Routing-Aware Shaping for Feasible Multi-Domain Determinism

Andrea Francini

Network Systems and Security Research Lab

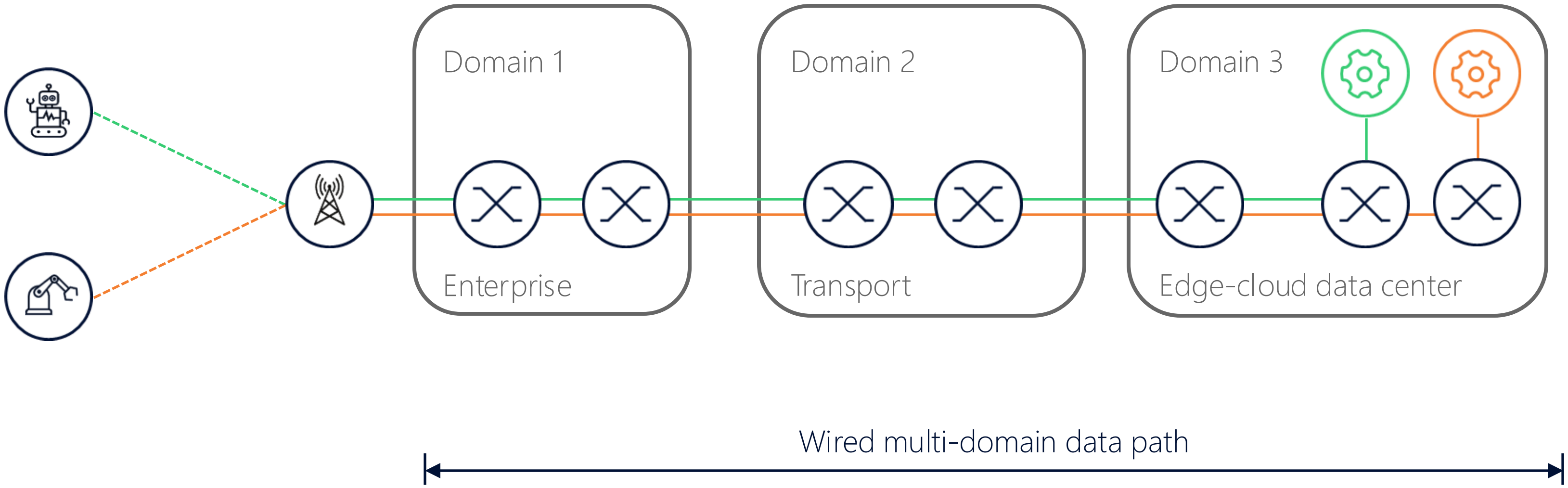
Nokia Bell Labs Core Research

November 18, 2024

6G-PDN2 Workshop, ACM Mobicom 2024

Multi-domain determinism: a typical use case

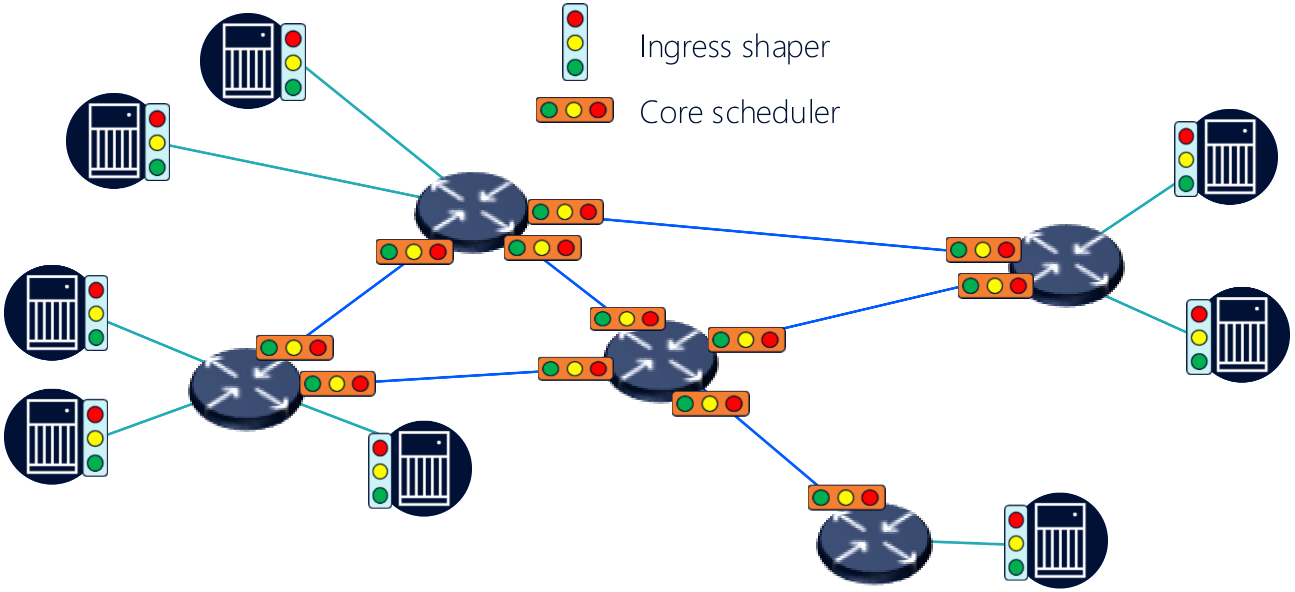
Industrial automation with virtualized controllers



Our design approach: concatenation of per-domain instances of a common deterministic framework

NOKIA BELL LABS

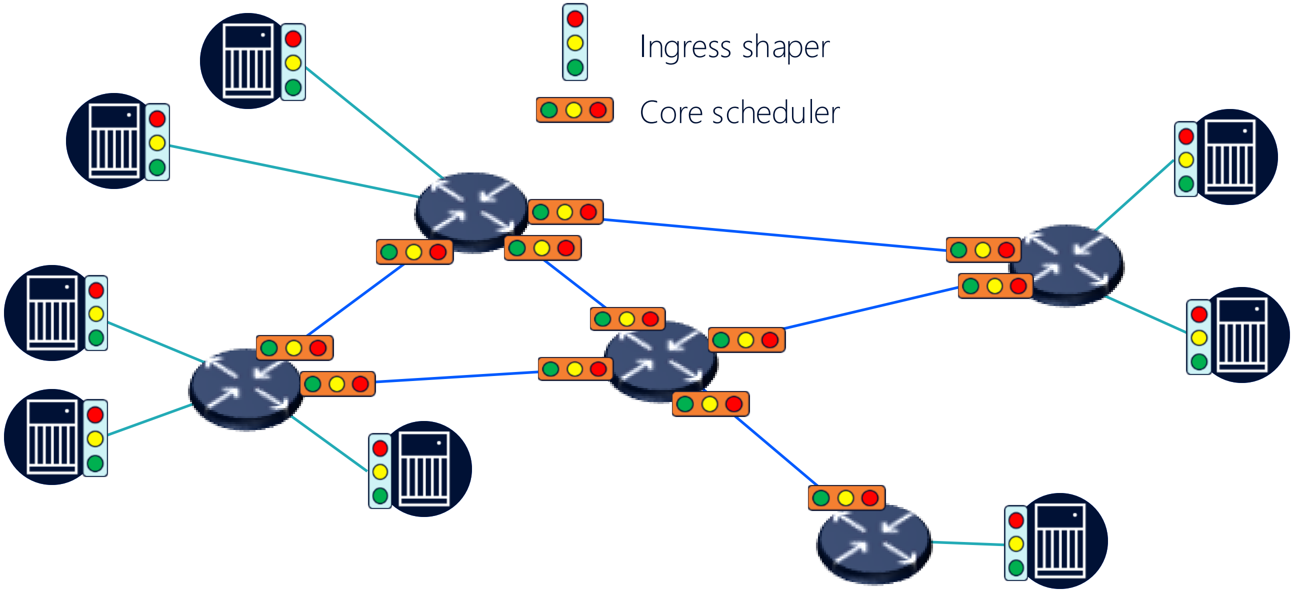
Domain determinism: the standard way (IEEE TSN, IETF DetNet)



Resource provisioning for new deterministic services scales poorly with the size of the network and the dynamicity of the services

Core schedulers require new hardware and per-flow provisioning

Domain determinism: the feasible way

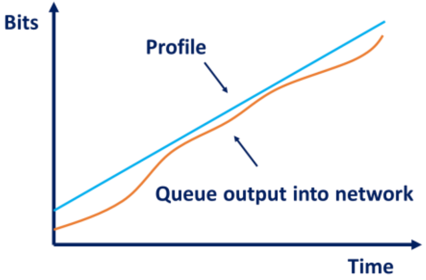


Resource provisioning for new deterministic services scales well with the size of the network and the dynamicity of the services

No core schedulers: no new hardware, flow provisioning at ingress links only

Under the spotlight: the ingress shaper

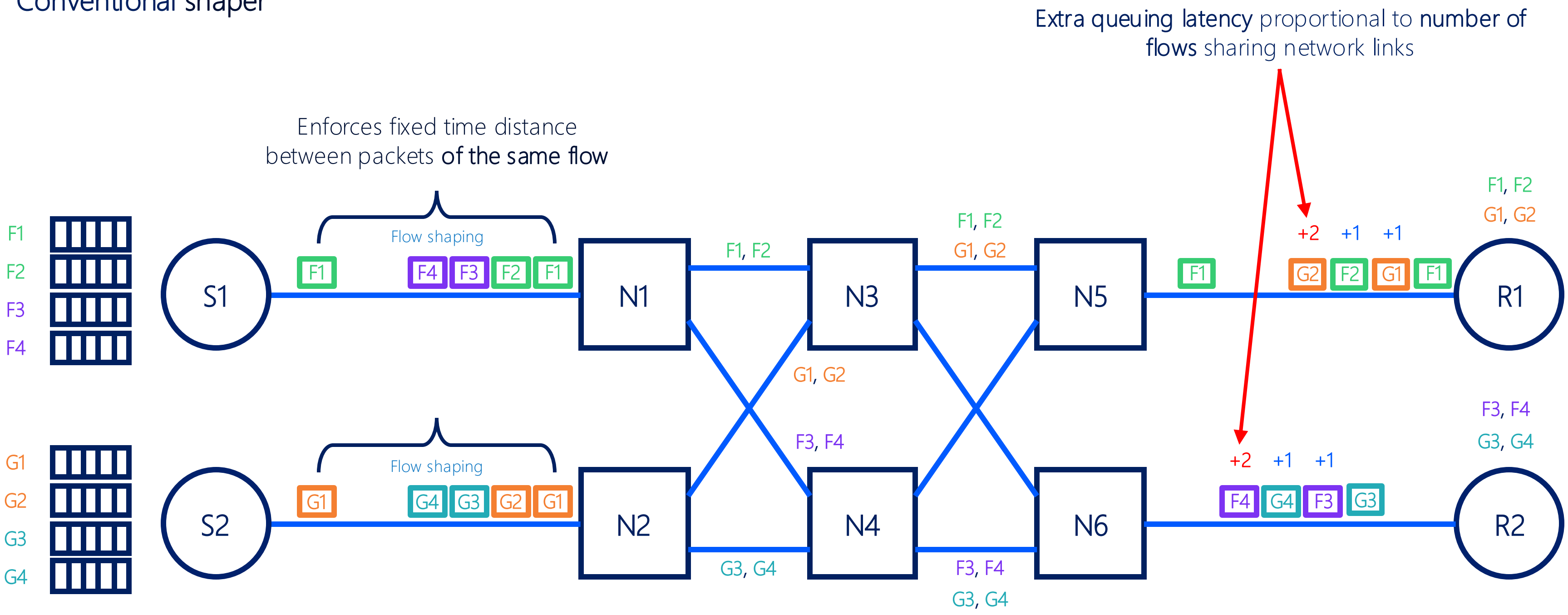
- Commonly used to enforce compliance of traffic flows with respective profiles



- Central to older frameworks for feasible determinism
  - IETF DiffServ [1998]
  - SharpEdge [2020]
  - Chameleon [2020]
- Still missing in those frameworks:
  - Latency isolation from number of flows
- Still needed in those frameworks:
  - Heavy overprovisioning of network capacity

Our game changer: routing awareness

Conventional shaper

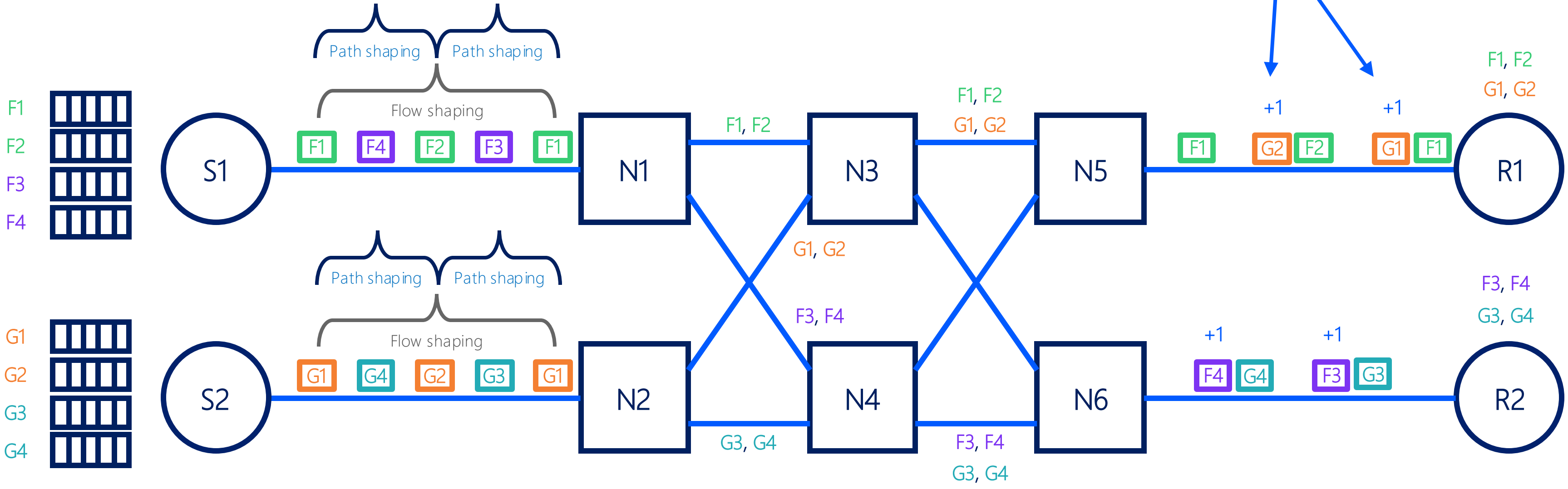


Our game changer: routing awareness

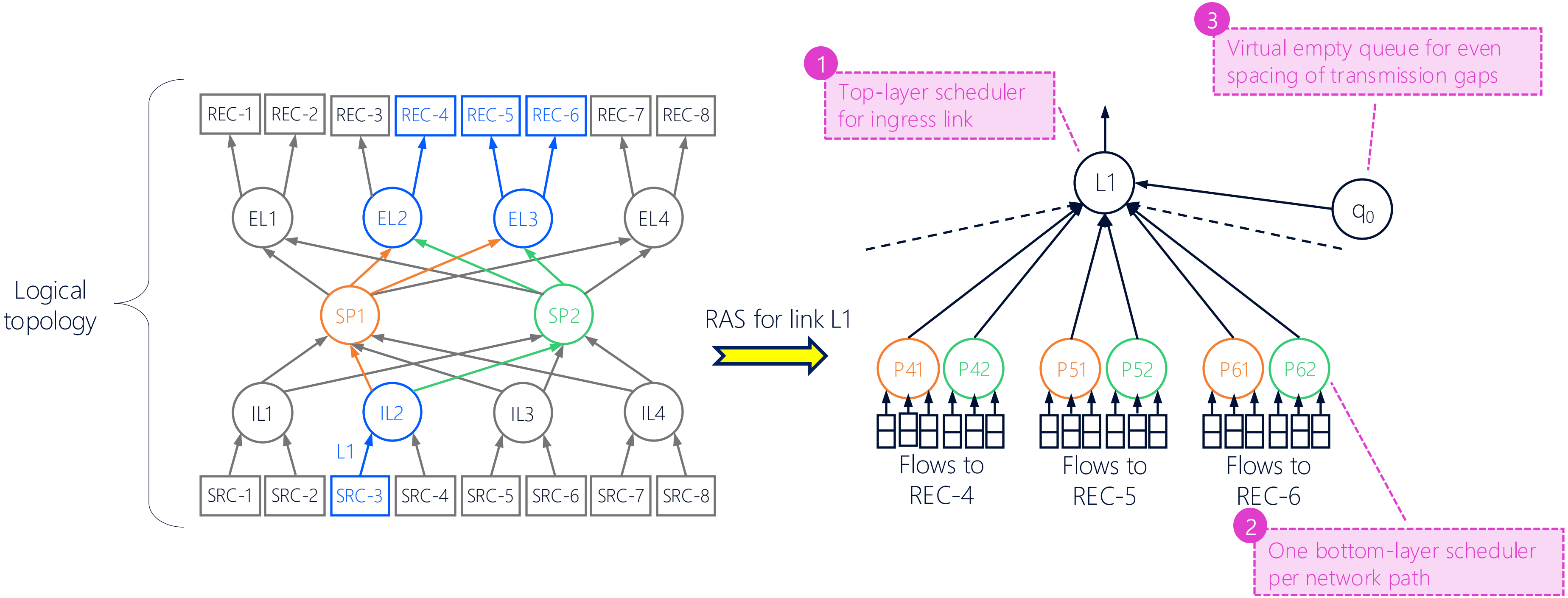
Routing-aware shaper (RAS)

Enforces fixed time distance between packets **of the same flow** and packets **with identical network path**

Extra queuing latency proportional to **number of ingress links** with flows sharing network links



The RAS: how we build it



Two-layer scheduling hierarchy, irrespective of network topology

NOKIA BELL LABS

The RAS: how we run it

Path **selection** in top-layer shaping node

```
// Variables
LinkVect[NUM_LINKS];   // Array of all network links
Node.PathList;         // Path pointers sorted by FinishTime
Node.LinkId[NUM_LINKS];// Identifiers for links of a path
Node.NumLinks;         // Number of links in routing path
Node *sched;           // Top shaping node
Node *selpath = NULL;  // Path selected for current timeslot
Node *auxpath;         // Auxiliary variable for path search
int ii;                // Auxiliary counter for link search

// Code
auxpath = sched->PathList; // Path with lowest FinishTime
while(selpath == NULL) {
    if(auxpath->StartTime <= sched->VirtualTime) {
        // Path has eligible StartTime
        if (auxpath->NumLinks == 0) {
            // No links: path is virtual empty queue, choose it
            selpath = auxpath;
        }
        else {
            // Path has links: scan them for eligibility
            for (ii = 0; ii < auxpath->NumLinks; ii++) {
                if (LinkVect[auxpath->LinkId[ii]].StartTime >
                    sched.VirtualTime) {
                    // Link not eligible: path not eligible
                    break;
                }
            }
            if (ii == auxpath->NumLinks) {
                // All links are eligible: select the path
                selpath = auxpath;
            }
        }
    }
    // Keep searching
    auxpath = auxpath->next;
}
return (selpath);
```

Canonical WF2Q(*)

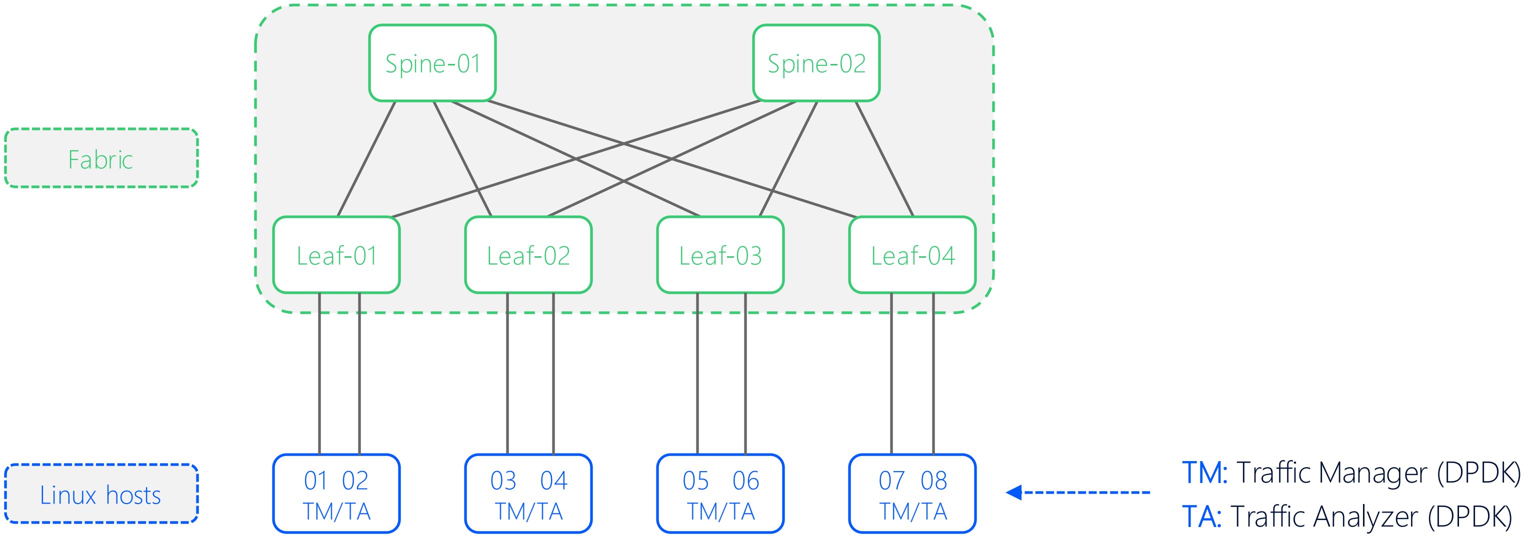
Main novelty here: candidate path is eligible for service only if all its links are eligible

(*) J. C. R. Bennett and Hui Zhang, "WF2Q: worst-case fair weighted fair queueing," Proceedings of IEEE INFOCOM '96.

A hierarchy of Worst-case-Fair Weighted-Fair-Queuing (WF2Q) nodes, augmented with link eligibility on top-layer node

RAS evaluation testbed

Leaf-spine data center fabric with 8 servers, all links are 25 Gb/s



Results from the testbed: domain traversal latency (1)

FLAT: conventional per-flow shaper (WF2Q)

RAS: routing-aware shaper

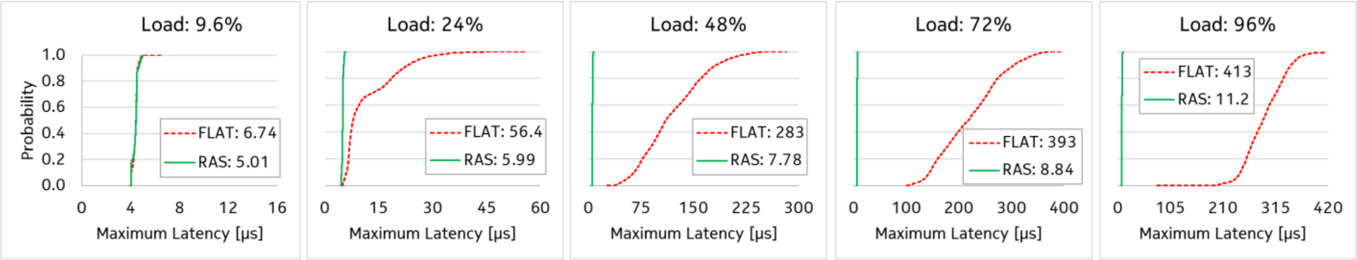
Conjectured upper bound on network traversal latency:

$$D_i = 2T \sum_{l \in \Pi_i} n_l$$

Where:

- T is the duration of the TDM timeslot
- Π_i is the fabric path of flow i
- l is a fabric ink in the path of flow i
- n_l is the amount of ingress links that send flows to link l

In the 25 Gb/s testbed: $\sum_{l \in \Pi_i} n_l = 14$ => $\boldsymbol{D_i = 14.4\ \mu s}$



Test: uniform traffic from *identical* flows, multiple load levels

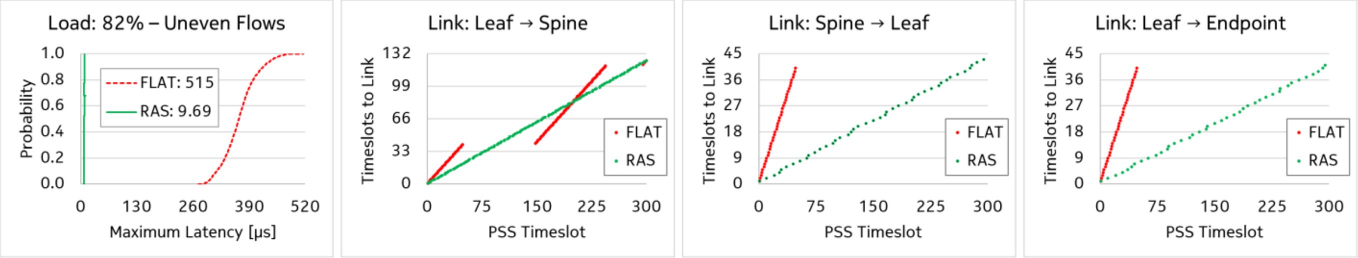
RAS highlight: maximum latency remains low irrespective of load level, and well below the conjectured bound

Results from the testbed: domain traversal latency (2)

“Why” it works

FLAT: conventional per-flow shaper (WF2Q)

RAS: routing-aware shaper



Test: uniform traffic from *diverse* flows, 82% load

RAS highlights:
(a) 20x reduction of maximum latency; (b) smooth distribution of packet transmissions to interior links

Conclusions

Routing-aware shaping as the building block for feasible multi-domain determinism

- Robust performance, easy to provision, implementable in software

Proven in small data-center setup

- Now looking at industrial automation use case

Open for collaboration

Thank you!

NOKIA BELL LABS

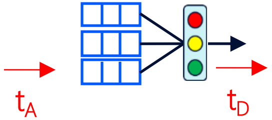
Backup slides

The RAS: how we concatenate it over multiple domains

Theoretical latency bounds

1. Ingress shaping latency (proven)

$$D = t_D - t_A$$

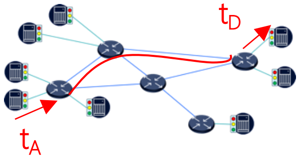


$$D_i^{\sigma} = \frac{b_i}{r_i} + \frac{2TC}{r_i}$$

Source burstiness

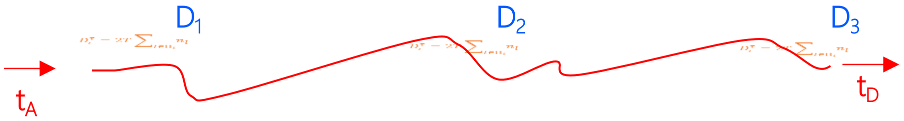
Shaper latency

2. Domain traversal latency (conjectured)



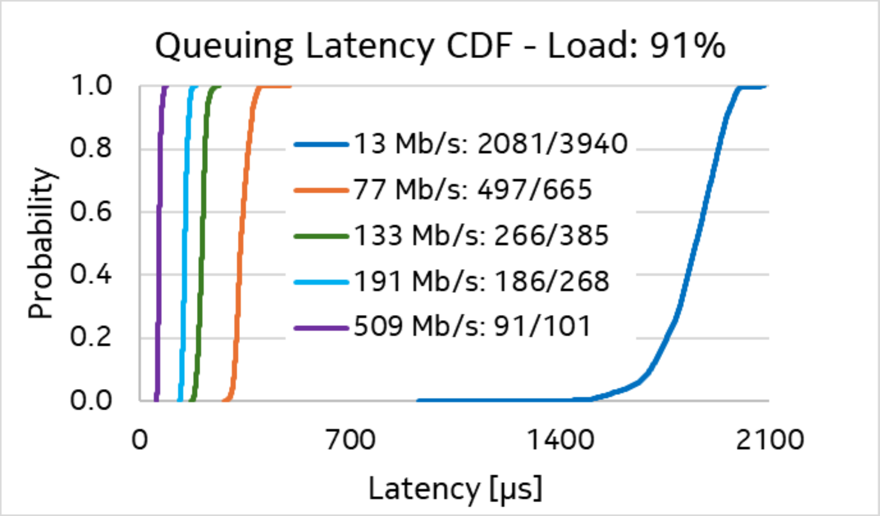
$$D_i^{\tau} = 2T \sum_{l \in \Pi_i} n_l$$

3. Multi-domain latency (true if 2. is true)



$$D_i^{M} = \frac{b_i}{r_i} + 2T \sum_{k=1}^{M} \left(\frac{C}{r_i} + \sum_{l \in \Pi_i^k} n_l \right)$$

Results from the testbed: Ingress shaping latency



Upper bound on queuing latency of ingress shaper:

$$D_I = D_{I,b} + D_{I,s} = \frac{2T}{r_i/C} + \frac{2T}{r_i/C} = \frac{4T}{r_i/C}$$

Where:

- T is the timeslot duration in the TDM frame
- r_i is the shaping rate of flow *I*
- C is the capacity of the ingress link

Test: non-uniform traffic from *diverse* flows, 91% load, 4% service rate *overprovisioning*

RAS highlights:

(a) Results in line with conjectured bound

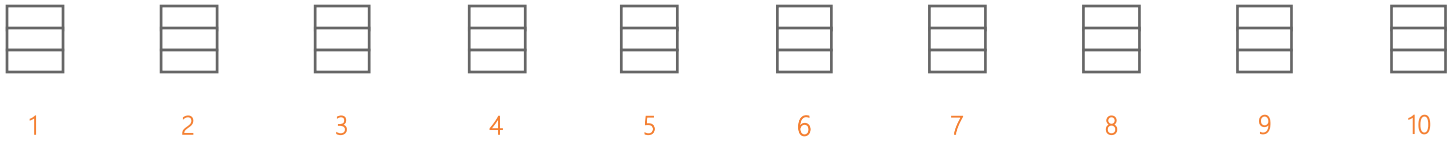
(b) No latency penalty compared to conventional shaper

NOKIA BELL LABS

The RAS: how we make it feasible

A periodic TDM frame configured after offline execution of the shaping algorithm

2	6	3	1	2	10	9	-	2	6	4	8	2	7	5	-



A centralized controller computes the TDM frame at flow creation timescale
- Width of hierarchy can be arbitrarily large

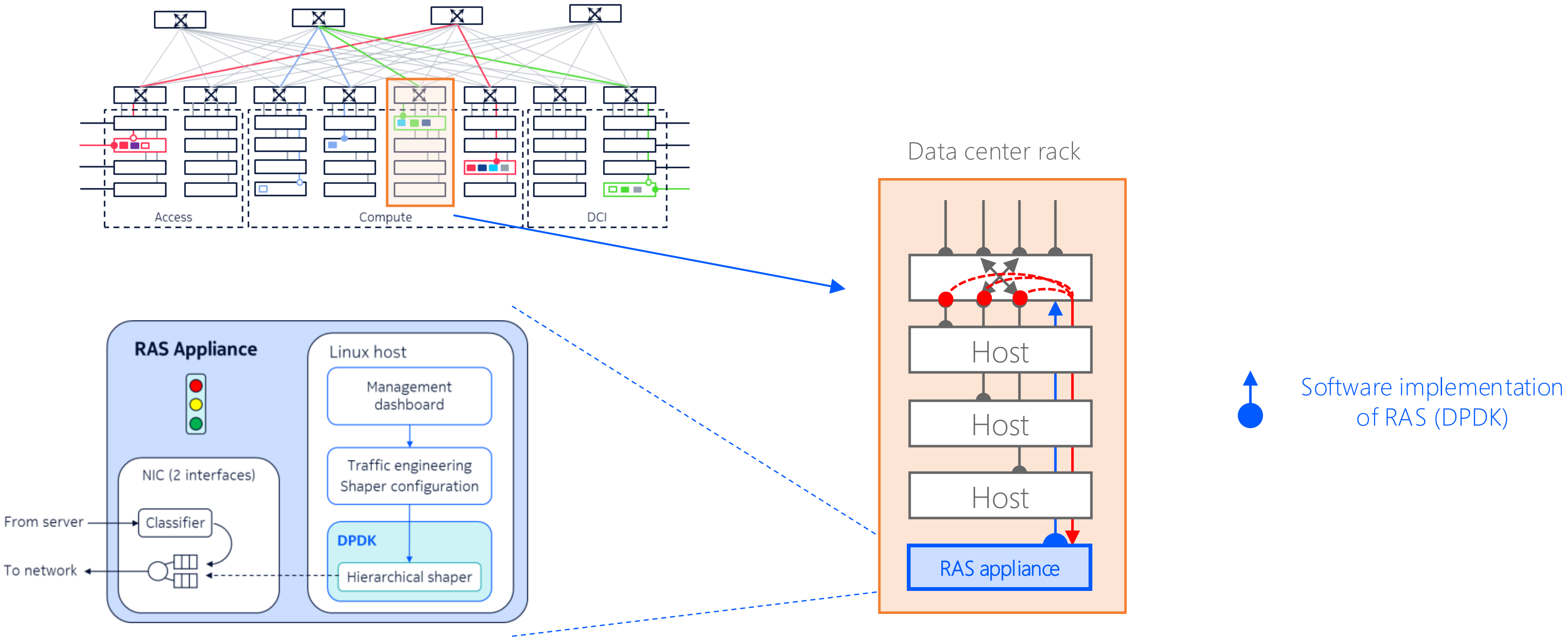
Timeslot has fixed duration, larger than largest packet (e.g., 1600 bytes): one designated queue per timeslot
- Packet transmissions cross timeslot boundaries without degrading fairness

Effective handling low-throughput low-latency flows
- Flow bundling scheme mitigates over-allocation of service rates

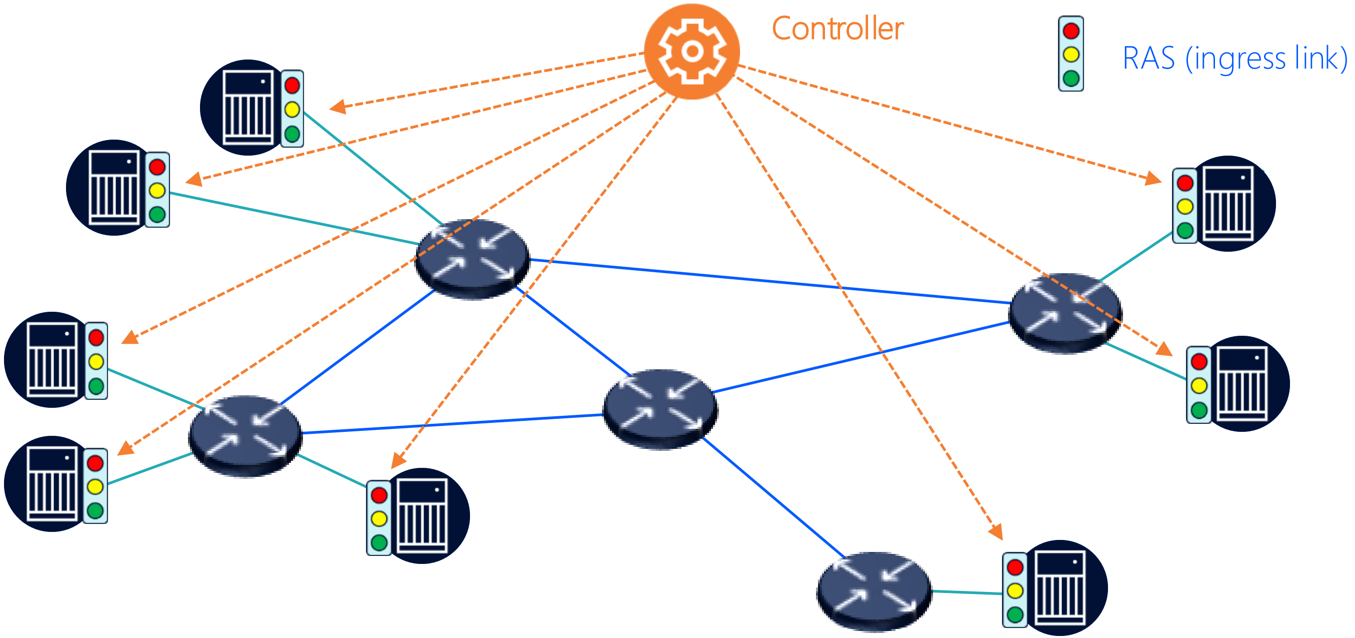
NOKIA BELL LABS

The RAS: data center deployment

One or more **RAS appliances** per rack

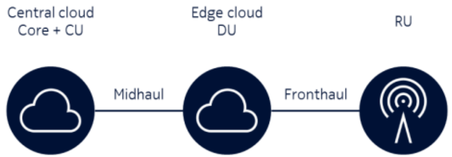


Overall solution: Determinism as a Service (DaaS)



The RAS and the controller can be implemented as software modules and deployed on demand, as a service

Who needs determinism?



Edge-Cloud Applications

Round-trip time (RTT) impacts interactivity (10 km = 100 µs)

Critical KPI: latency in access, transport, and data center networks

5G/6G Disaggregation

Inter-function latency adds to data-plane latency

Critical KPI: latency in core and edge data centers and Xhaul transport segments

HPC, AI Workflows

Inter-node latency adds to compute time and energy consumption

Critical KPI: latency in data center network

NOKIA BELL LABS

Bound validation: RAS evaluation testbed

Leaf-spine data center fabric with 8 servers, all links are 25 Gb/s