UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Recursive Inter-Network Architecture (RINA)

Future Internet Networks (FINE)

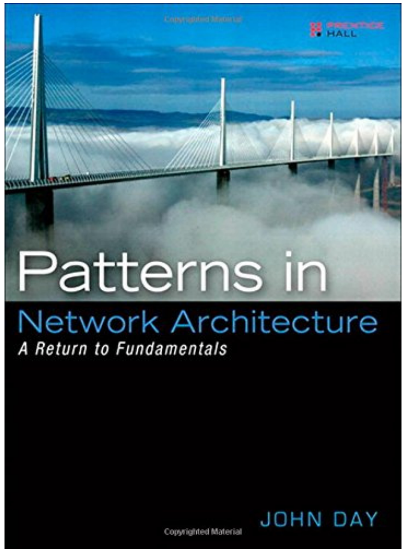
Master in Innovation and Research in Informatics (MIRI)

Jordi Perelló Muntan, academic year 2018-2019

Although not cited every time, many of the figures used in this presentation have been taken from previous works of RINA research community members. So credit goes to them.

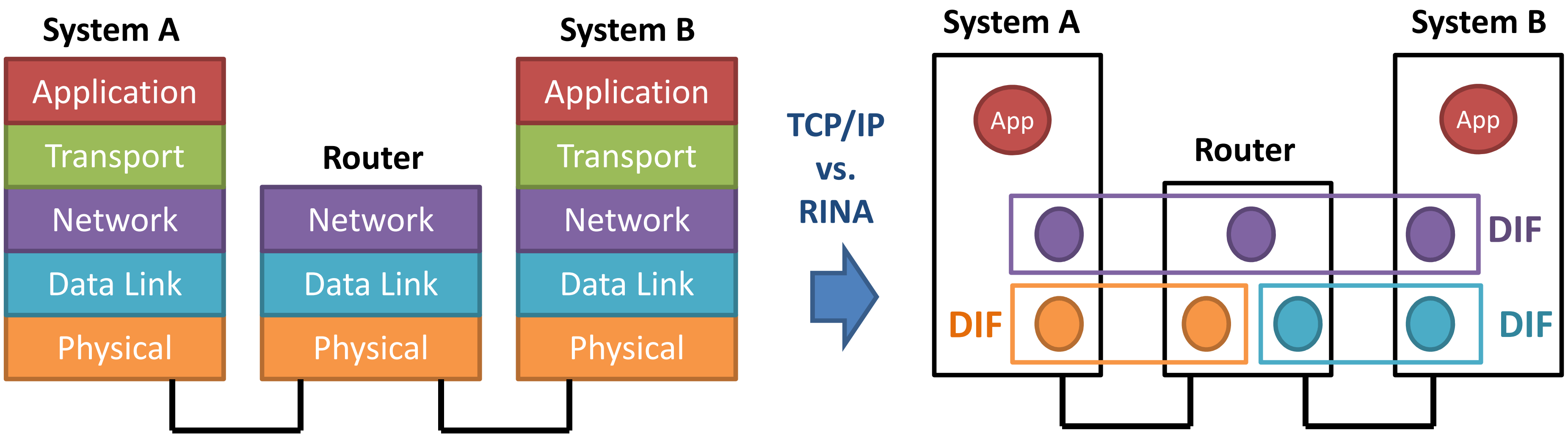
What is RINA? (1/2)

- It is a clean-slate network architecture for the future Internet, proposed by Prof. John Day in his book:
  - “*Patterns in Network Architecture: A Return to Fundamentals*”, Prentice Hall (Ed.), December 2007
- It is referred to as “clean-slate”, meaning that it is not any patch to improve the performance of the current TCP/IP protocols, but a complete redesign from scratch of the Internet
- RINA’s fundamental principle: *“Networking is Inter-Process Communication (IPC) and only IPC”*
- That is, it generalizes the model of local inter-process communications (as in a single computer) to apply it network-wide



What is RINA? (2/2)

- RINA rebuilds the entire structure of the Internet by means of a **single repeating layer**, called **Distributed IPC Facility (DIF)**, hence the recursivity of the architecture
- Each DIF instance implements the **same mechanisms**, whose specific operation can be **customized via programmable policies**, so as to tightly adapt to every network scope (e.g., LAN, DCN, backbone network, etc.)



But why RINA? Recall (some of) the limitations of TCP/IP!

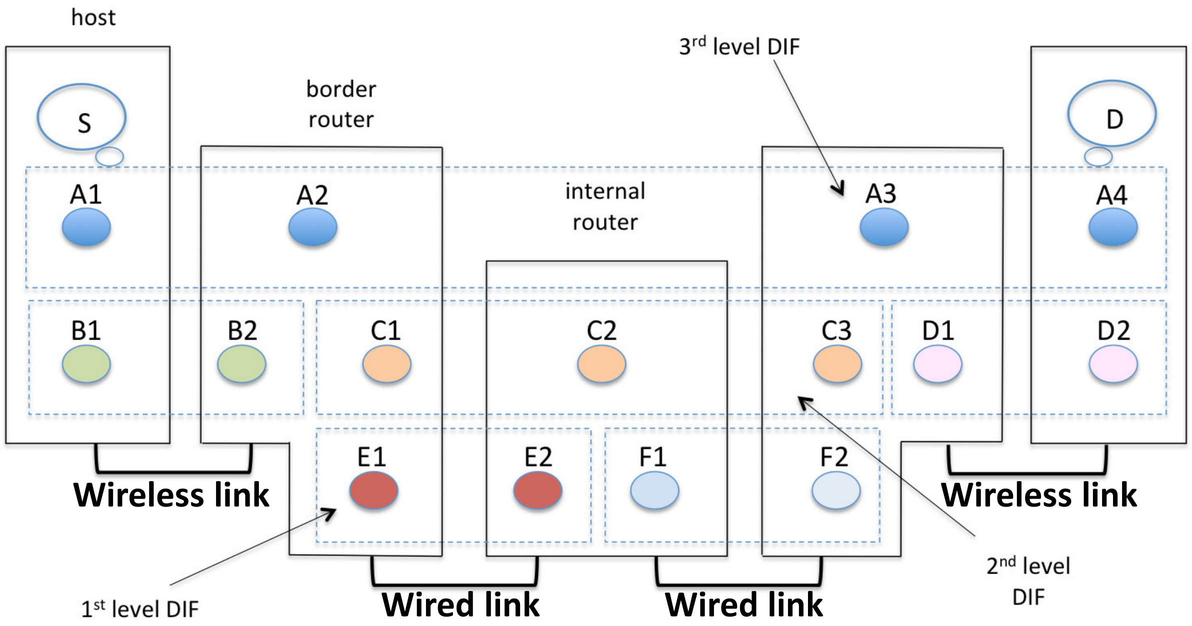
1. Incomplete naming & addressing scheme:
   - Only IP addresses (interface addresses)
   - No way to know if two IP addresses belong to the same host
   - Multi-homing difficult to manage (and mobility too)
2. Global Internet-wide scope of public IP addresses:
   - Poor routing scalability: routing table size explosion!
   - The transition from IPv4 to IPv6 exacerbates the problem even more
3. Applications have no way to express their desired Quality of Service (QoS), rather than reliable (TCP) or unreliable (UDP) communication
4. No built-in security in the Internet architecture
   - Security has to be provided by additional protocols (e.g., IPSec), which complicates the scenario every time a new security threat appears

Conversely, in RINA...

1. A complete naming and addressing scheme is introduced, differentiating between node and interface addresses:
   - This inherently simplifies mobility and multi-homing compared to TCP/IP, where are cumbersome to manage
2. The scope of an address is the DIF. That is, addresses are private inside a DIF and can be repeated in different DIFs when desired:
   - *Divide-and-conquer* approach: the routing scalability problem of TCP/IP is inherently solved by the architecture
3. Each DIF is free to provide its own QoS classes by running programmable routing, resource allocation or data transfer policies. Moreover, apps can request different QoS metrics for their flows (data loss, delay, jitter, etc.)
4. DIFs are securable containers by applying programmable access control policies during the node enrollment (i.e., join) phase

RINA basics (1/6)

- As said before, RINA introduces a **single repeating layer**, called DIF, which offers IPC services over a limited scope:
  - ➢ Just **like a distributed application**, whose members (called IPC processes) are **specialized in providing IPC services to higher layers**
  - ➢ The **protocols at each layer are the same, configured** (i.e., programmed) differently **to fulfill the particular requirements of the layer**

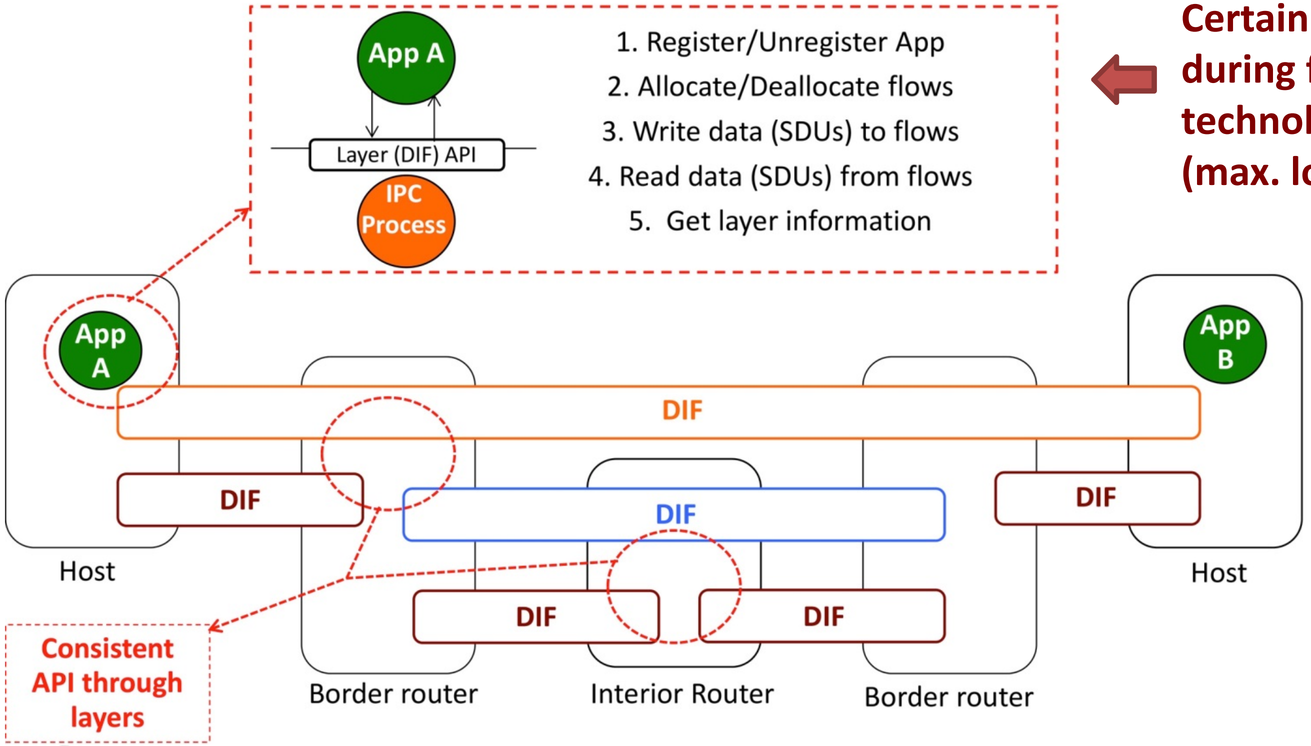


RINA network scenario with 3 levels of DIFs

S,D -> Application Processes (APs)
A1, …, F2 -> IPC Proceses (IPCPs)

RINA basics (2/6)

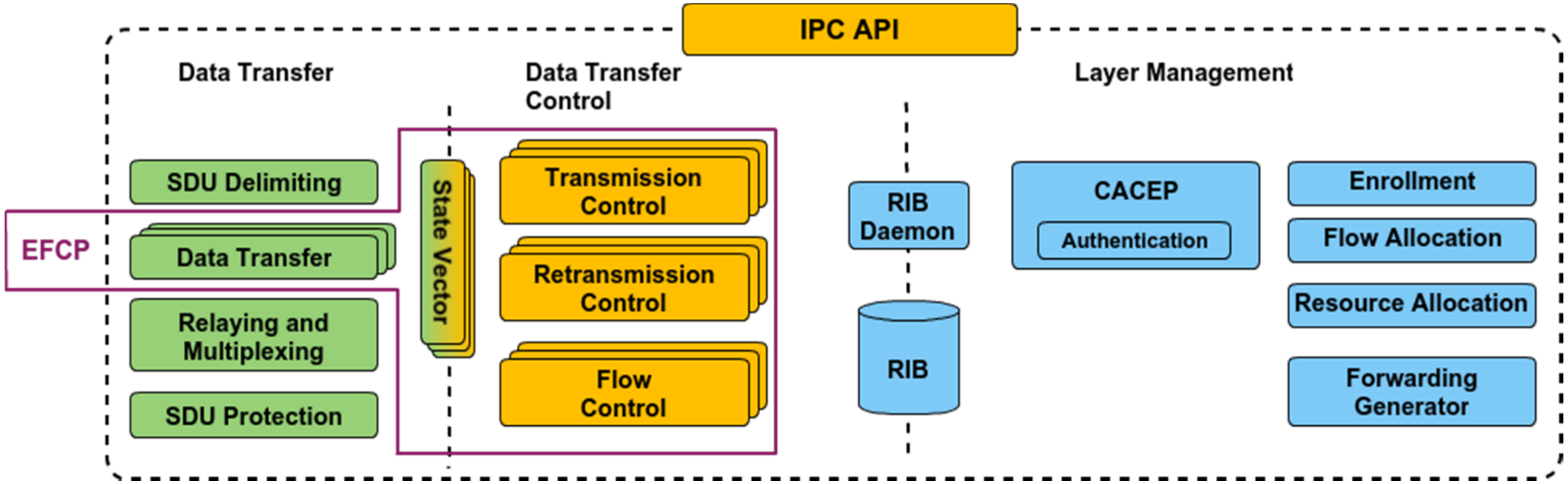
- **APs can interact with an underlying IPCP through the RINA API**: 6 main operations, namely, register/unregister application in the DIF, allocate/release flow, read/write SDU (a packet) from/to a flow
  - ➢ **Same API** also to allow interaction **between upper/lower-level IPCPs**!



Certain QoS can be requested during flow allocation in a technology agnostic way (max. losses, delay, jitter, etc.)

RINA basics (3/6)

- An IPCP carries out three different kinds of functions:
  - **Data transfer functions**: performed for every single PDU (packet), such as fragmentation/assembly, forwarding, scheduling, etc.
  - **Data transfer control functions**: performed on a per-flow basis, to ensure its proper operation, like congestion control, retransmission, etc.
  - **Layer management functions**: to manage the proper layer operation, i.e., functions like authentication, routing, resource allocation or security

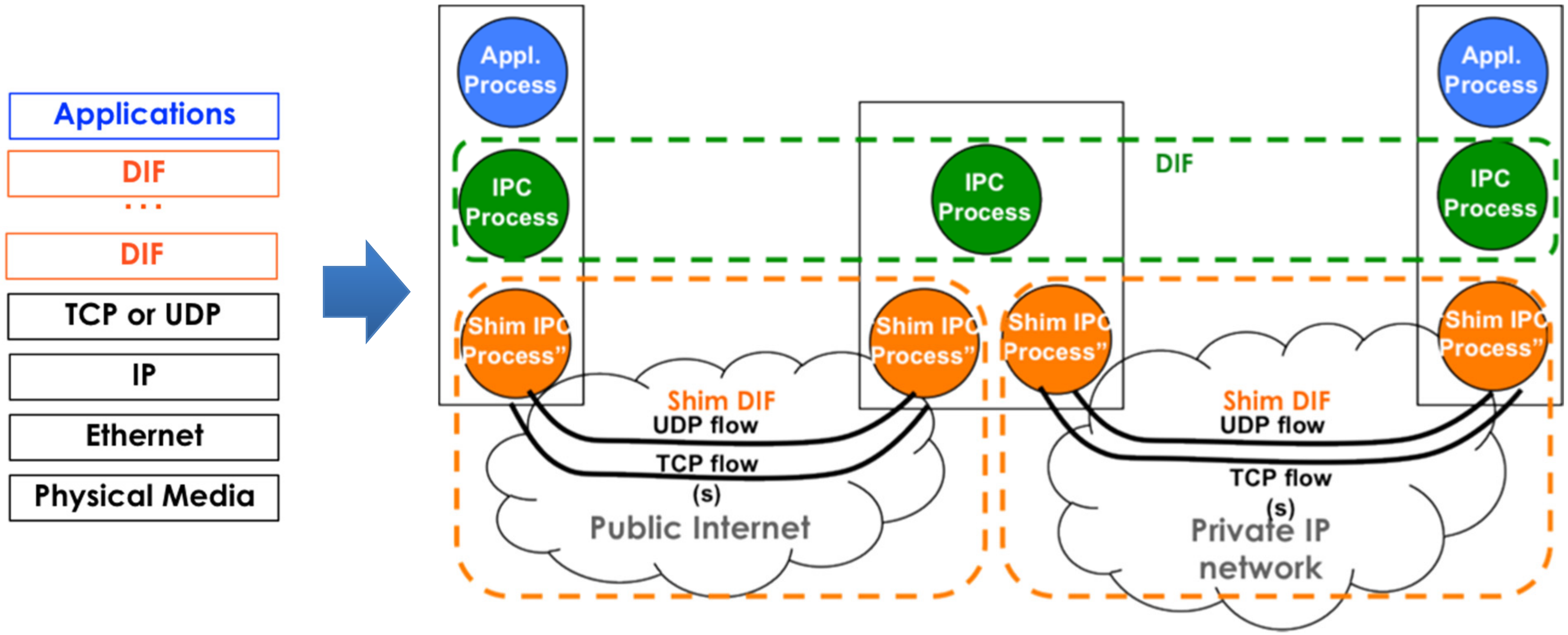


RINA basics (4/6)

- To carry out all these functions, **RINA introduces only 2 protocols**, whose behavior can be customized for any specific DIF environment via policies:
  - **Error and Flow Control Protocol (EFCP)** <- Data transfer control
  - **Common Distributed Application Protocol (CDAP)** <- Layer management
- RINA follows the well-known concept of **separating mechanisms** (common behaviors) **from policies** (variable behaviors) from OSs:
  - DIFs provide a common minimal set of mechanisms that with the appropriate policies allows any transport solution to be realized
  - DIFs can be seen as fully configurable containers, able to tightly adapt their behavior to heterogeneous network scenarios
  - This contrasts with the current TCP/IP model, where the same protocol (e.g., TCP) has to work Internet-wide, even though the characteristics of the underlying networks may substantially differ

RINA basics (5/6)

- At the bottom part of the stack, a DIF has to be interfaced with a legacy technology that does not understand the RINA API ☹
- **Shim DIF**: composed of shim IPC processes (i.e., minimally functional IPC processes) putting a **veneer over a legacy protocol to allow a RINA DIF to use it unchanged**



RINA basics (6/6)

- As mentioned before, **an AP (or an IPCP) can inform about certain desired QoS metrics in a technology agnostic way when requesting the allocation of a flow** to an underlying IPCP (e.g., maximum allowed losses, delay, etc.)
  - Within RINA, each DIF can support a set of **QoS cubes**, i.e., **QoS classes with restrictions on several QoS parameters**
  - Upon flow allocation, the QoS cube best matching the desired QoS metrics will be selected by the DIF
- Of course, **QoS cubes must be enforced by the programmable policies instantiated in the DIF** (e.g., routing, resource allocation, congestion control, retransmission, etc.)
- For example, a shim-DIF over TCP/IP may support only 2 QoS cubes:
  - Unreliable transmission (i.e., UDP employed)
  - Reliable transmission (i.e., TCP employed)

Naming & addressing in RINA (1/4)

- Based on J. Shoch and J. Saltzer works (e.g., see RFC 1498), RINA accounts for a **<u>complete naming and addressing scheme</u>** that considers:
  - Application names
  - Node addresses
  - Point of Attachment (PoA) addresses, i.e., interface addresses
- Moreover, it suggests that:
  - Application names should be *location-independent* to let them be at different places and move around freely
  - Node addresses should *be location-dependent* to give hints in the forwarding decisions (e.g., hierarchical routing), but *route-independent* to untie them to the underlying network graph that may change too often

Naming & addressing in RINA (2/4)

- Note that the **TCP/IP addressing scheme** misses both application names and node addresses, thus being **incomplete**, i.e., a big source of problems!

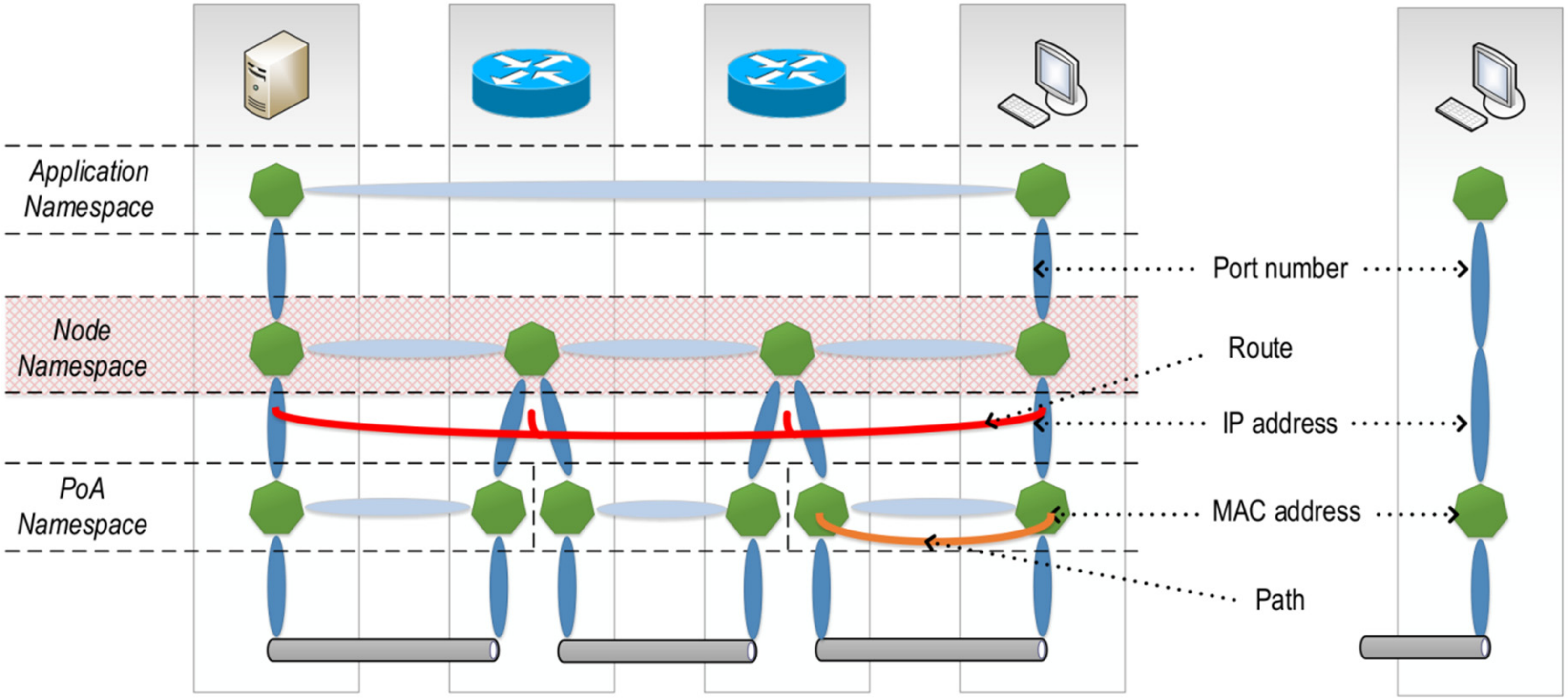


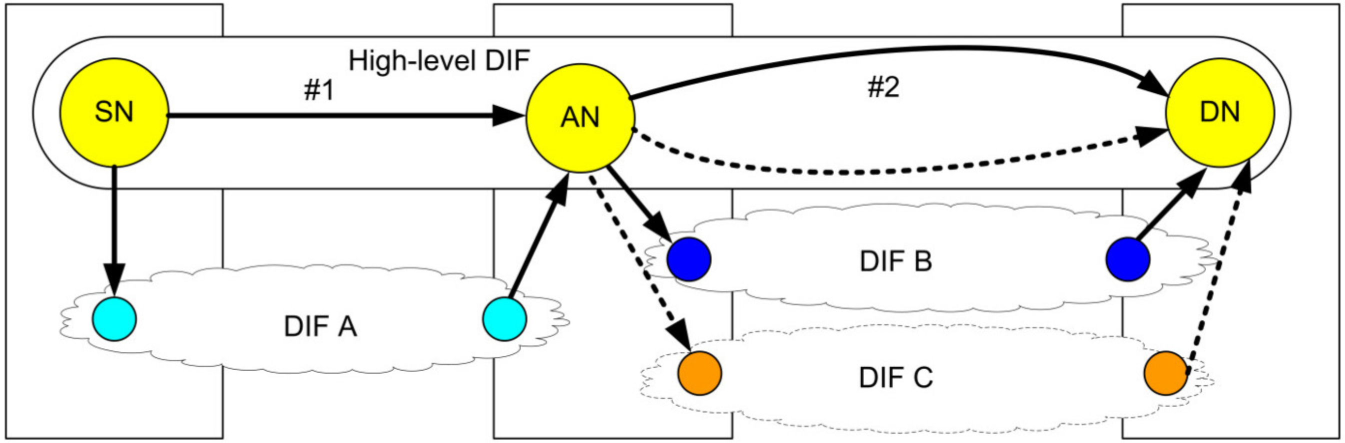
***Figure extracted from V. Veselý, "A new dawn of naming, addressing and routing on the Internet"**

Naming & addressing in RINA (3/4)

- With this naming & addressing scheme, routing is a two-step process:
  1. The **route as a sequence of node addresses** is calculated
  2. For **each hop**, **an appropriate PoA** to select the specific path to be traversed **is chosen**
- These two steps are followed to populate the forwarding table. From that moment, however, forwarding can be performed with a single lookup
- This late binding of next hop node address to underlying PoAs makes multi-homing inherent in RINA:
  - **If a path (PoA) to a next hop node address fails, map it to another operational path (PoA)**
  - This operation can be **performed locally to the routing hop**, i.e., quick and easy to manage

Naming & addressing in RINA (4/4)

- Multi-homing example scenario:



- ➢ SN establishes a flow to DN (traversing AN) over a high-level DIF
- ➢ For resiliency reasons, AN is multi-homed to DIF B and DIF C, over which DN is reachable. Initially, DN address is mapped to the blue IPCP in DIF B
- ➢ Upon failure detection DN address can simply be remapped locally in AN to the orange IPCP in DIF C

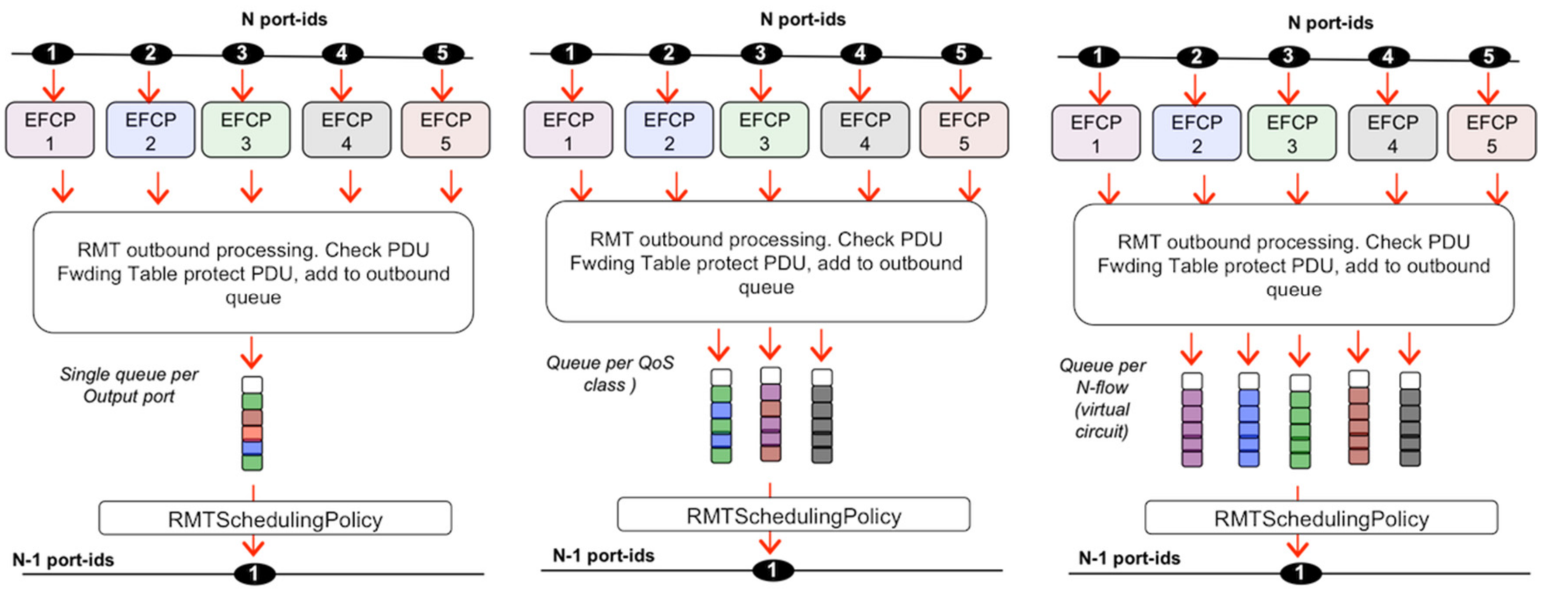
***Example from V. Ishakian, et al., "On the Cost of Supporting Mobility and Multihoming", BU Tech. Report, 2009**

QoS support in RINA DIFs (1/2)

- To address the QoS requirements of the application flows, **a DIF can support different QoS Cubes**, **defining each one a region in the QoS space** (certain ranges of allowed losses, delay, jitter etc.)
- When a **new flow** is allocated, EFCP instances are created at the flow termination endpoints, and **associated to an available QoS cube-id**
  - ➢ PDUs of an EFCP connection should be served in the RMT modules of all IPC processes along its route in a way that its QoS requirements are met
- The Resource Allocator (RA) is the module responsible in the IPC process for the allocation of the sets resources to the queues in the RMT
- Moreover, within the RMT, the Scheduling Policy is the responsible for:
  1. Assigning incoming PDUs from either local EFCP instances or (N-1) ports to the QoS queues according to the PDU Forwarding Table and their qos-id
  2. When an (N-1) output port is available, service one PDU from any of the RMT queues associated to that port, so that QoS requirements are met (for all flows!)

QoS support in RINA DIFs (2/2)

- The joint operation of the RA and the Scheduling Policy in the RMT has a profound impact on the QoS levels that a DIF can provide:
  - RA: Number of queues in the RMT and resources assigned to each of them
  - Scheduling Policy: Strategy employed to service these queues, meeting their QoS requirements as better as possible
- To this end, a wide range of possibilities exist:



Single queue **On queue per QoS class** **One queue per flow**

QTA-Mux (1/2)

- The Quantitative Timeliness Agreement Multiplexer (QTA-Mux) is **the top scheduling policy candidate** for providing **application-specific QoS support in RINA** networks
- It builds upon the ΔQ framework(*) proposed by the company Predictable Network Solutions (www.pnsol.com) in UK
- QTA-Mux includes:
  - A Cherish/Urgency (C/U) multiplexer for QoS Cube differentiation based on delay and loss requirements
  - Policer/Shaper (P/S) modules for intra-flow contention, allowing operations such as flow data rate control, inter-packet spacing, quality degradation, etc.

(*) N. Davies, "Delivering predictable quality in saturated networks", Predictable Network Solutions (PNSol) Technical Report, 2003.

QTA-Mux (2/2)

- Thanks to the C/U Mux, flows receive different treatment at an output port, given the delay & loss requirements defined by the assigned QoS Cube
  - ➢ It enforces a bi-dimensional QoS differentiation based on Urgency (i.e., delay) and Cherish (loss requirements), as specified by the C/U matrix

2

PDUs of flows serviced based on their urgency level (One queue per Urgency level, strict priority)

+ Cherished -

Urgency		
+	**A1**	**A2**
-	**B1**	**B2**

2x2 C/U matrix

1

PDUs of flows admitted based on their cherish level and the current occupancy of their urgency queue

- ➢ **A1**: Prioritized w.r.t. losses & prioritized w.r.t. delay
- ➢ **A2**: Un-prioritized w.r.t. losses, but prioritized w.r.t. delay
- ➢ **B1**: Prioritized w.r.t. losses, but un-prioritized w.r.t. delay
- ➢ **B2**: Un-prioritized w.r.t. losses & un-prioritized w.r.t. delay

The ERASER experiment

- ERASER (*) aims at evaluating the QoS guarantees that RINA can deliver to heterogeneous applications in large-scale network scenarios
- Medium experiment accepted in the 3rd Fed4FIRE+ Open Call
  - Duration: March-September, 2018
- Fed4FIRE+ is an H2020 European IP project building a federation of experimentation facilities (wired, wireless/5G/IoT, OpenFlow, etc.) across Europe for large-scale FIRE, which organizes a series of competitive open calls for innovative experiments over their infrastructures
  - Experiments to test/implement/optimize existing products or services on Fed4FIRE+, not to propose/develop new ideas from scratch!

(*) https://zenodo.org/record/1420391

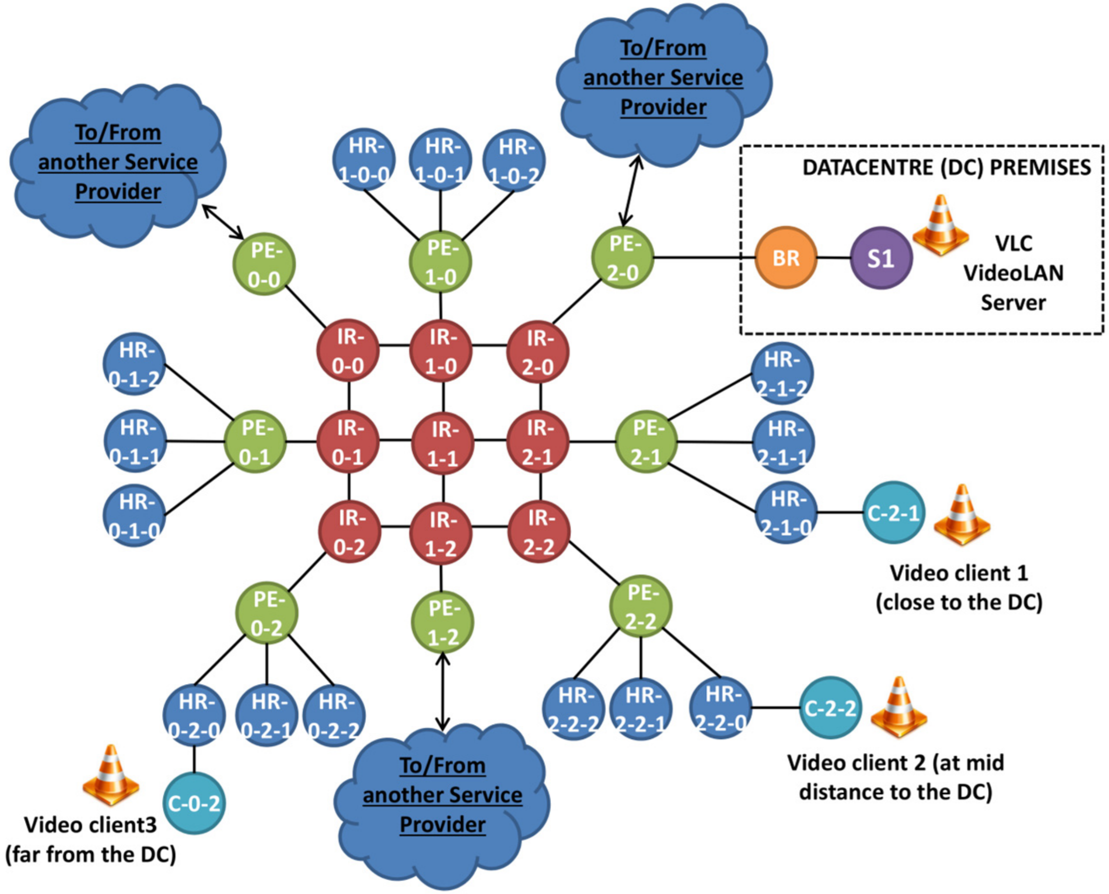
https://www.fed4fire.eu/

Objectives

1. To define **QTA-Mux deployment scenarios** and **QoS Cubes** to be enforced in a **large-scale RINA network** scenario, with multiple DIFs stacked one on top of another
2. To emulate a **realistic large-scale RINA network** scenario by injecting **synthetic traffic flows** reproducing **heterogeneous applications**
3. To **evaluate the RINA QoS support** by measuring the **QoS metrics perceived** by synthetic traffic flows end-to-end
4. To perform a **real HD video (1080p) streaming demo** to better appreciate the RINA QoS support under high congestion

Evaluation scenario (1/4)

- A **37-Node metro-regional network** has been setup **on the Fed4FIRE+ Virtual Wall test-bed** hosted by imec in Ghent (Belgium):



A slice of 37 physical machines with Linux OS has been requested, acting as:

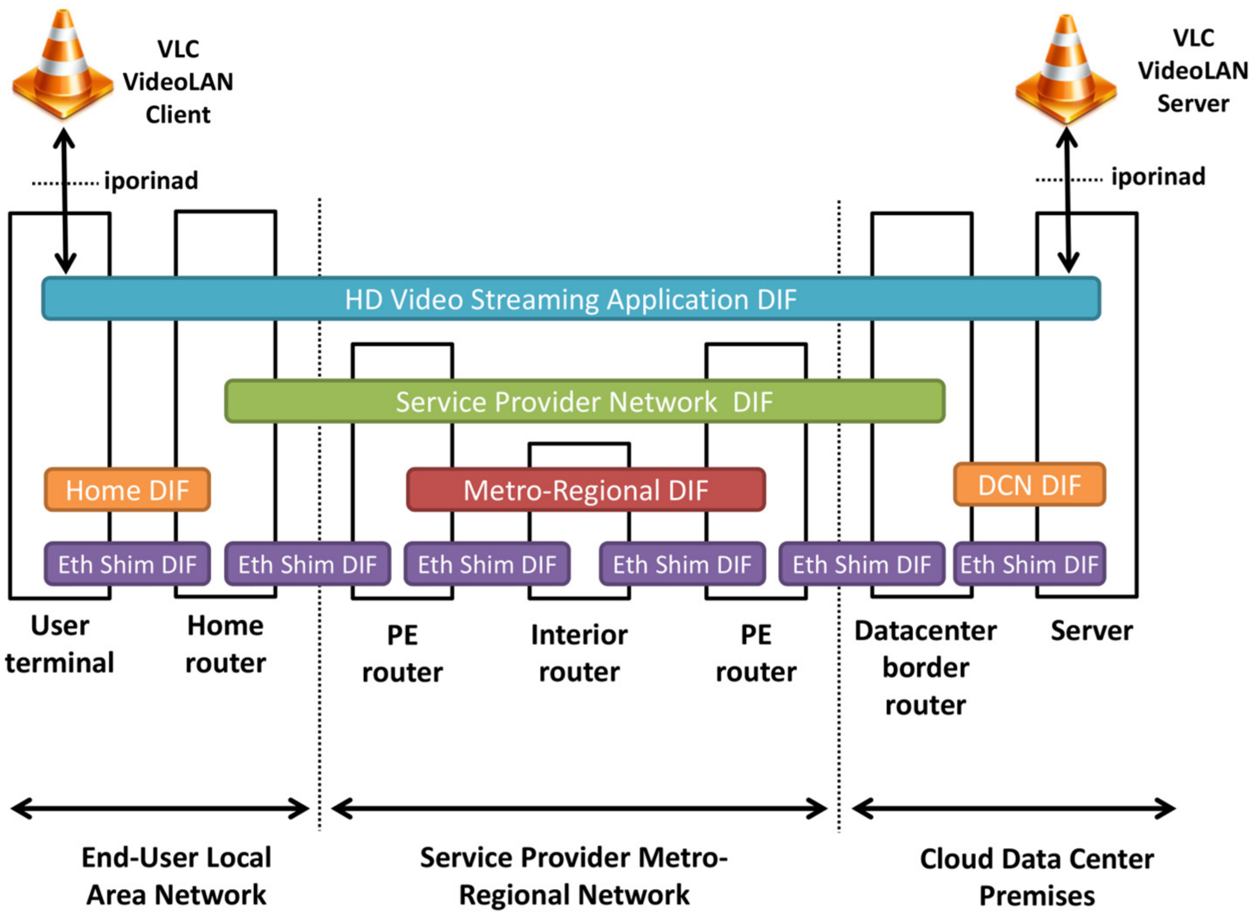
9 Interior Routers (IRs)
8 Provider Edge (PE) routers
15 Home Routers (HRs)
3 end-user terminals (Us)
1 DC Border Router (BR)
1 Server (S)

...interconnected by GbE links

The open source IRATI RINA stack has been installed in each one of them

Evaluation scenario (2/4)

- Moreover, the following DIFs have been configured there:



Evaluation scenario (3/4)

- Evaluated QTA-Mux deployment scenarios:
  - ➢ At both Service Provider Network (SPN) and Metro-Regional (MR) DIFs
  - ➢ Only at the SPN DIF
- QoS Cube differentiation enforced by QTA-Mux at SPN and MR DIFs:

	+	Cherished	-
+	Gold		-
Urgency	Silver		Sensitive BE
-	Bronze		BE

QoS Cubes offered by the SPN DIF

+	Cherished	-	
A1		A2	+
			Urgency
B1		B2	-

QoS Cubes offered by MR DIF

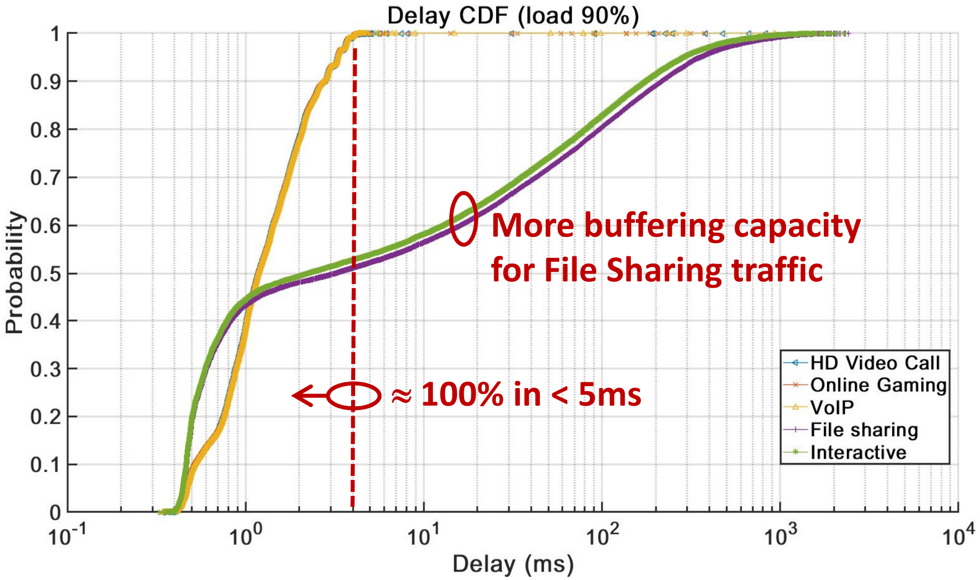
Evaluation scenario (4/4)

- Employing user-developed client-server applications, five types of synthetic flows have been injected into the network:

Application flow	Traffic distribution	Traffic direction	Details	Assigned to QoS Cube
HD Video Call	CBR	Bidirectional	CBR bitrate: 1.5 Mbps	Gold
Online Gaming	ON-OFF	Bidirectional	ON-OFF period avg. duration: 4s – 2s CBR bitrate during ON period: 4 Mbps	Silver
VoIP	ON-OFF	Bidirectional	ON-OFF period avg. duration: 3s – 3s CBR bitrate during ON period: 64 kbps	Sensitive BE
File Sharing	ON-OFF	Bidirectional	ON-OFF period avg. duration: 2s – 1s CBR bitrate during ON period: 5 Mbps	Bronze
Interactive traffic	Poisson	Bidirectional	Avg. bitrate: 2 Mbps	BE

Obtained results (1/3): QTA-Mux at both SPN and MR DIFs

- 75 bidirectional synthetic traffic flows injected into the SPN DIF between PE-0-0, PE-2-0 and PE-1-2 and specific HRs in the network
- NIC capacities limited to reproduce 70, 80 and 90% offered load scenarios
- **Adequate QoS differentiation even under high congestion**:

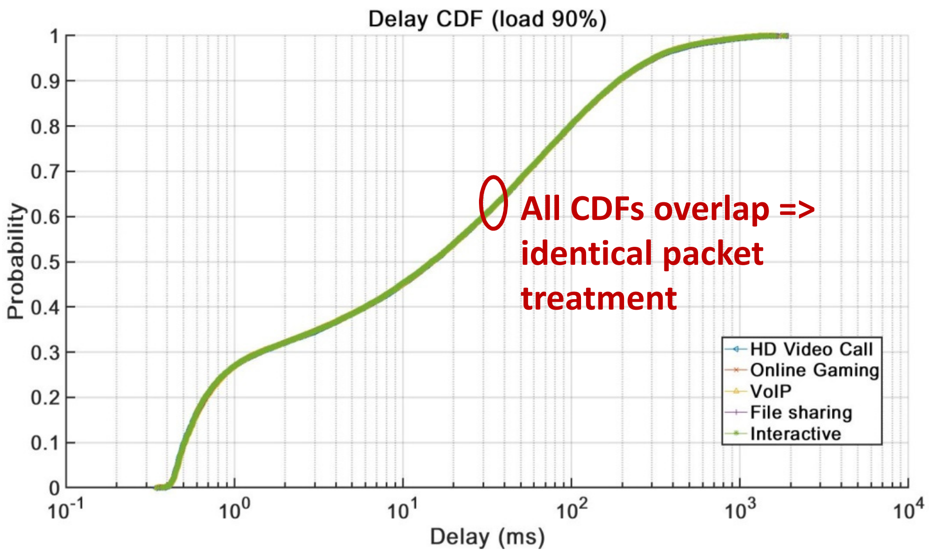


Perceived QoS metrics (90% load)

Traffic type	Min. latency (ms)	Max. latency (ms)	Avg. latency (ms)	Avg. packet loss (%)
HD Video Call	0.345145	830.5	1.483987	0.001659
Online Gaming	0.3626	375.76	1.468362	0.002339
VoIP	0.345375	924.95	1.469585	0.001342
File Sharing	0.354075	2408.85	72.310218	0.231222
Interactive	0.32893	2276	58.650972	3.469220

Obtained results (2/3): QTA-Mux at the SPN DIF only

- 75 bidirectional synthetic traffic flows injected into the SPN DIF between PE-0-0, PE-2-0 and PE-1-2 and specific HRs in the network
- NIC capacities limited to reproduce 70, 80 and 90% offered load scenarios
- **QoS differentiation lost at the MR DIF** (default FIFO scheduling):

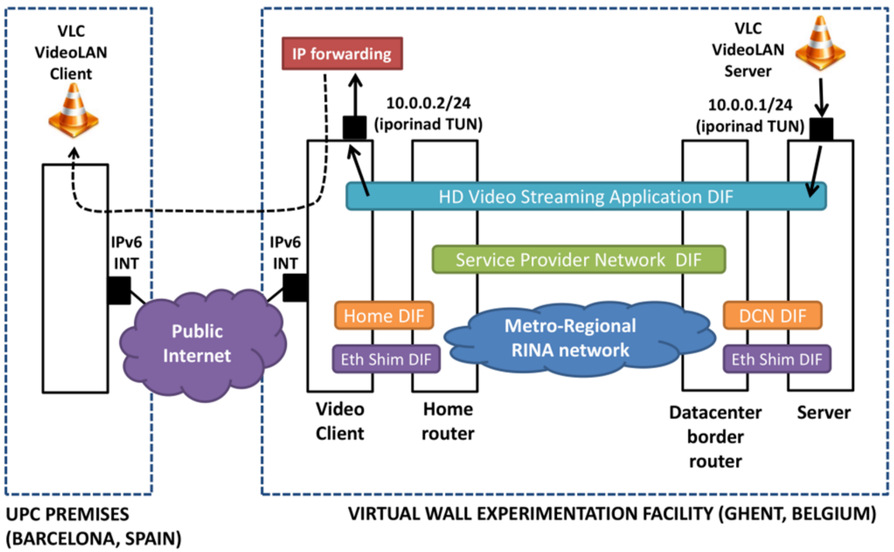


Perceived QoS metrics (90% load)

Traffic type	Min. latency (ms)	Max. latency (ms)	Avg. latency (ms)	Avg. packet loss (%)
HD Video Call	0.342600	1889.6	69.764187	0.139793
Online Gaming	0.343865	1893.9	70.126824	0.173666
VoIP	0.345180	1846.05	69.738218	0.150003
File Sharing	0.351630	1876.3	70.223907	0.155559
Interactive	0.346925	1865.15	69.441732	0.185567

Obtained results (3/3): HD video streaming demo over RINA

- 75 bidirectional synthetic traffic flows injected into the SPN DIF between PE-0-0, PE-2-0 and PE-1-2 and specific HRs in the network
- NIC capacities limited to reproduce 70, 80 and 90% offered load scenarios
- QTA-Mux configured at HD Video Streaming, SPN & MR DIFs



Perceived QoE (80% load) from 1 (worst) to 5 (best)

QoS Cube assigned at HD Video Streaming DIF	S1 -> C-2-1 (1 hop at MR DIF)	S1 -> C-2-2 (2 hops at MR DIF)	S1 -> C-0-2 (4 hops at MR DIF)
Gold	5 (perfect quality)	5 (perfect quality)	5 (perfect quality)
Best Effort	2 (severe stuttering & frames lost)	1 (completely frozen)	2 (severe stuttering & frames lost)

Perceived QoE (90% load) from 1 (worst) to 5 (best)

QoS Cube assigned at HD Video Streaming DIF	S1 -> C-2-1 (1 hop at MR DIF)	S1 -> C-2-2 (2 hops at MR DIF)	S1 -> C-0-2 (4 hops at MR DIF)
Gold	5 (perfect quality)	3 (moderate stuttering & frames lost)	5 (perfect quality)
Best Effort	2 (severe stuttering & frames lost)	1 (completely frozen)	1 (completely frozen)

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

The end

Future Internet Networks (FINE)

Master in Innovation and Research in Informatics (MIRI)