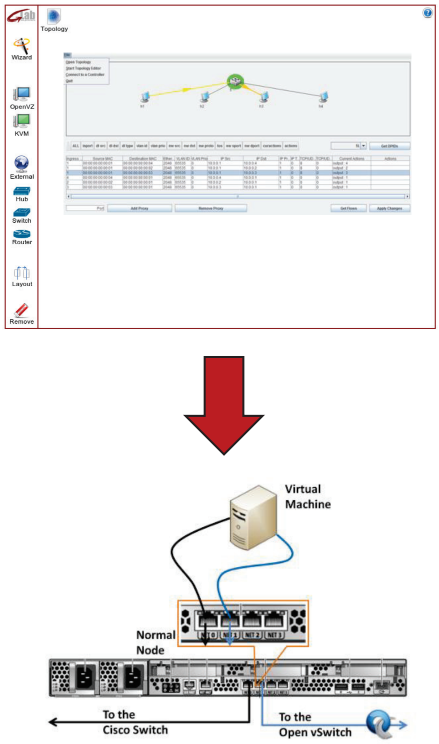


- Experimenter has influence on devices and connectors
- Configuration via easy-to-use web-frontend

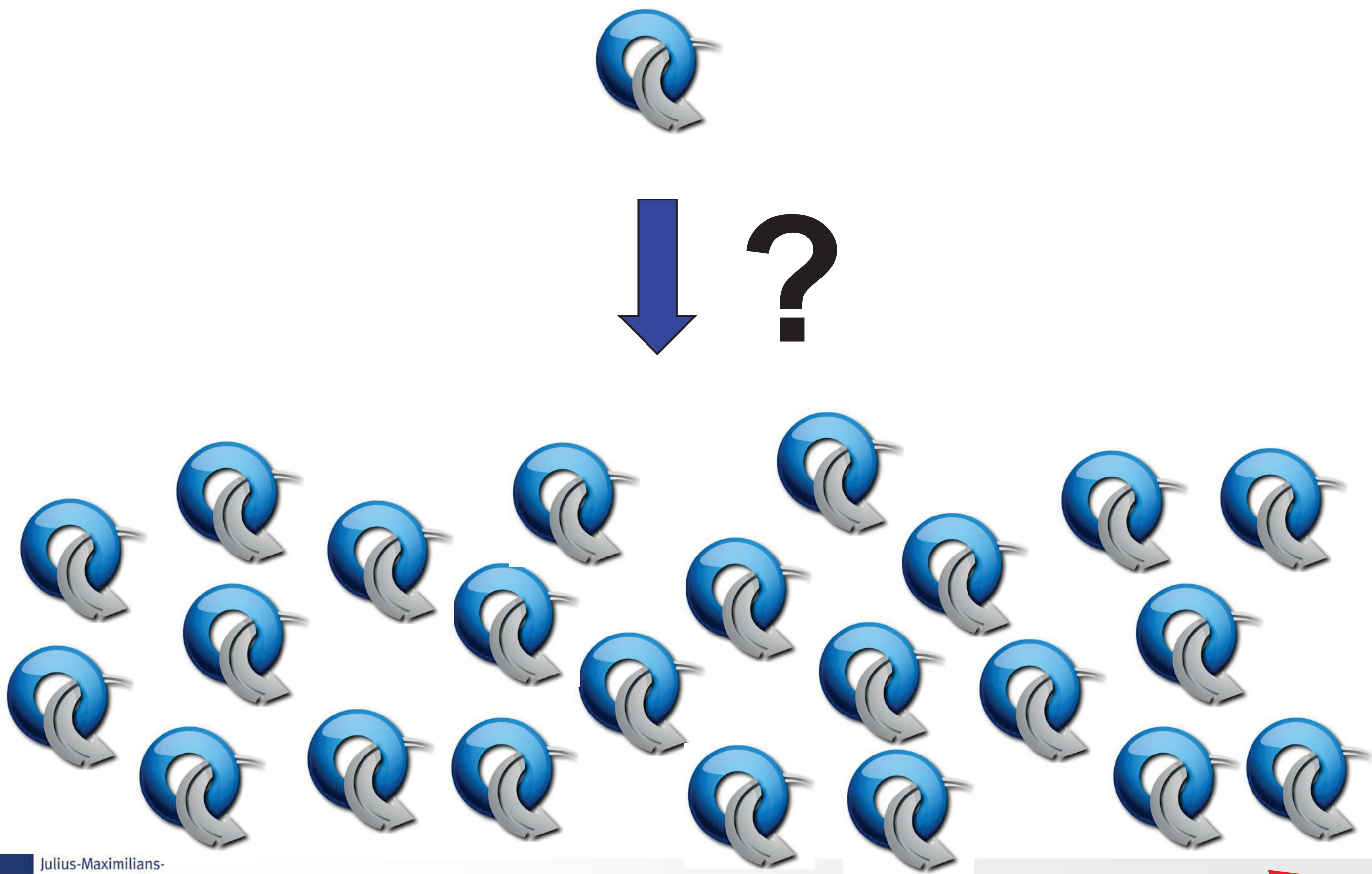
Future of OpenFlow in G-Lab

G-Lab ToMaTo Topology management tool

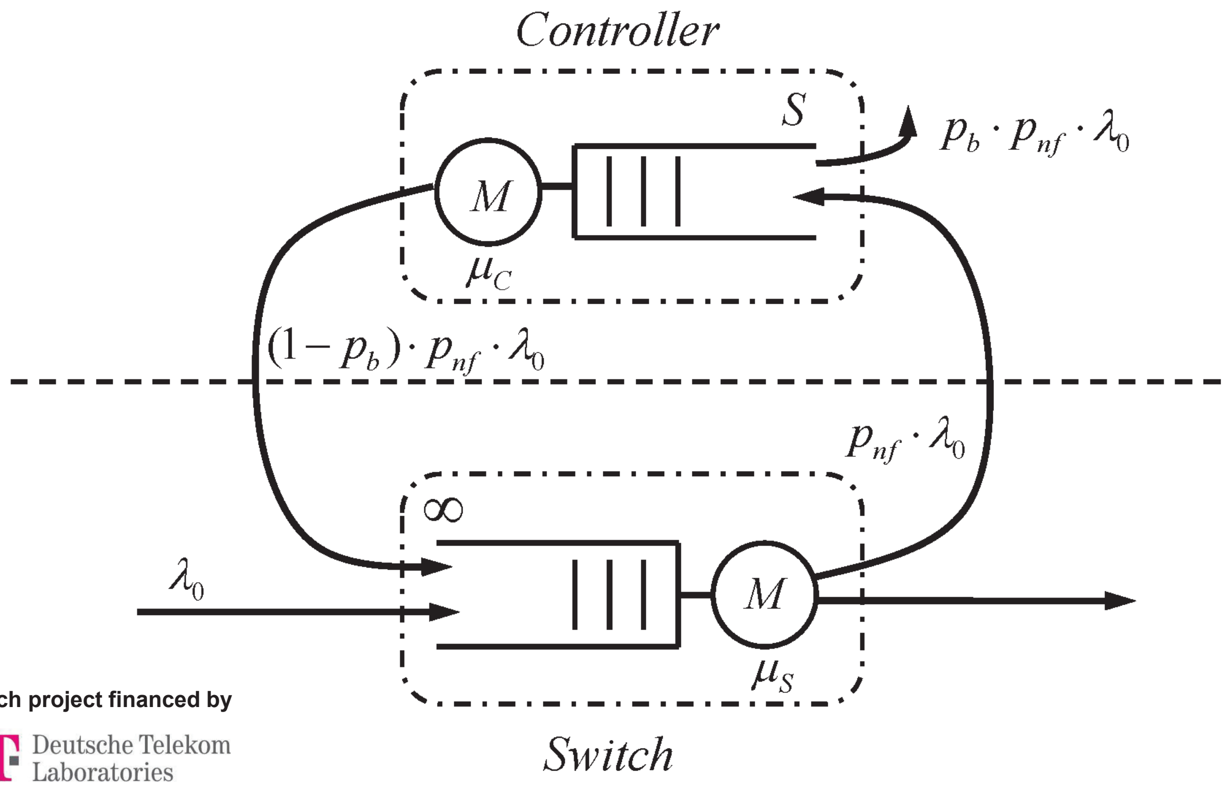
- Provisioning of OpenFlow controllers
- Integration of controller functionality into ToMaTo
- Flow Monitoring with ToMaTo
- Integration of existing OpenFlow projects, e.g. RouteFlow, FlowScale, …
- Evaluation of alternative slicing concepts



OpenFlow Scalability



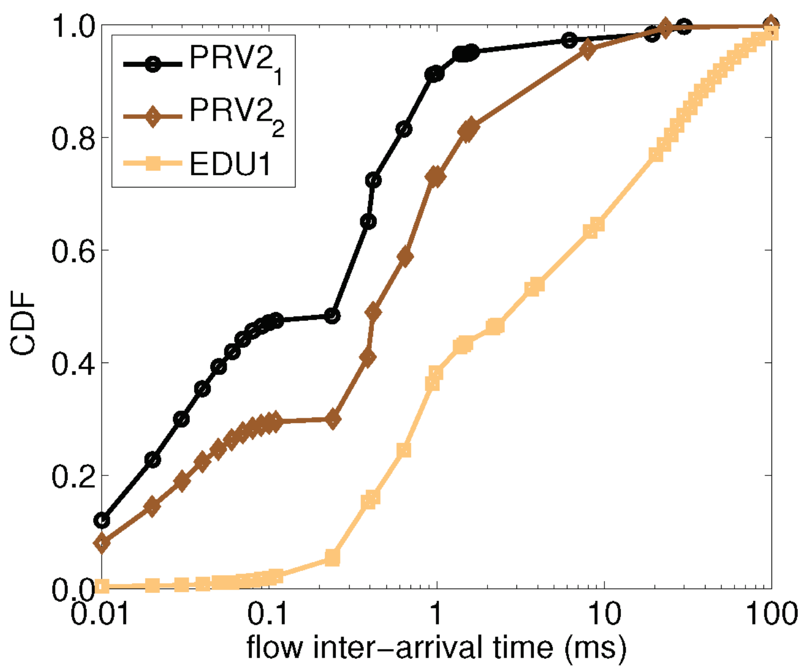
Simplified OpenFlow Switch-Controller Model



Research project financed by

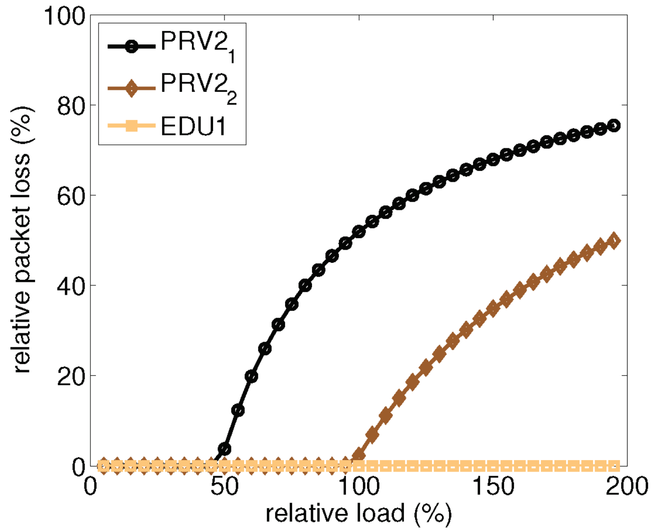
Deutsche Telekom Laboratories

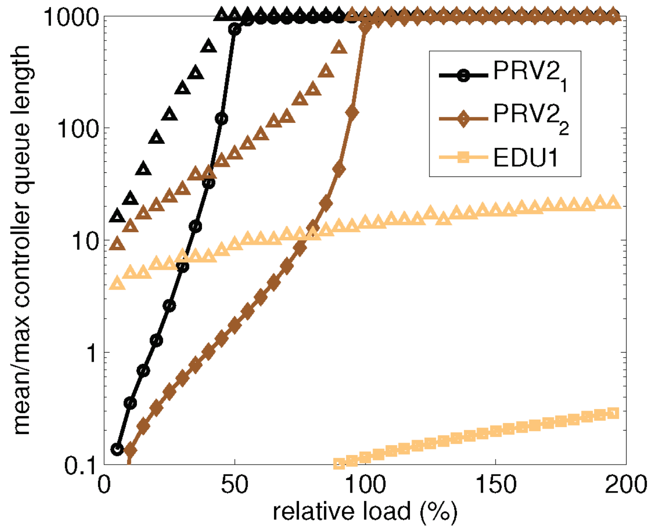
Data Center Traffic (Benson et al.)



- **Network traffic characteristics of data centers in the wild (Benson et al., IMC 10)**

Simulation Results





- **On the Usability of OpenFlow in Data Center Environments (Pries et al., ICC'12 WS - CloudNetsDataCenters)**

Ongoing Research

- OpenFlow simulation model
  - Construction of a detailed OpenFlow simulation model
- OpenFlow controller benchmark
  - Distributed
  - Configurable
- Generation of flow rules using machine learning
  - Using monitoring data as training set
- Using SDN for reducing the energy consumption
  - In data centers
  - In the core

Questions and Comments ?

Wireless MAC Programmability: architectural insights

With proof-of-concept validation on commodity HW

FP7-FLAVIA project

Giuseppe Bianchi

CNIT/Univ. Roma Tor Vergata - giuseppe.bianchi@uniroma2.it

Work contributors: **I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli**

The quest for wireless flexibility

- **Dynamic spectrum access**
  - Adapt access mechanisms to context
- **Service-specific, niche scenarios**
  - Hardly worth development costs
  - Sometimes ULTRA specific requirements (e.g. home networks)
- **New PHYs, new technologies**
  - 802.11n, 802.11ac, 802.11ad, …
  - Directional antennas, MIMO, CoMP, …
- **New applications, services, deployment scenarios**
  - vehicular, ad hoc, mesh, DTN, …
  - X-layer, SVC support, QoE, …
- **Don’t worry: LOTS of brilliant solutions**
  - … on paper… deployment is THE issue!

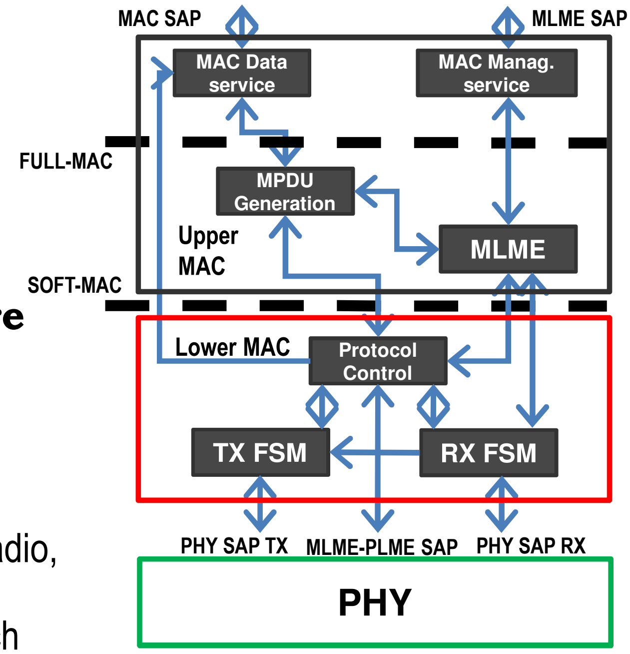
Wireless programmability
(a biased view towards WLAN)

➔ **Past: fullMAC**
- ⇨ Very little flexibility

➔ **Today: softMAC**
- ⇨ Only Upper MAC, no protocol ctrl

➔ **Sneaked out: open firmware**
- ⇨ Probably only openFWWF…
- ⇨ not "much" (?!) vendor support

➔ **Elite: DSP/FPGA boards**
- ⇨ WARP, AirBlue, CalRadio, GNURadio, RUNIC…
- ⇨ Cost, performance: just for research



… programmability ≠ open SW

Programmability = abstract, decouple, make it easy

- ➔ **PC analogy: SW ←→ Instruction set**
  - ⇨ *Is MS-Word built-in in the PC?*
    - →MS-Word = SW developed by a vendor
    - →using a different vendor's **CPU instruction set**
  - ⇨ "instruction sets" evolved, significantly!
    - →RISC
    - →Special-purpose computing platforms
      - » GPUs, DSPs, Crypto, Embedded, etc
- ➔ **Rethink wireless stacks:**
  - ⇨ An architecture issue!
    - →Which "instruction sets"?
    - →Which "CPU" operation?
  - ⇨ Little help from today (strictly monolitic / all-in-one)

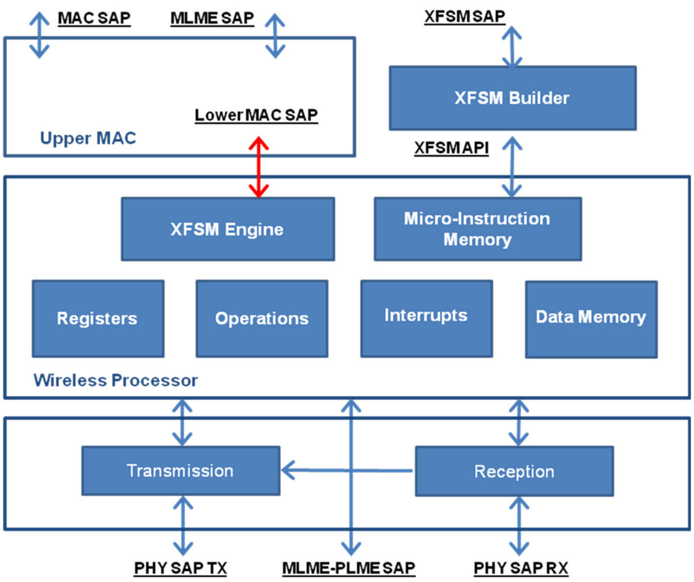
MAC protocols: anatomy

- ➔ **ACTIONS: frame management, radio control, time scheduling, …**
  - ⇨ TX frame, RX frame, set timer, freeze counter, build header, switch channel, etc
- ➔ **EVENTS: available HW/SW signals/interrupts**
  - ⇨ Busy channel signal, RX indication, enqueued frame, end timer, etc
- ➔ **CONDITIONS: boolean/arithmetic expressions on available registers/info**
    - →Frame address, queue length, ACK received, power level, etc
- ➔ **Prococol control LOGIC: orchestrates all this**

Wireless Processors

- ➔ **"instruction set":**
  available actions/events/conditions
  - ⇨ *vendor pre-implemented* "building blocks"
  - ⇨ Closed, not (necessarily) programmable
    - →vendor's core domain, rarely change (e.g. new PHY)
- ➔ **MAC "CPU":**
  engine executing a MAC protocol logic
  - ⇨ MAC protocol agnostic, just execute user-defined protocol logic
  - ⇨ *Vendor pre-implemented*, in principle might never change
- ➔ **Our proposal for Wireless Processor engine:**
  generic XFSM executor
  - ⇨ Most standardized MAC are designed as such

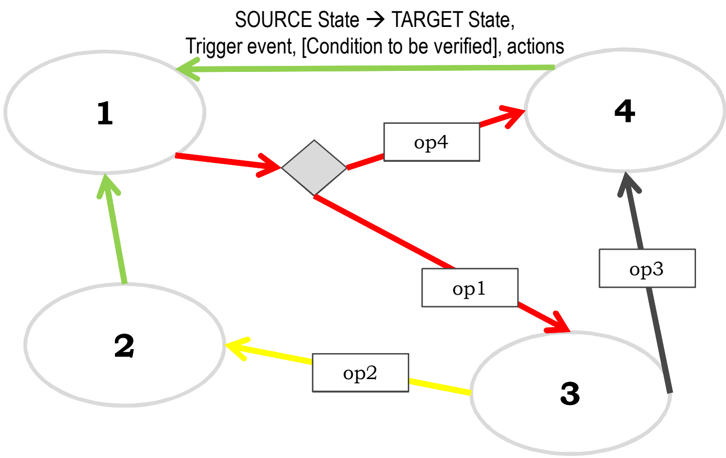
Wireless Processor Architecture



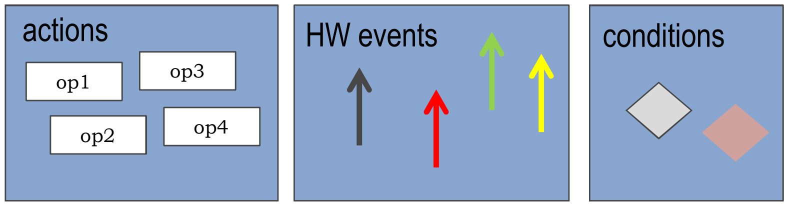
- ➔ **<u>MAC Engine:</u> XFSM executor**
- ➔ **<u>Memory blocks:</u> data, prog**
- ➔ **<u>Registers:</u> save system state (*conditions*);**
- ➔ **<u>Interrupts block</u> passing HW signals to Engine *(events)*;**
- ➔ **<u>Operations</u> invoked by the engine for driving the hardware (*actions*)**

Giuseppe Bianchi

MAC Program as XFSM

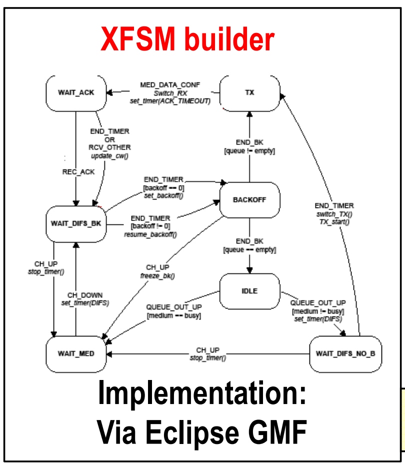


	XFSM formal notation	MAC engine meaning
S	symbolic states	MAC protocol states
I	input symbols	triggering *events*, e.g., hardware signals, timer expiration generated by the interrupt block, etc.
O	output symbols	MAC *actions*: commands acting on the hardware, performed by atomic functions either native in the device or implemented in the pre-loaded operations module (including arithmetic and logic operations, data creation and deletion, etc)
D	n-dimensional linear space $D_1 \times \cdots \times D_n$	all possible settings of n *configuration registers*
F	set of enabling functions $f_i : D \to \{0, 1\}$	set of *conditions* to be verified on the configuration registers for enabling the transitions
U	set of update functions $u_i : D \to D$	*configuration commands* devised to change the value of the configuration registers
T	transition relation $T : S \times F \times I \to S \times U \times O$	indicates the target state, the MAC commands and the configuration commands to be associated to each transition

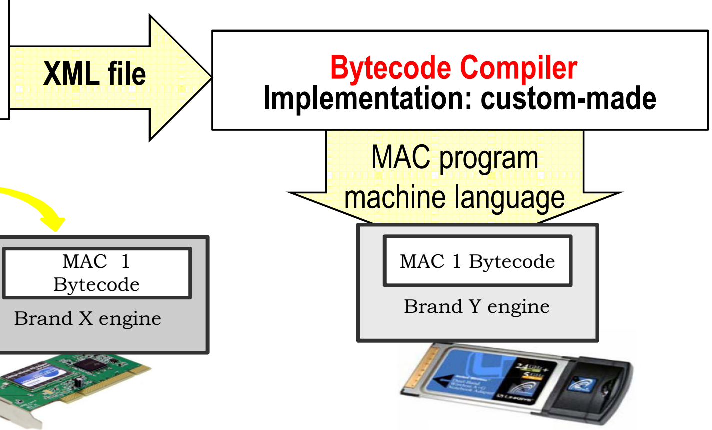


MAC Programming Interface

MAC programs



- ➔ **Wireless MAC processors**
  - ⇨ Special-purpose processor for executing a MAC protocol
  - ⇨ integrated in the NIC, real-time handling of NIC resources
- ➔ **MAC Programs: "bytecode", dynamically loaded**
  - ⇨ MAC programs = Machine-coded XFSM
  - ⇨ Dynamic injection possible (and tested)



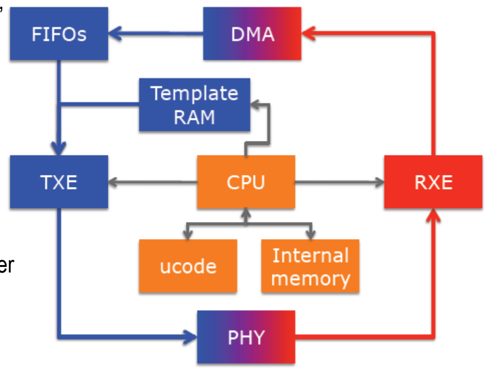
Proof-of-concept

➔ **Only convincing answer (we believe): <u>make it work</u> on a cheap real-world card!**

- ⇨ WARP/FPGA implementation → would have left doubts

➔ **Development platform: Broadcom AirForce54g**

- ⇨ General purpose 8 MHz CPU, 64 registers
  - → Support for arithmetic, logic and binary operations (sum, subtract, xor, shift, …)
  - → Support for basic flow control (routine call, jump if condition, …)
  - → packet queued, plcp end, rx end, rx
- ⇨ Memory: 4kB data, 32 KB code
- ⇨ Packet composition facilities, and RX/TX commands
- ⇨ HW configuration registers for radio resource and event handling
  - → Frequency, power, channel sensing, etc
  - → packet queued, plcp end, rx end, rx correct frame, crc failure, timer expiration, carrier sense, etc



➔ **Development support:**

- ⇨ Native assembly/disassembly tools
- ⇨ Some hands-on ideas (thanks to openFWWF experience and relevant hacks ☺ by Gringoli)

Implementation at a glance

➔ **Delete 802.11 firmware**

- ⇨ Both Broadcom and openFWWF: *we do NOT want a firmware MAC!*

➔ **Replace it with** *[once for all developed]*:

- ⇨ Implementation of actions, events, conditions
  - →in part reusing some openFWWF experience & primitives
- ⇨ XFSM MAC engine executor
  - →Continuous state transition loop:
    - » No events: stay in state
    - » (state-related) event E occurs: trigger state transition and associated action(s)

➔ **Develop "machine language" for MAC engine**

- ⇨ Custom made "bytecode" specified and implemented

➔ **Address several annoying technical hurdles**

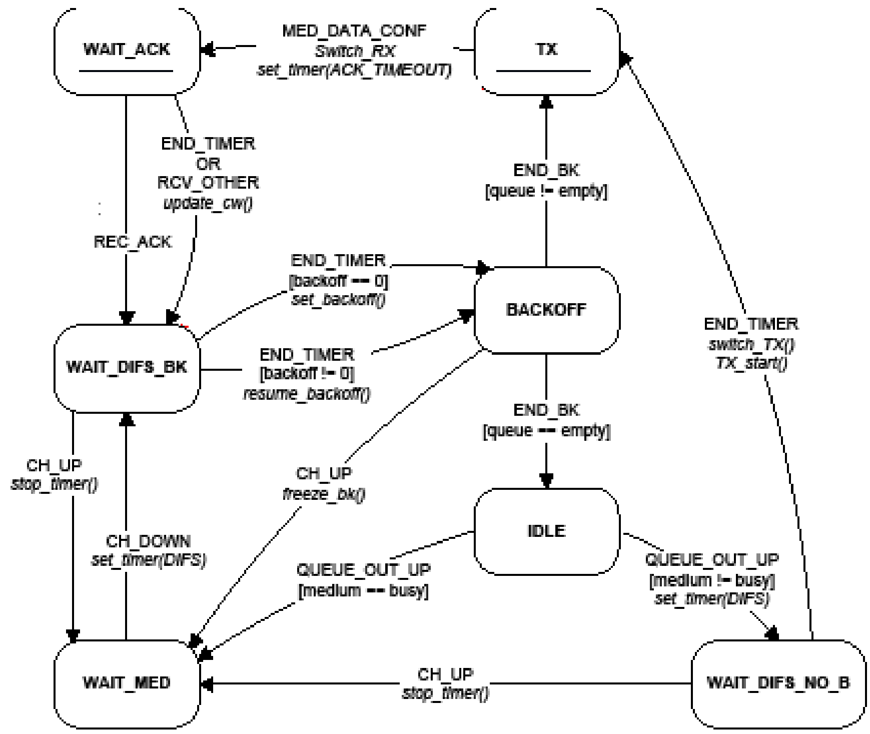
- ⇨ NO direct HW interrupts control available in Broadcom
- ⇨ State and state transition optimizations, …

Actually implemented API

events	*actions*	*conditions*
CH_UP	set/get(reg, value)	dstaddr
CH_DOWN	switch_RX()	myaddr
RCV_ACK	tx_ACK()	queue_length
RCV_DATA	tx_beacon()	queue_type
RCV_PLCP	tx_data()	cw
RCV_RTS	tx_RTS()	cwmin
RCV_CTS	tx_CTS()	cwmax
RCV_BEACON	switch_TX()	backoff
HEADER_END	set_timer(value)	RTS_thr
COLLISION	set_bk()	ACK_on
MED_DATA_CONF	freeze_bk()	srcaddr
MED_DATA_START	update_retry()	frame_type
MED_DATA_END	more_frag()	fragment
QUEUE_OUT_UP	prepare_header()	channel
QUEUE_IN_OVER		tx_power
END_TIMER		

Not yet optimized / to be improved

Standard DCF as XFSM



Actions:
set_timer, stop_timer, set_backoff, resume_backoff, update_cw, switch_TX, TX_start

Events:
END_TIMER, QUEUE_OUT_UP, CH_DOWN, CH_UP, END_BK, MED_DATA_CONF

Conditions:
medium, backoff, queue

DCF-XFSM: same performance (up to 54 mbps max) as i) legacy Broadcom and ii) openFWWF

Functional validation

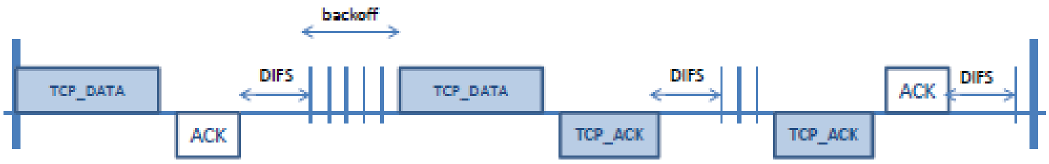
Success IF WMP permits very easy/fast Lower MAC modifications or re-design (vs months or hands-on experience with openFWWF/assembly

- ➔ **Three "scientifically trivial" use cases, tackling distinct MAC aspects recurring in literature proposals**
  - ⇨ Piggybacked ACK
    - →Programmable management of frame replies
  - ⇨ Pseudo-TDMA
    - →Precise scheduling of the medium access times
  - ⇨ Randomized multi-channel access
    - →Fine-grained radio channels control

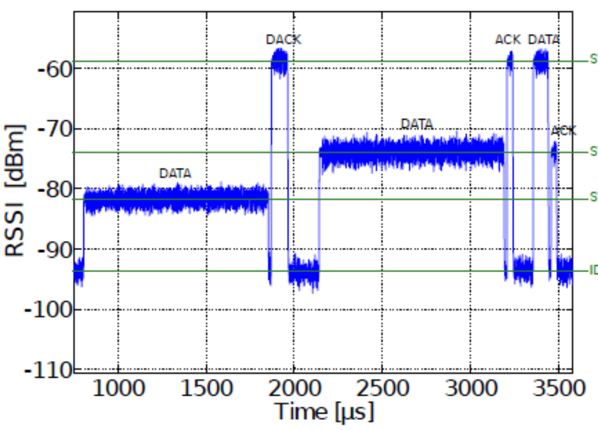
- ➔ **Development time: O(days)**
  - ⇨ Including bug fixing in engine/API, otherwise hours

Piggybacked ACK

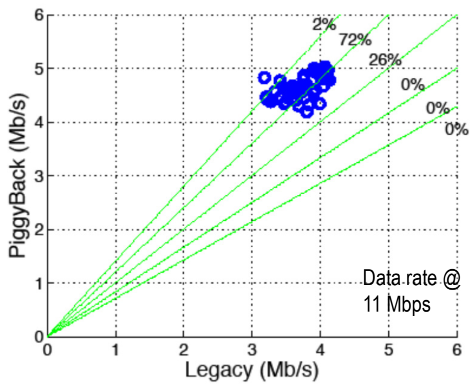
If available, send TCP ACK instead of MAC ACK, otherwise send normal ACK



Channel activity trace



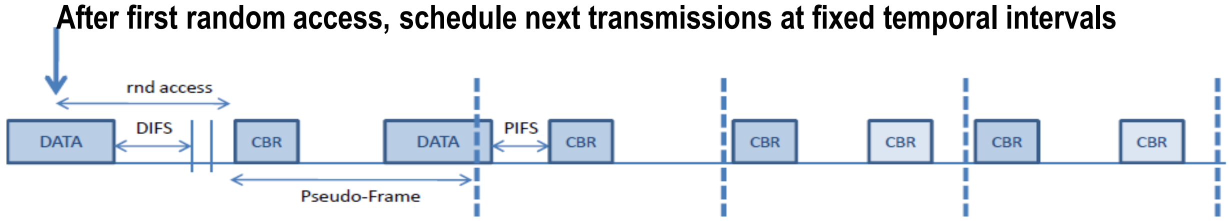
Performance gain

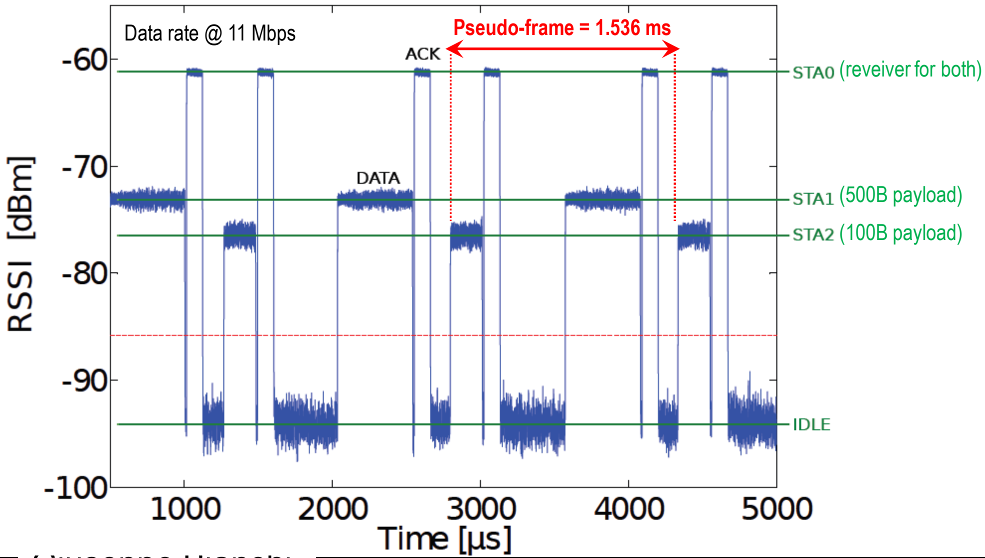


Pseudo-TDMA

[literature proposal]

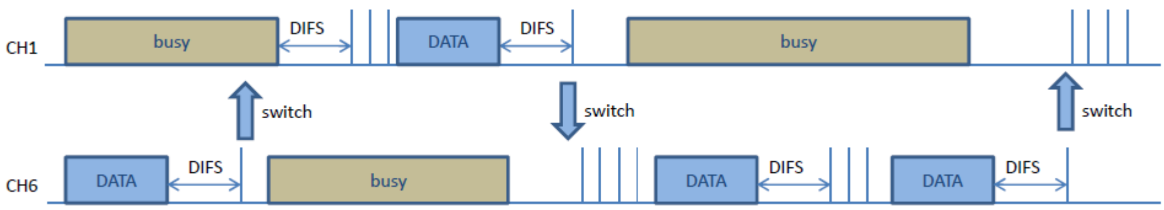
After first random access, schedule next transmissions at fixed temporal intervals





Randomized multichannel access

Per EACH frame, randomly select backoff AND channel (switch on as little as per frame basis)

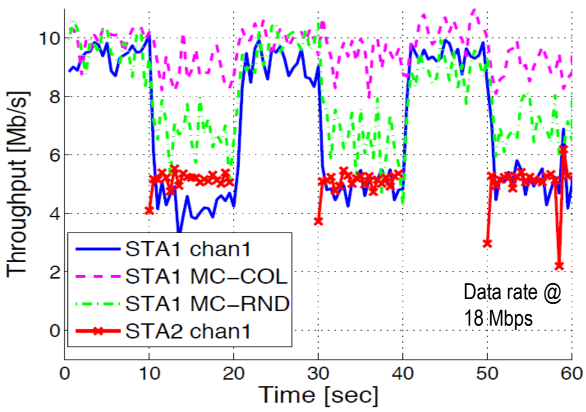


Experimental setup

Two channels:
  C1 = 2412MHz
  C2 = 2457MHz
Two stations scenario:
  STA_A, on/off, 5Mb/s, fixed@2412MHz
  STA_B, persistent, 10Mb/s

Three experiments:
  STA_B fixed @ C1
  STA_B per-frame random hopping
  STA_B random hop after collision

Performance gain



Conclusions

- ➔ **Wireless Processor concept:**
  - ⇨ Promoting a new direction towards wireless stack programmability
  - ⇨ Not (necessarily) requiring open SW/FW/HW
    - →MAC Processor can be a closed product, as any PC processor
- ➔ **Feasibility proof-of-concept on commodity HW**
  - ⇨ Broadcom Airforce54g implementation, thanks to our hands-on
  - ⇨ Most major chipsets have similar internals
- ➔ **What's next?**
  - ⇨ Targeting first public release (april-may?)
    - →Very useful to research community, fast prototyping of ideas
  - ⇨ Challenging WMP limits with a serious MAC (MCCA)
  - ⇨ Core technical extensions (MIMO-aware API, engine virtualization, …)
  - ⇨ Real world exploitation?
    - →Potential for significant changes, if deployed by a chipset maker