Congestion Notification for Multi-Resource Allocation Control

Mario Patetta

Center for Studies and Research in Computer Science and Communication

Cnam, Paris

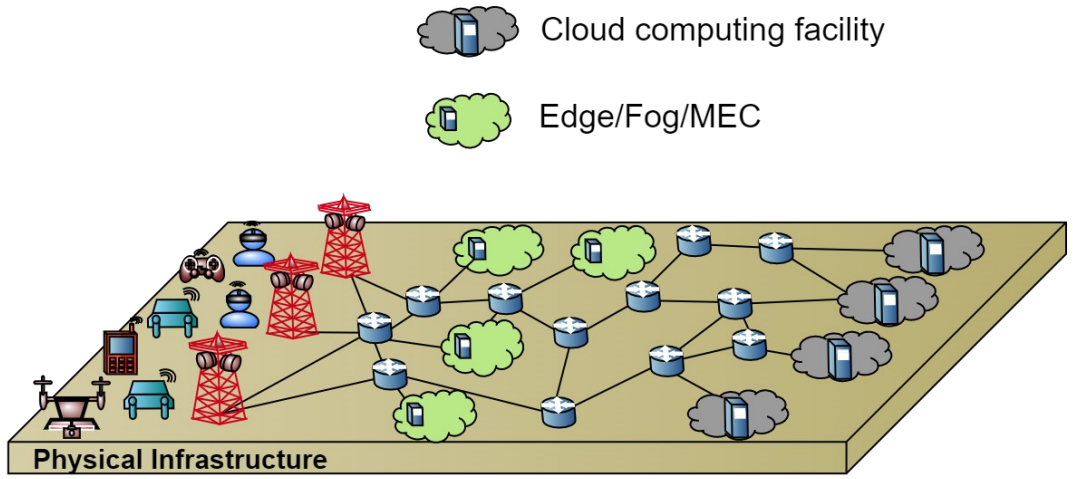
mario.patetta@cnam.fr

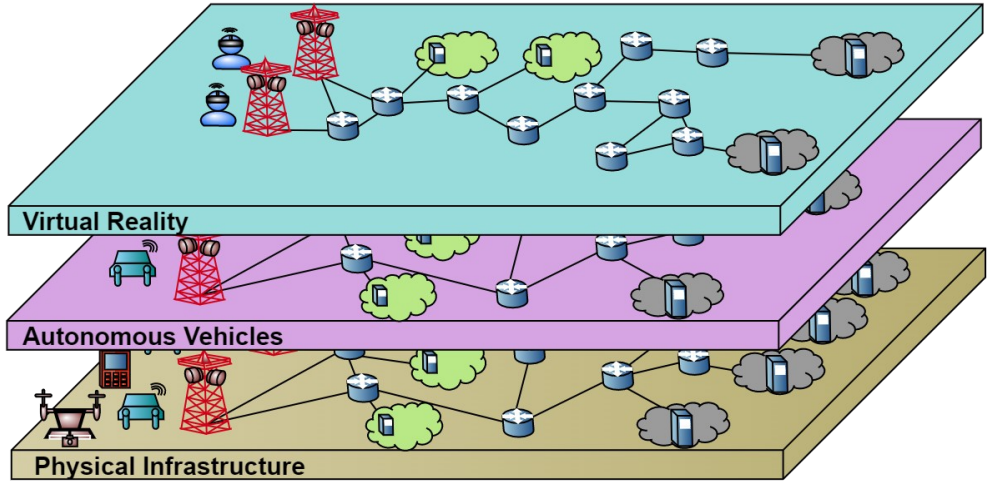
Outline

- Context & Motivation
- Multi-Resource Explicit Congestion Notification
  - Definition of Multi-Resource Congestion and Waste
  - Comparison between ECN and MR-ECN
  - Possible add-on: Tailored-MR Adjustment
- Final notes

Part 1: Context & Motivation

Towards coordinated end to end resource allocation





Credit: Mandar Datar et al

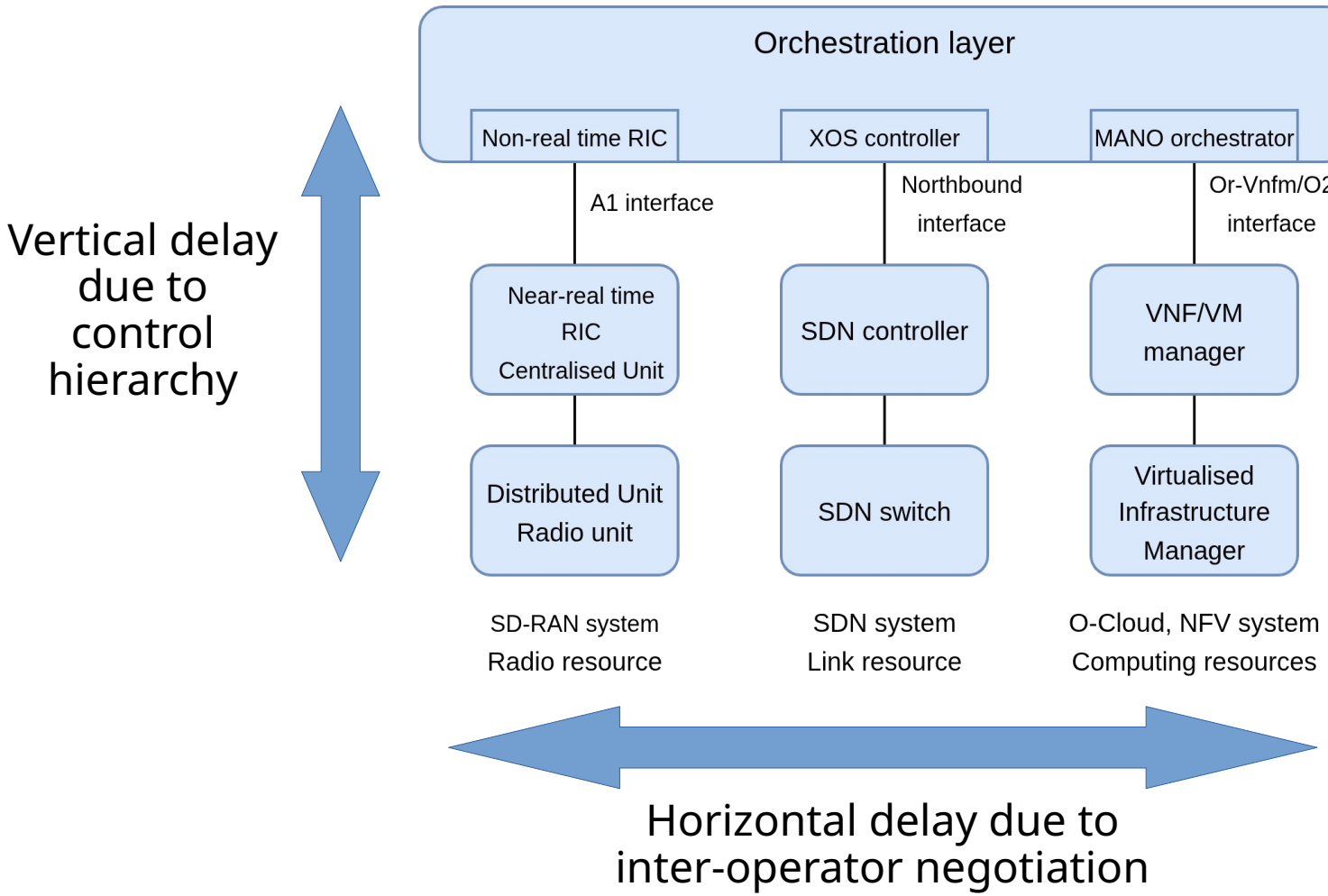
Services supported by modern networks imply:

- Different resources (radio spectrum, bandwidth, computing power, ...)
- Tailored allocation → network slicing
- Shared physical infrastructure, managed by several operators

le cnam cedric EA4629

Multi-Resource, Multi-Operator Allocation…

… Leading to a Dynamic, Hierarchical, Multi-Resource Scheduling Framework



- ✔ It works for dynamically allocating resources to users.
- ✖ But what if a sudden resource loss occurs (e.g., device failure/reboot)?
- 💡 Can we bypass controllers and apply coarse but real-time allocation adjustments?

Part 2: Multi-Resource Explicit Congestion Notification

Multi-Resource Congestion
Cross-Resource Waste

Congestion of resource *j* allocated to 'slice' *i*:

$$congestion_{ij} = \frac{loss_{ij}}{allocated_{ij}}$$

Congested resources constitute a **bottleneck**

→ should adjust other allocations

$$waste_{ij} = \sum_{t=0}^{T} \left| \frac{adjustment_{ij}(t)}{allocated_{ij}} - \frac{loss_i^c(t)}{allocated_i^c} \right| \Delta t \quad *$$

- If adjustment = 0 → waste is the cumsum of congestion over time
- If adjustment = loss → no waste

* assuming proportional dependency, aka Leontief utility

Example of Cross-Resource Waste

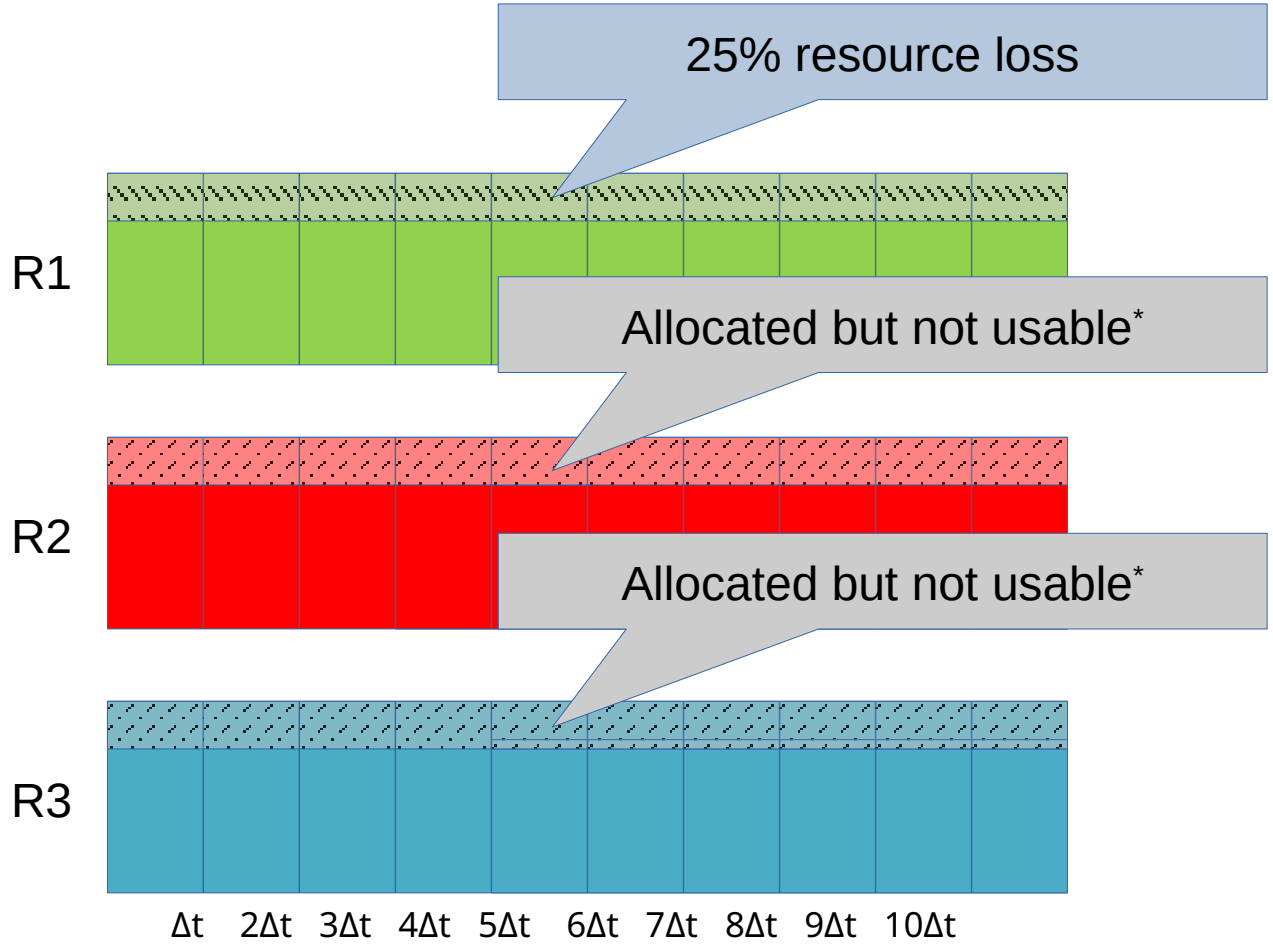
- Slice *i* is allocated resources:
  R1=20 R2=8 R3=480

- R1 availability suddenly drops to 15 (congestion=0.25) for T = 10 Δt

→ 25% of R2 and R3 is wasted*

- After 3 Δt, R2 is adjusted to 8*0.75 = 6
- After 4 Δt, R3 is adjusted to 480*0.8 = 384 (under adjusted)

→ wasted R2 = 0.25*3 Δt = 0.75 Δt
→ wasted R3 = 0.25*4 Δt + 0.05*6 Δt = 1.3 Δt

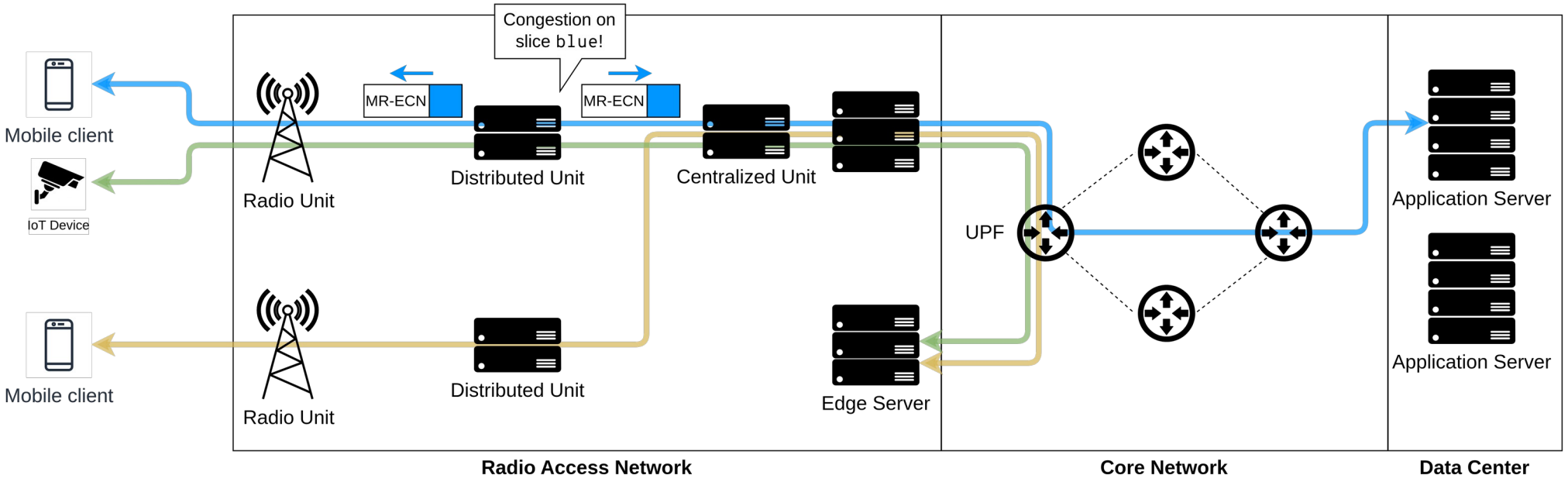


* assuming proportional dependency, aka Leontief utility

Multi-Resource Explicit Congestion Notification

distributed RAN use-case

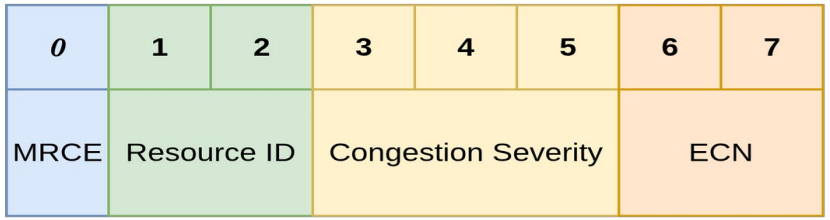
- Notification injected into affected packets (both directions)
- Recipients → equipment traversed by the affected ‘slice’
- Effect → reduce allocation to affected ‘slice’



ECN → MR-ECN

from flow congestion to resource congestion notification

	ECN	MR-ECN
Purpose	Bypass pkt loss in TCP's congestion avoidance	Bypass controller hierarchies in multi-operator allocation adjustment
Recipients	End devices	Data plane nodes
Resource	Link capacity	Any
Granularity	Flow or MPLS trunk (RCF 5129)	Slice or similar (e.g., MPLS trunk)



Possible use of DSCP bits (MR-ECN adjacent to ECN)

- Multi-Resource Congestion Experienced (1 bit)
- Resource ID (2 bits) → radio, computing (RAN), bandwidth, computing (cloud/edge)
- Congestion Severity (3 bits)
- ECN untouched

DSCP: DiffServ Code Point

Possible add-on: Tailored-MR Adjustment

Operators can flexibly define, for each slice:

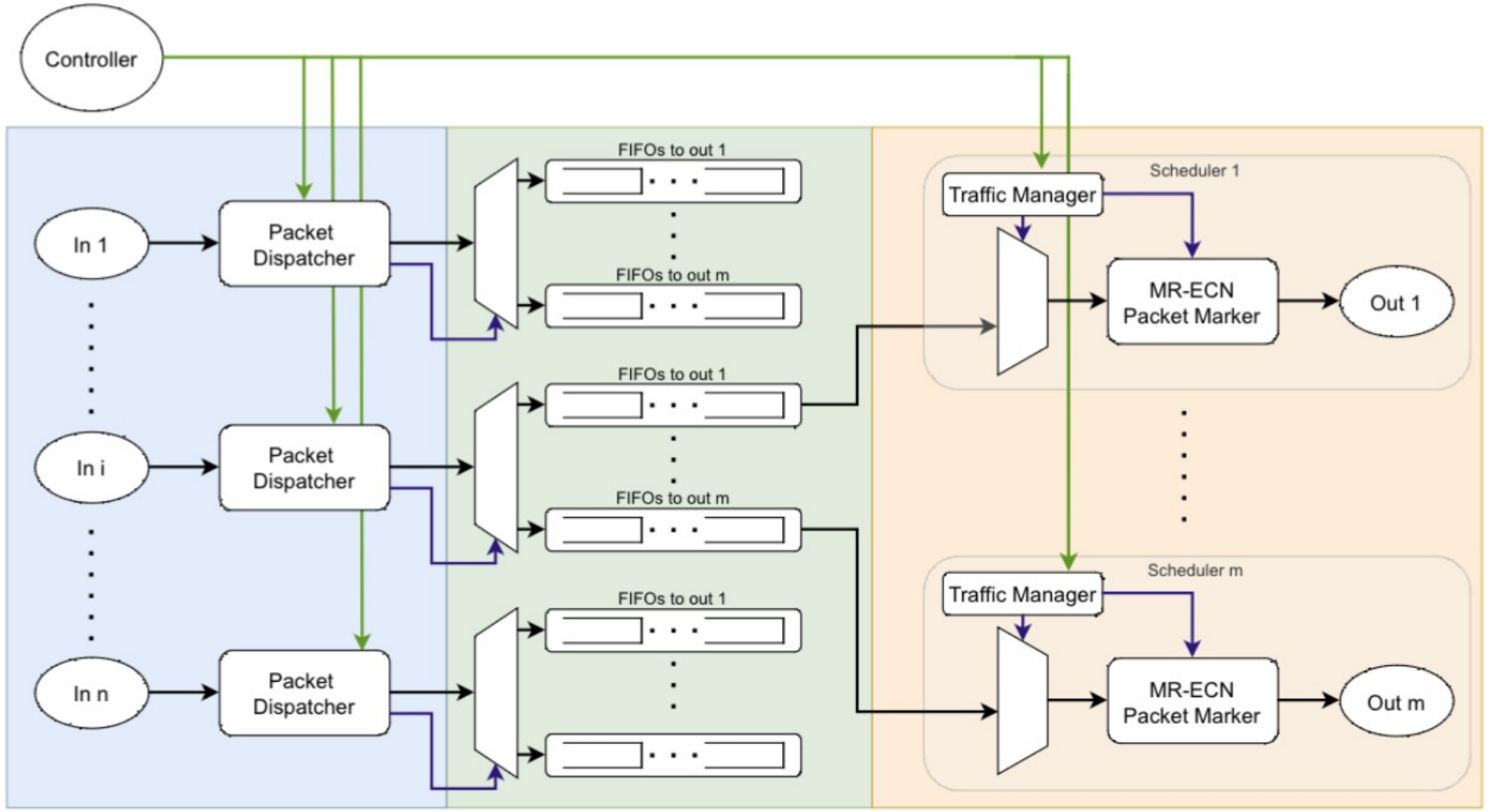
Congestion Severity encoding (how to quantise congestion into 3 bits)

- Match congestion → minimise waste, but unstable resulting bandwidth
- Max congestion over last period → fast and stable allocation, but might over-adjust
- Avg congestion over last period → less over-adjustment, but also less stable
- …

Allocation adjustment based on the received notification

- Proportional → minimise waste (assuming Leontief utility function)
- Exponential → suppress anomalous, low-priority slices
- …

MR-ECN integration in FPGA switch



MR-ECN programmable scheduler block diagram.

Part 3: Final Notes

Notes :

Slicing is just a use case example, MR-ECN can be employed in any situation with multiple controllers/inter-resource dependencies

We use DSCP field just for illustrative purposes

What can be improved :

Only 2 bits for Resource ID, 3 bits for Congestion Severity

What if more than one resource is congested?

Possible alternatives :

Use MPLS

- Exp field → only 3 bits (1 for MRCE + 2 Cong Sev) – à la E-LSP
- Label+exp …. – à la L-LSP

Use dedicated header between L2 and L3 – à la In-band Network Telemetry

Use extension header between IPv6 and L4

Questions or Comments ?

Annex

Sample of MR-ECN behavior