Elastic and Reliable Compute for 6G: From Server Farm Design to Virtualized RAN

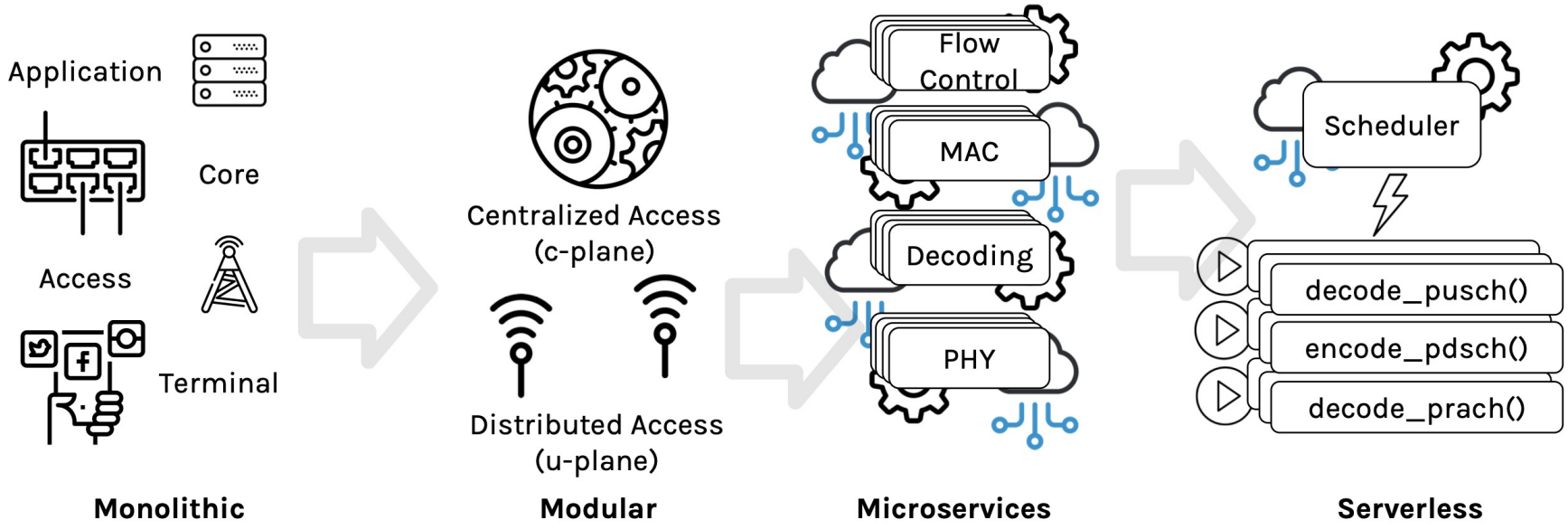
Pablo Serrano
pablo@it.uc3m.es

Huawei Future of Wireless Workshop 2026

uc3m | Universidad Carlos III de Madrid

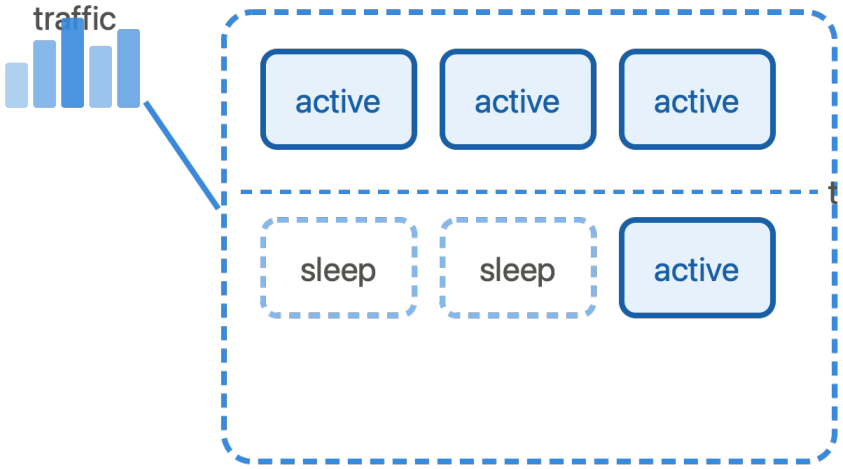
THE CHALLENGE

Toward Elastic Mobile Networks



[1] M. Gramaglia et al. “The case for serverless mobile networking”. IFIP Networking 2020 (Network Slicing Workshop)

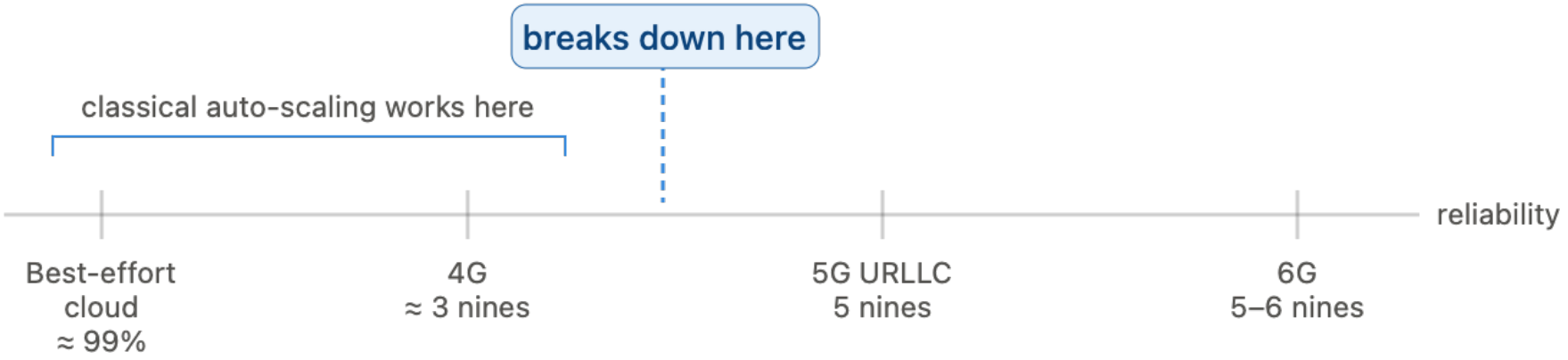
The Efficiency Imperative



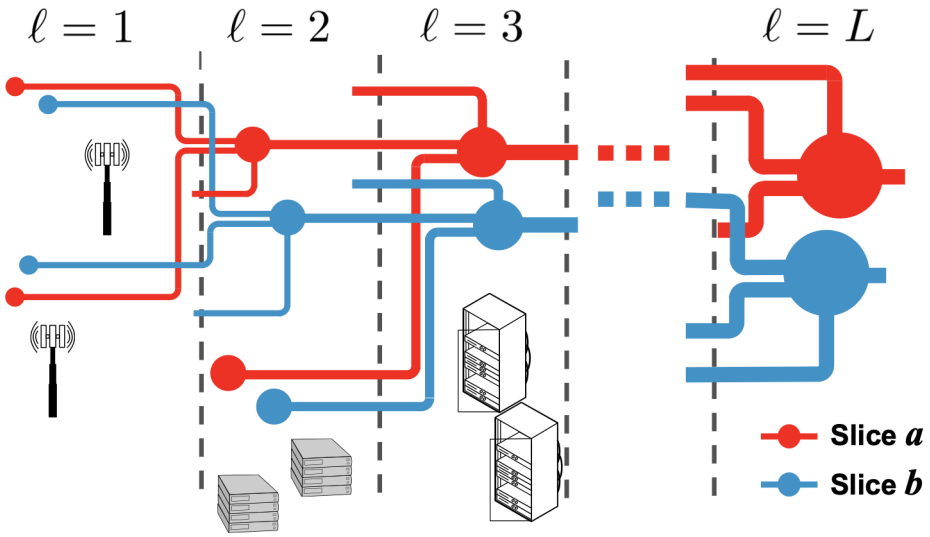
[2] J. Perez-Valero et al. “Performance Trade-offs of Auto Scaling Schemes for NFV with Reliability Requirements”. Elsevier Computer Networks 2023

The Reliability Wall

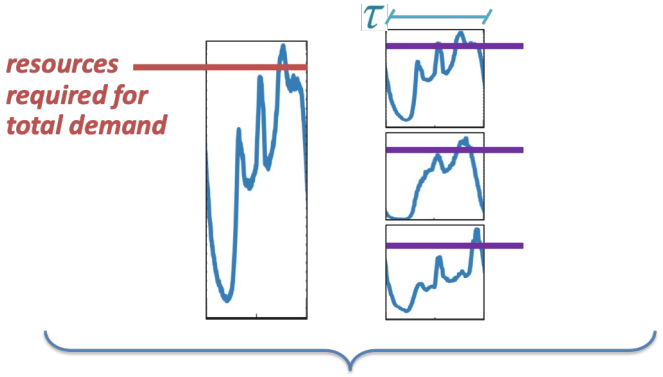
- Servers are fallible: they fail with non-negligible probability
- Activation is not instantaneous: servers take time to boot
- Traffic is bursty: demand spikes must be anticipated, not just reacted to

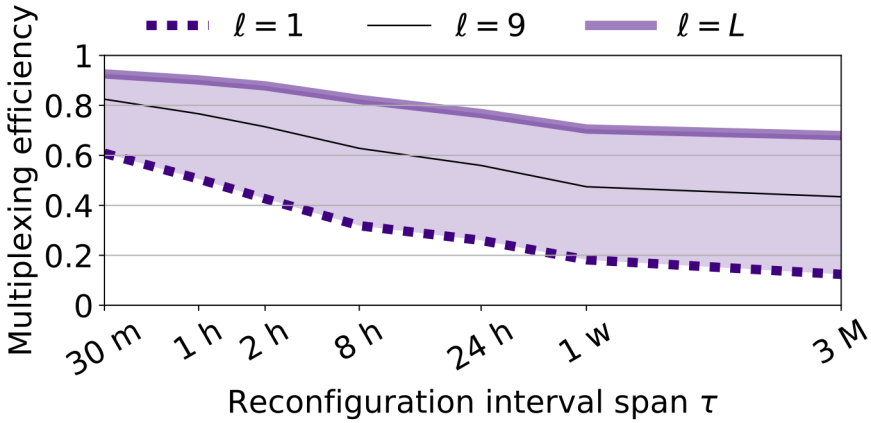


Sharing Efficiency Across the Hierarchy



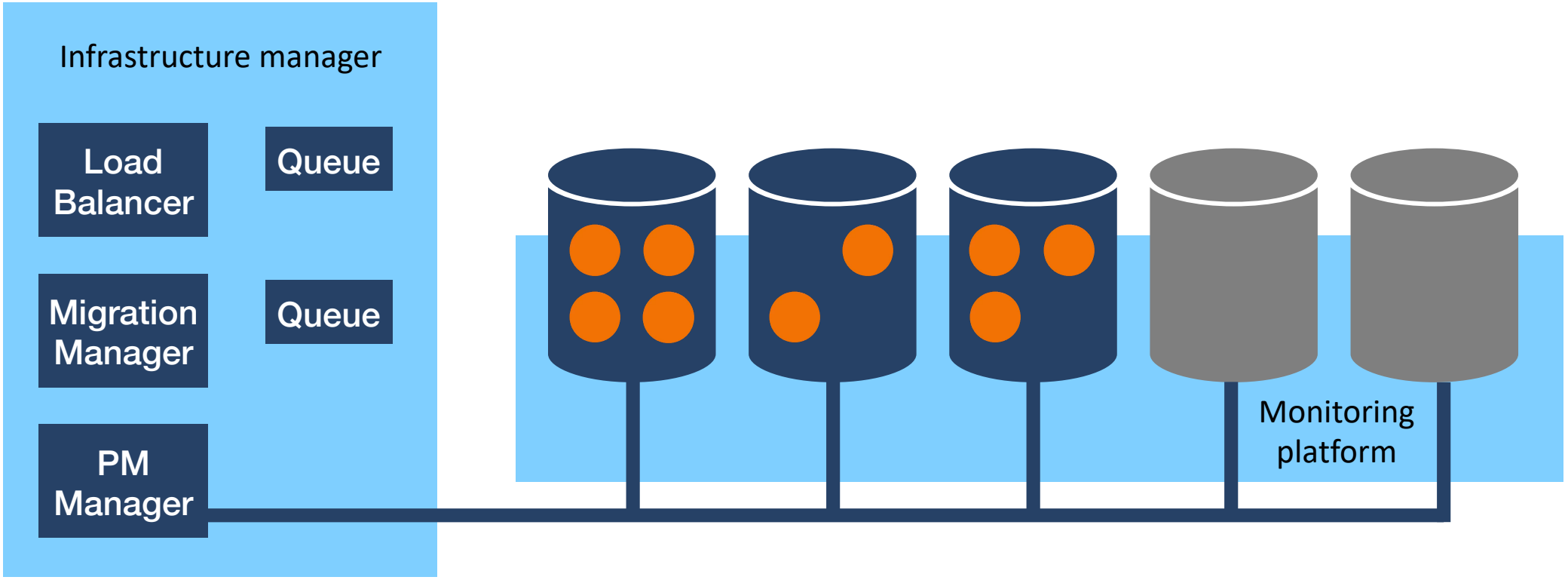
[3] C. Marquez et al., "How should I slice my network? A multi-service empirical evaluation of resource sharing efficiency," ACM MobiCom 2018, New Delhi, India





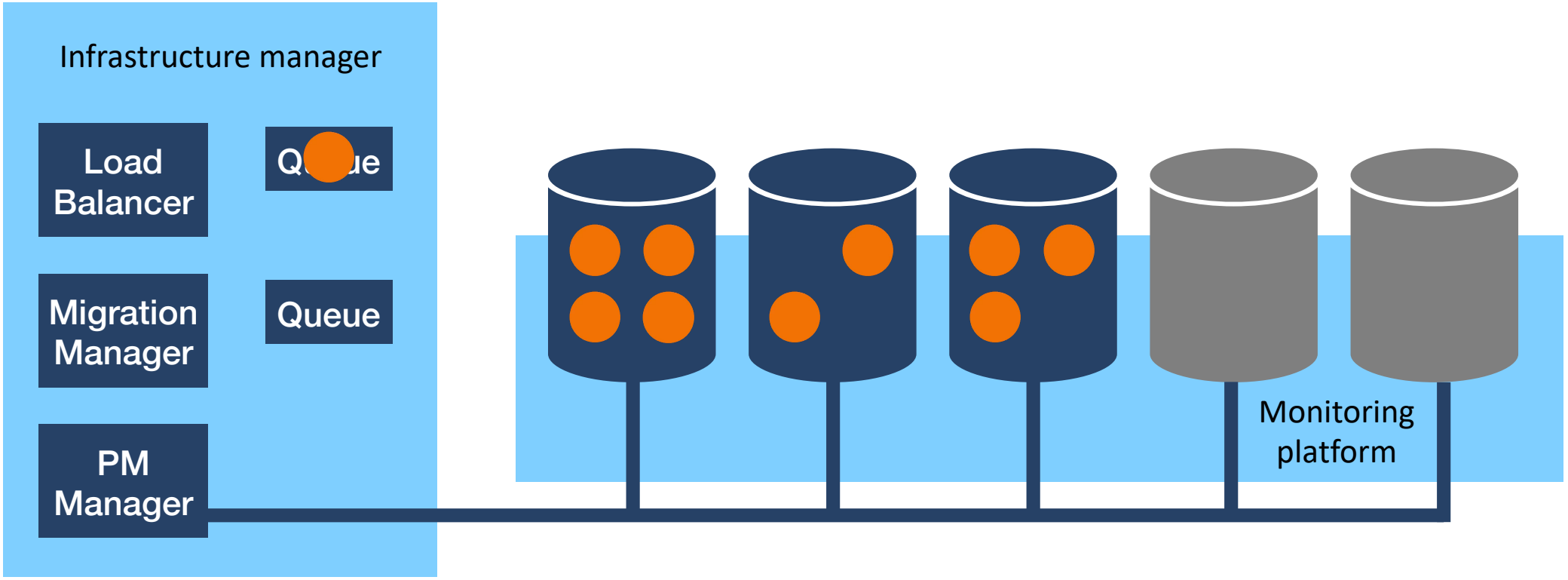
A LOOK INTO THE TRADE-OFFS

Server Model



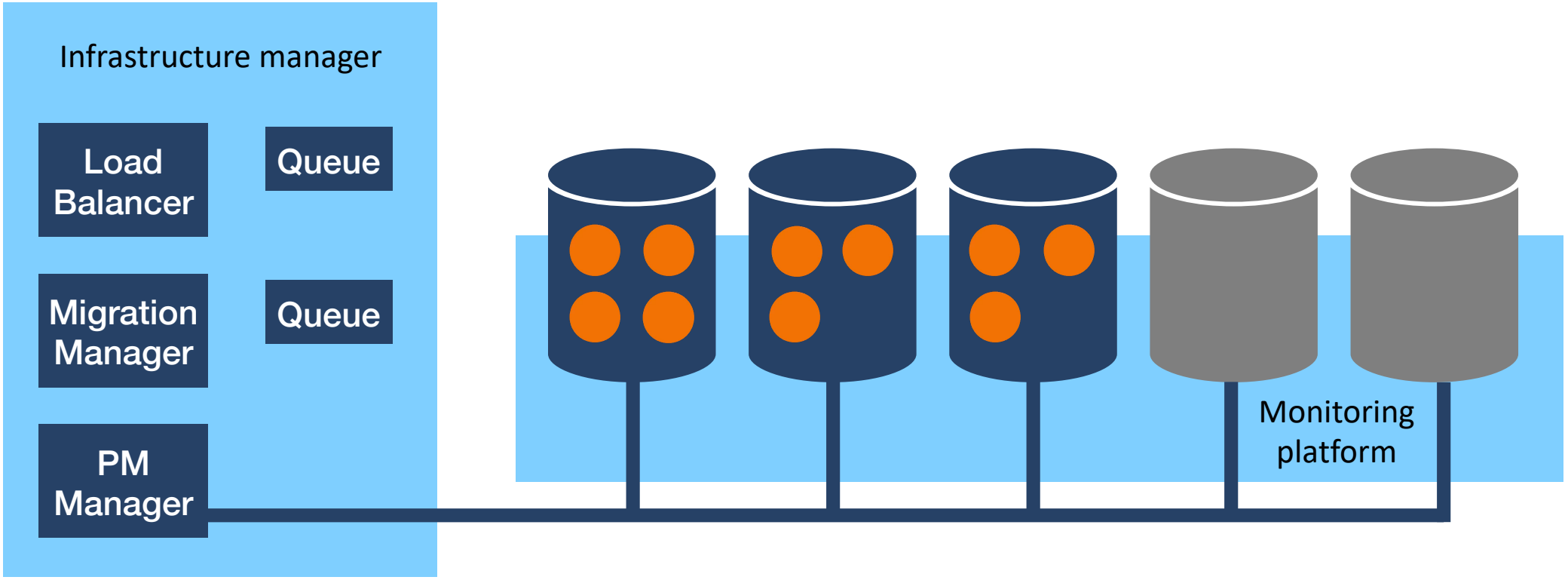
[2] J. Perez-Valero et al. “Performance Trade-offs of Auto Scaling Schemes for NFV with Reliability Requirements”. Elsevier Computer Networks 2023

Server Model

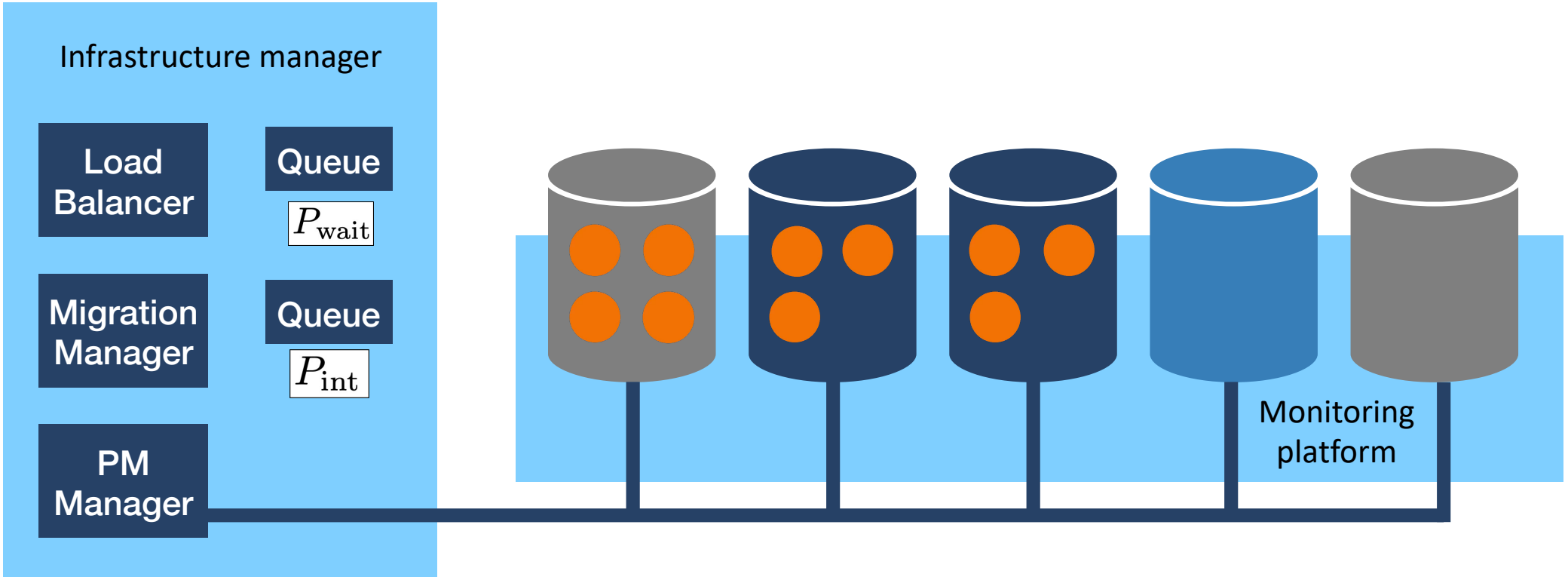


[2] J. Perez-Valero et al. “Performance Trade-offs of Auto Scaling Schemes for NFV with Reliability Requirements”. Elsevier Computer Networks 2023

Server Model

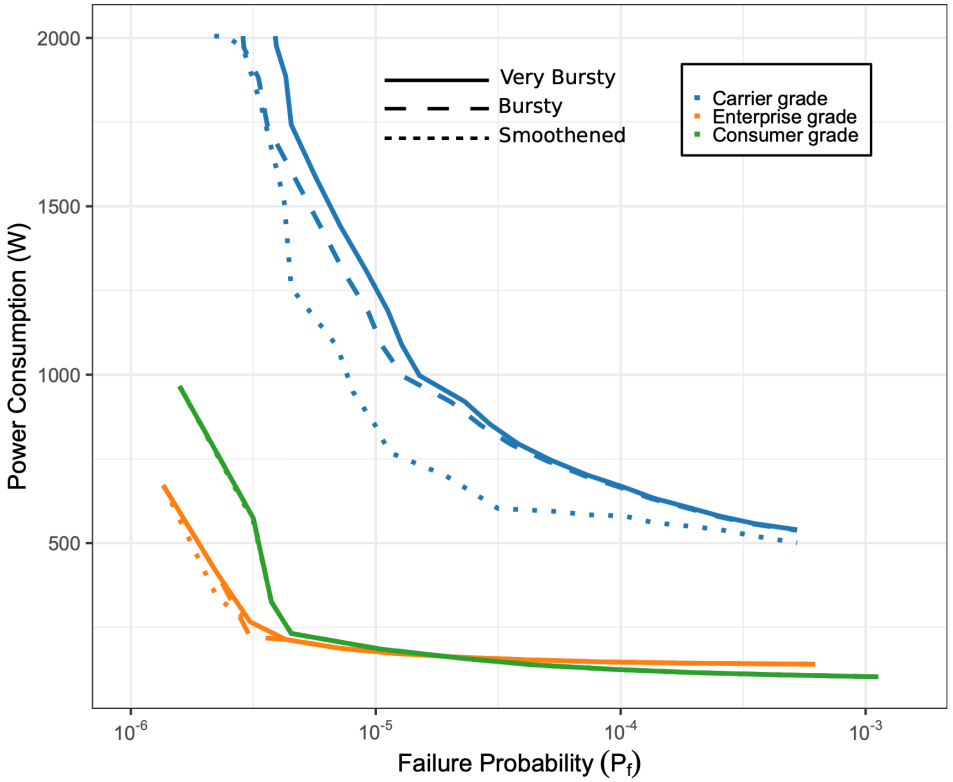


Server Model



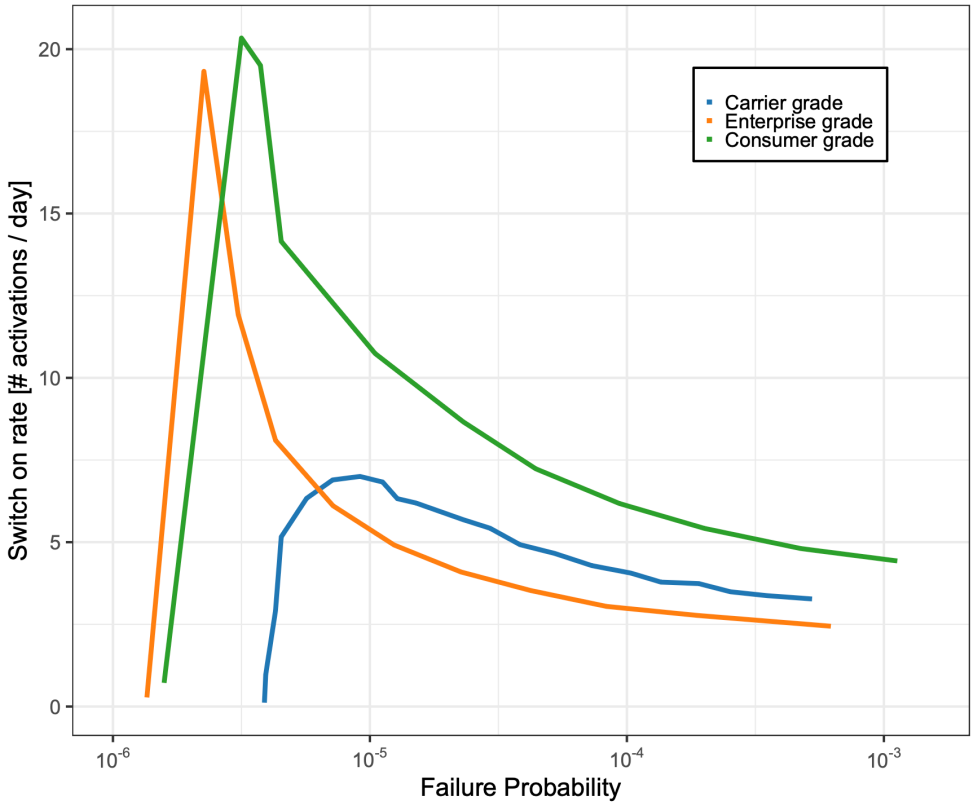
Pareto Frontiers

	Carrier	Enterprise	Consumer
RAM Capacity (GB)	16	8	8
CPU	Intel Xeon Silver 4310	Intel Core i5-8259U	BCM2711
CPU Capacity (Relative)	424	46	1
Power consumption (P_{idle})	150 W	5.1 W	4.6 W
Power consumption (P_{peak})	270 W	78 W	7.6 W
Boot time ($1/\alpha$)	3 min	8 s	18 s
Mean time to failure (MTTF)	32 days	16 days	8 days
Total # servers (N)	13	104	208
Total cost (relative)	22.1×	11.4×	1
Peak power consumption	3.5 kW	8.1 kW	1.6 kW



Other variables

	Carrier	Enterprise	Consumer
RAM Capacity (GB)	16	8	8
CPU	Intel Xeon Silver 4310	Intel Core i5-8259U	BCM2711
CPU Capacity (Relative)	424	46	1
Power consumption (P_{idle})	150 W	5.1 W	4.6 W
Power consumption (P_{peak})	270 W	78 W	7.6 W
Boot time (1/α)	3 min	8 s	18 s
Mean time to failure (MTTF)	32 days	16 days	8 days
Total # servers (N)	13	104	208
Total cost (relative)	22.1×	11.4×	1
Peak power consumption	3.5 kW	8.1 kW	1.6 kW



No Universal Optimum

- The right server type depends jointly on three factors: the traffic profile, the reliability target, and the cost model
- No single type dominates across all scenarios
  - At very high reliability targets, carrier-grade servers require enormous power budgets
  - Consumer-grade servers achieve the same target at a fraction of the cost, but require many more units
- This motivates the need for a design-time optimization framework
  - Given your e.g. SLA, your traffic model, and your cost function, what is the optimal configuration?

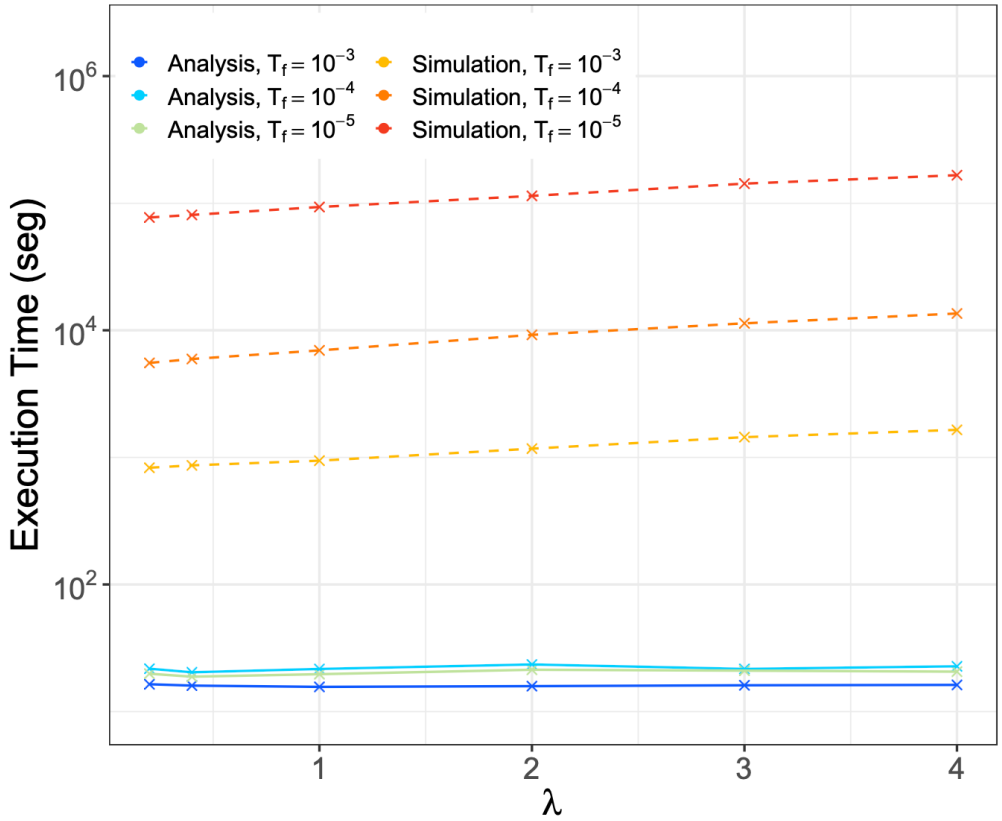
ALGORITHMS FOR DESIGN AND OPERATION

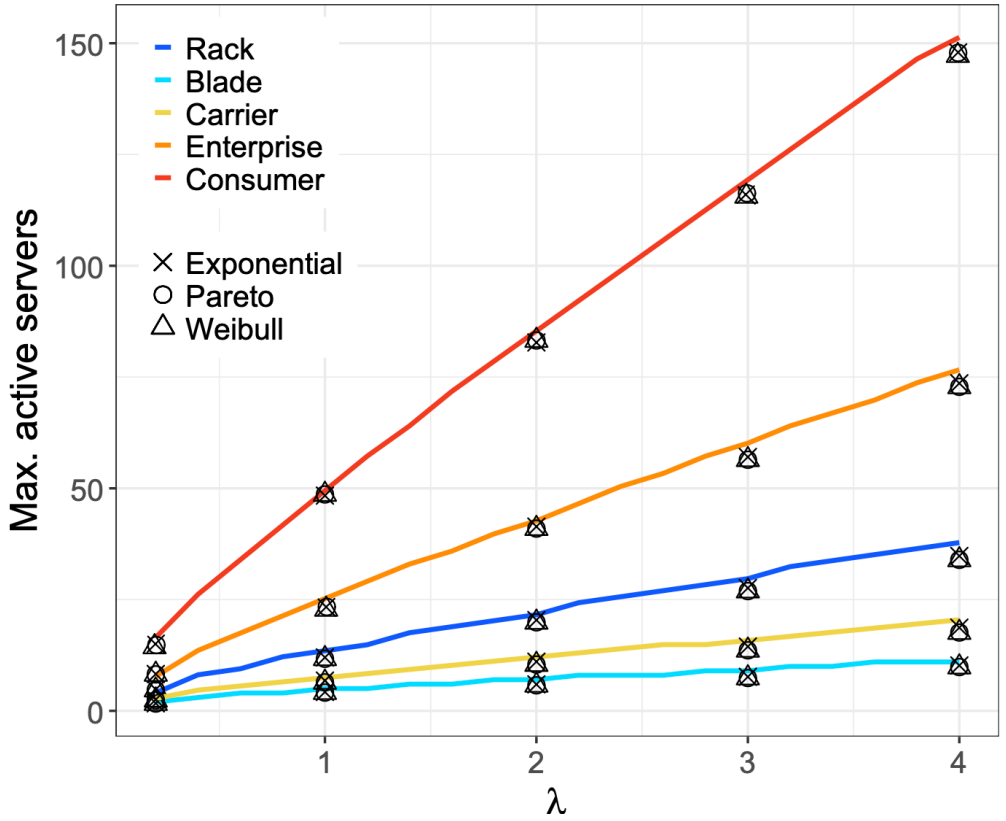
Designing the Optimal Server Farm

- Given a service SLA, what is the optimal server type, fleet size, and activation threshold?
- This is a design-time decision: made once, before deployment, based on expected operating conditions
- The design space is large
  - A brute-force search is computationally intractable
- We solve this analytically: queueing-theoretic model + cost optimization via Multiple-Choice Knapsack Problem

[4] J. Perez-Valero et al. “Minimum-Cost Design of Auto-Scaling Server Farms Providing Reliability Guarantees”. IEEE OJCOMS, 2025

Results


Analysis, $T_f = 10^{-3}$
Analysis, $T_f = 10^{-4}$
Analysis, $T_f = 10^{-5}$
Simulation, $T_f = 10^{-3}$
Simulation, $T_f = 10^{-4}$
Simulation, $T_f = 10^{-5}$
Execution Time (seg)
10^6
10^4
10^2
1
2
3
4
λ

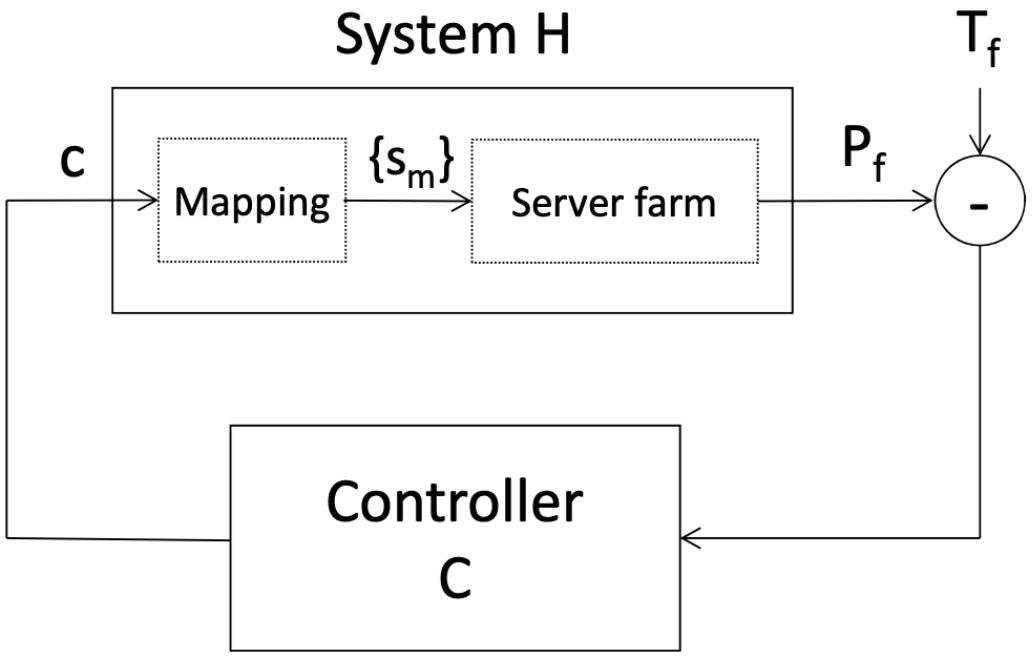

Rack
Blade
Carrier
Enterprise
Consumer
Exponential
Pareto
Weibull
Max. active servers
150
100
50
0
1
2
3
4
λ

Results

λ	T_f	Simulation Cost (C_s)					Analysis			Benchmark	
		Carrier	Enterprise	Consumer	Rack	Blade	τ*	C_a	ΔC	C_b	ΔB
0.2	10^{-3}	495.98	256.26	665.26	855.40	260.03	Blade	260.62	1.70%	388.35	49.01%
	10^{-4}	498.17	268.04	668.52	858.25	261.73	Enterprise	265.13	1.30%	408.30	54.00%
	10^{-5}	502.53	275.78	671.83	861.07	265.21	**Blade**	269.69	1.66%	474.01	75.76%
0.4	10^{-3}	824.32	712.03	1006.47	1569.52	706.90	**Blade**	716.08	1.28%	776.71	8.47%
	10^{-4}	828.42	763.08	1110.23	1573.58	710.14	**Enterprise**	766.43	0.44%	840.32	9.64%
	10^{-5}	835.07	770.93	1125.36	1580.05	770.55	**Blade**	787.81	2.24%	1084.30	37.63%
1.0	10^{-3}	1634.76	1557.30	1360.30	1860.12	1631.93	**Consumer**	1365.19	0.36%	1516.15	11.06%
	10^{-4}	1840.56	1729.00	1747.32	1865.57	1731.43	**Enterprise**	1747.15	1.05%	1860.23	6.47%
	10^{-5}	2055.09	2028.34	2022.25	2056.09	2031.67	**Consumer**	2189.73	6.50%	2430.62	11.00%
2.0	10^{-3}	2995.68	3131.26	2804.93	2812.14	3121.38	**Consumer**	3008.56	7.26%	3118.77	3.66%
	10^{-4}	3010.08	3150.12	2899.54	2892.20	3135.09	**Rack**	3051.56	5.51%	3343.76	9.58%
	10^{-5}	3140.14	3175.16	3126.13	3198.45	3150.17	**Consumer**	3219.60	2.99%	3684.36	14.44%
3.0	10^{-3}	3861.44	3891.09	3917.21	3901.75	4611.36	**Carrier**	4100.84	6.20%	4888.86	19.22%
	10^{-4}	3975.12	4005.11	3990.17	3947.93	4625.20	**Rack**	4148.09	5.07%	5120.98	23.45%
	10^{-5}	4210.21	4188.63	4147.31	4167.34	4660.33	**Consumer**	4333.52	4.49%	5876.56	35.61%
4.0	10^{-3}	5321.68	5561.98	5432.71	5412.82	5871.76	**Carrier**	5587.76	5.01%	6006.73	7.50%
	10^{-4}	5640.19	5580.29	5559.92	5512.15	5890.42	**Rack**	5643.89	2.39%	6345.41	12.43%
	10^{-5}	5980.31	5821.87	5820.19	5847.63	5930.54	**Consumer**	5899.34	1.36%	6640.85	12.57%

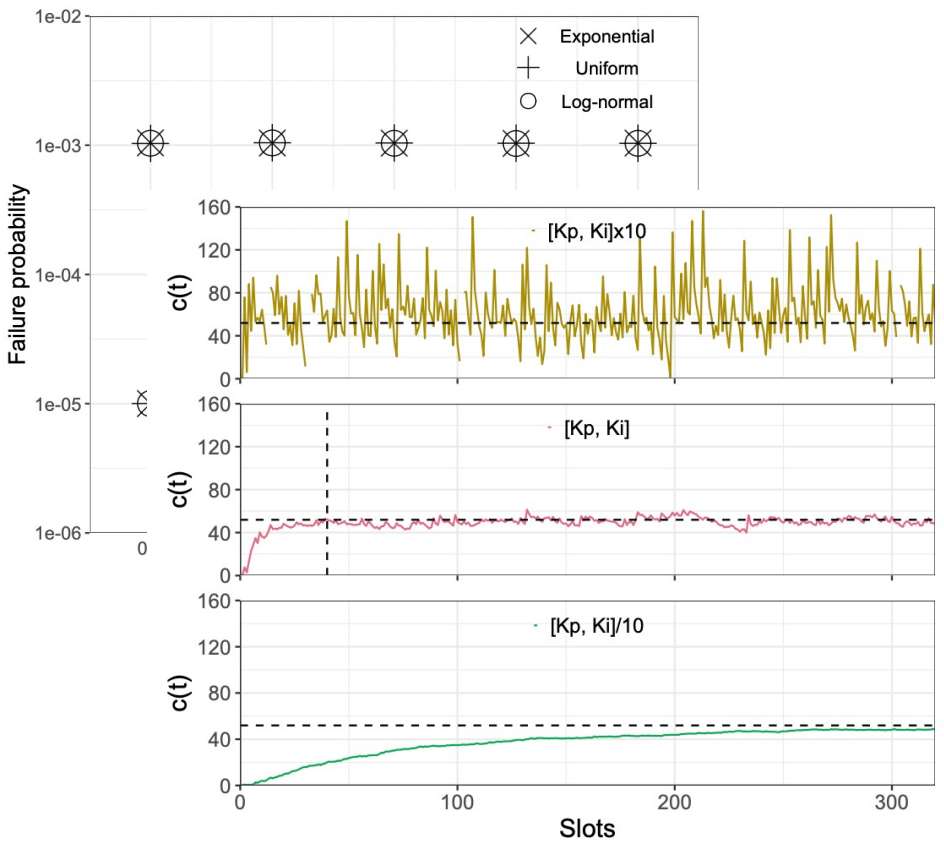
A3S — Adaptive Algorithm for Auto-Scaling

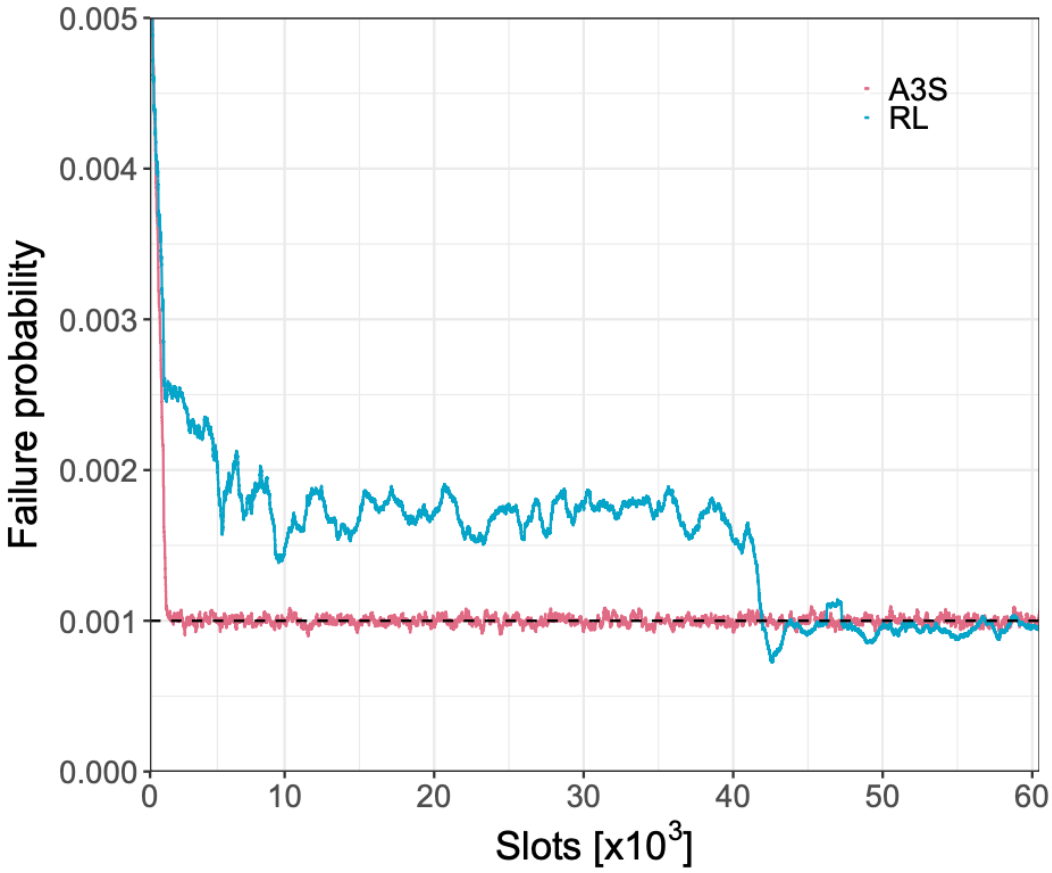
- In practice, traffic is non-stationary
  - Opt. Thresholds change
- A3S is a closed-loop control system
  - Measures Pf
  - Compares it to the target
  - Adjusts the thresholds
- PI Controller



[5] J. Perez-Valero et al. “Energy-Aware Adaptive Scaling of Server Farms for NFV with Reliability Requirements”. IEEE Trans. Mobile Computing, 2023

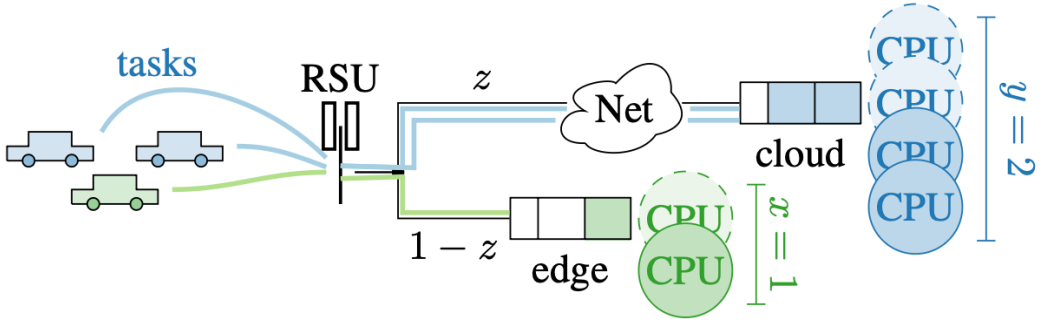
Results


Exponential
Uniform
Log-normal
Failure probability
1e-02
1e-03
1e-04
1e-05
1e-06
0
[Kp, Ki]x10
[Kp, Ki]
[Kp, Ki]/10
c(t)
160
120
80
40
0
100
200
300
Slots


A3S
RL
Failure probability
0.005
0.004
0.003
0.002
0.001
0.000
0
10
20
30
40
50
60
Slots [x10^3]

Edge-Cloud Offloading for URLLC V2N

- The server farm problem extended to distributed scenarios (Edge + Cloud)
- Vehicles generate URLLC traffic that can be processed locally at the edge or offloaded to the cloud



- Jointly optimizing across both: the V2N-COCA problem

[7] L. E. Chatzieleftheriou et al. “Optimal Scaling and Offloading for Sustainable Provision of Reliable V2N Services in Dynamic and Static Scenarios”, IEEE TNSM 2025

Contributions

- Minimizing operational costs while deciding CPU scaling and edge offloading for URLLC V2N video tasks.

$$\mathbb{P}(D_i(\cdot) \leq T) \geq P_G, \forall \text{ task } i. \;\; D_i(V,x,y,z) = D_{tran}(l_i) + D_{prop} + D_{sojo}(V,x,y,z),$$
$$K(V,x,y,z) := c_{0e}x + c_{0c}y + c_{1c}\lambda_C(z,V)s + c_{1e}\lambda_E(z,V)s, \tag{5}$$

- BiQui: provably asymptotically optimal algorithm with low complexity that we extensively evaluate over real-traces [6] MobiHoc
- Dynamic scenarios where computing load and resource availability change [7] IEEE TNSM
  - Swap-Prevention problem: minimize CPU on/off switching frequency.
  - We study the problem's structural properties.
  - Min-Swap algorithm to solve it

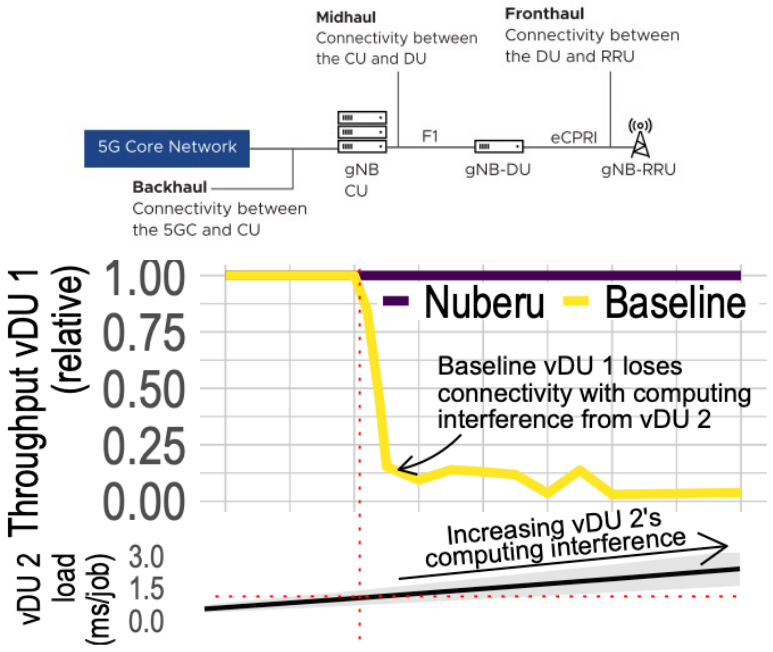
Results

Reliability Requirement	Load (Erlang)					
	l=1	l=2	l=4	l=8	l=16	l=32
100ms, 99.999%	(3,3)	(4,4)	(6,6)	(10,10)	(18,18)	(34,34)
100ms, 99.9%	(2,2)	(3,3)	(5,5)	(9,9)	**(18,17)**	(34,34)
50ms, 99.9%	(3,3)	(5,5)	(7,7)	(11,11)	**(20,19)**	**(36,35)**
50ms, 99%	(3,3)	(4,4)	(6,6)	(10,10)	(18,18)	**(35,34)**

RELIABLE VRAN IN SHARED PLATFORMS

vRAN - Sharing may cost connectivity

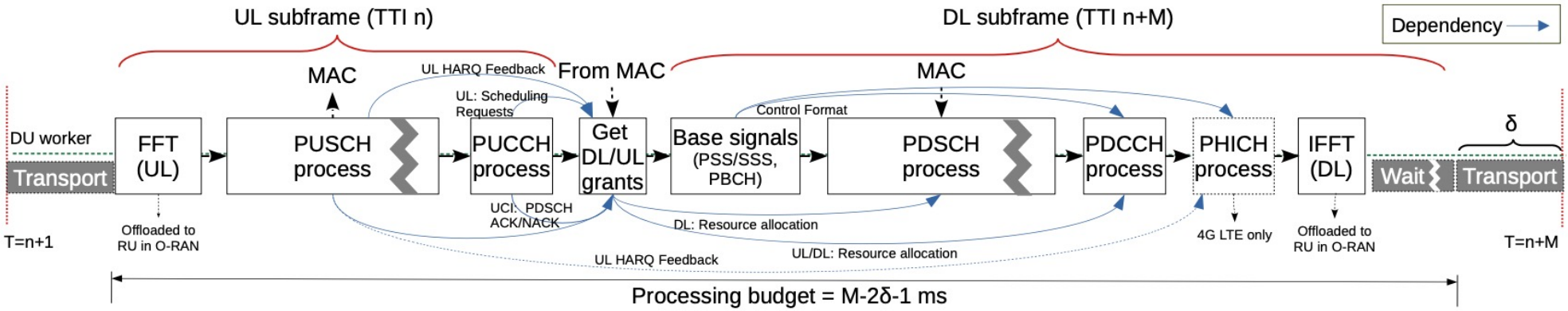
- O-RAN disaggregates the base station into a Central Unit, a Distributed Unit, and a Radio Unit
  - The DU hosts the physical layer and is the most compute-intensive component
  - Shared computing platforms are attractive for DU hosting
- But shared platforms introduce non-deterministic compute capacity (noisy neighbor)
- The DU pipeline operates under hard timing constraints
  - Every transmission time interval (TTI) of 1ms must be processed and delivered on time
  - Missing a deadline = disconnection



Two vDUs competing for computing resources. The UL/DL data load of vDU 1 is the highest possible while vDU 2's is variable

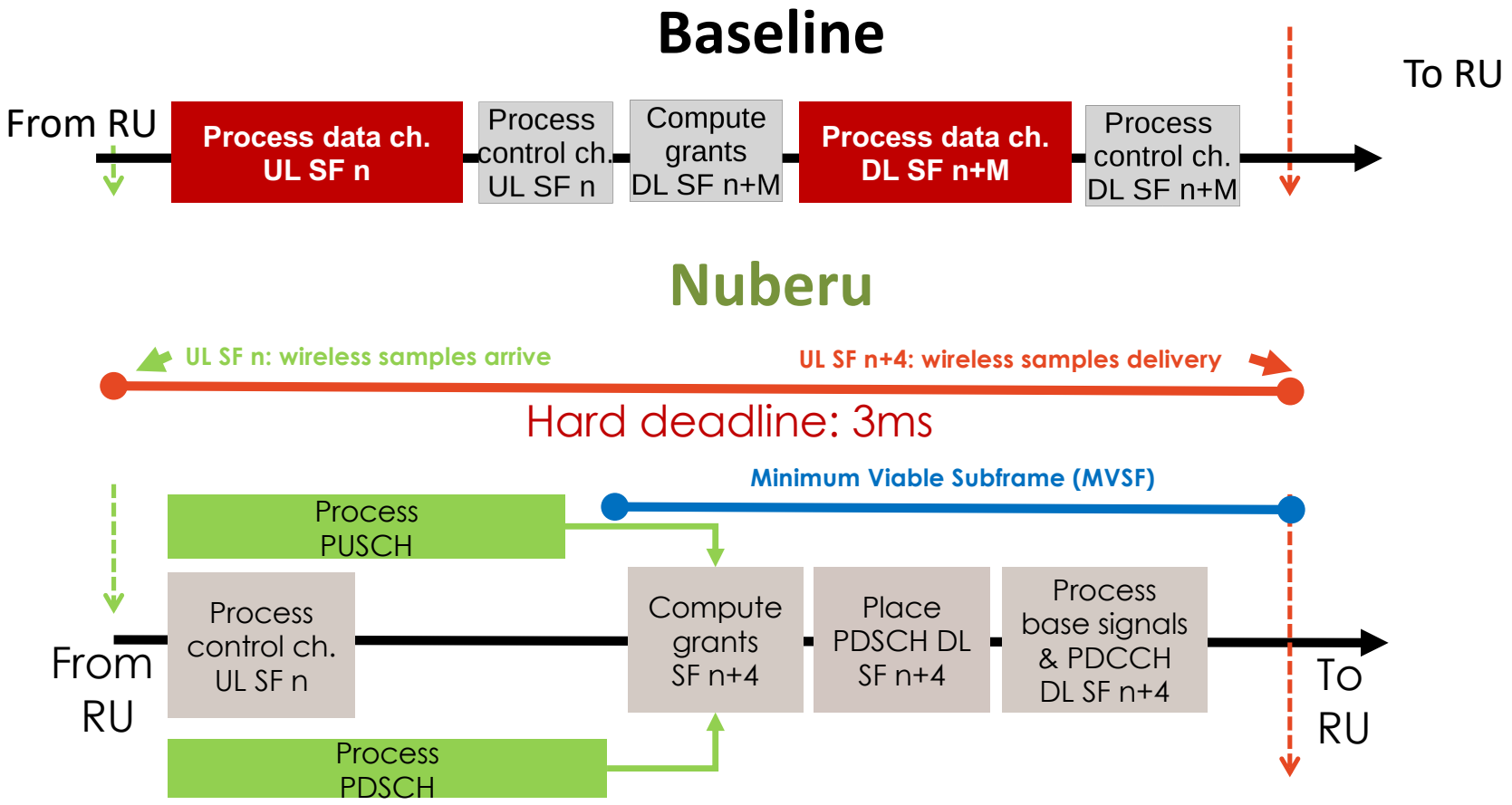
[8] G. Garcia-Aviles et al. "Nuberu: Reliable RAN Virtualization in Shared Platforms", ACM MobiCom 2021

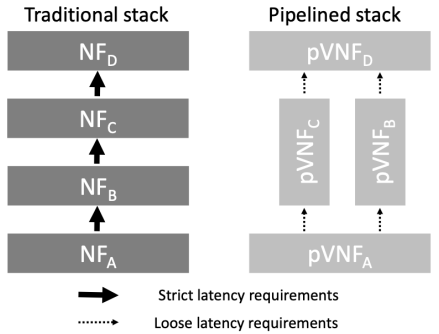
The Challenge: pipeline of tasks



- DL grants must be computed before PDSCH because they carry information required to encode and modulate DL
- PUSCH and PUCCH must be processed before computing DL grants because they carry users' feedback (DL HARQ, channel quality, etc.) in UCI messages (dotted arrows in Fig. 7), which is required to schedule DL data appropriately;
- UL HARQ feedback can only be computed after processing PUSCH; and this feedback is required to schedule UL grants within the same job to satisfy 3GPP timing constraints;
- UL grants must be computed before processing PDCCH, which carries those grants, and (in case of LTE) before processing PHICH, which carries UL HARQ feedback.

The Solution



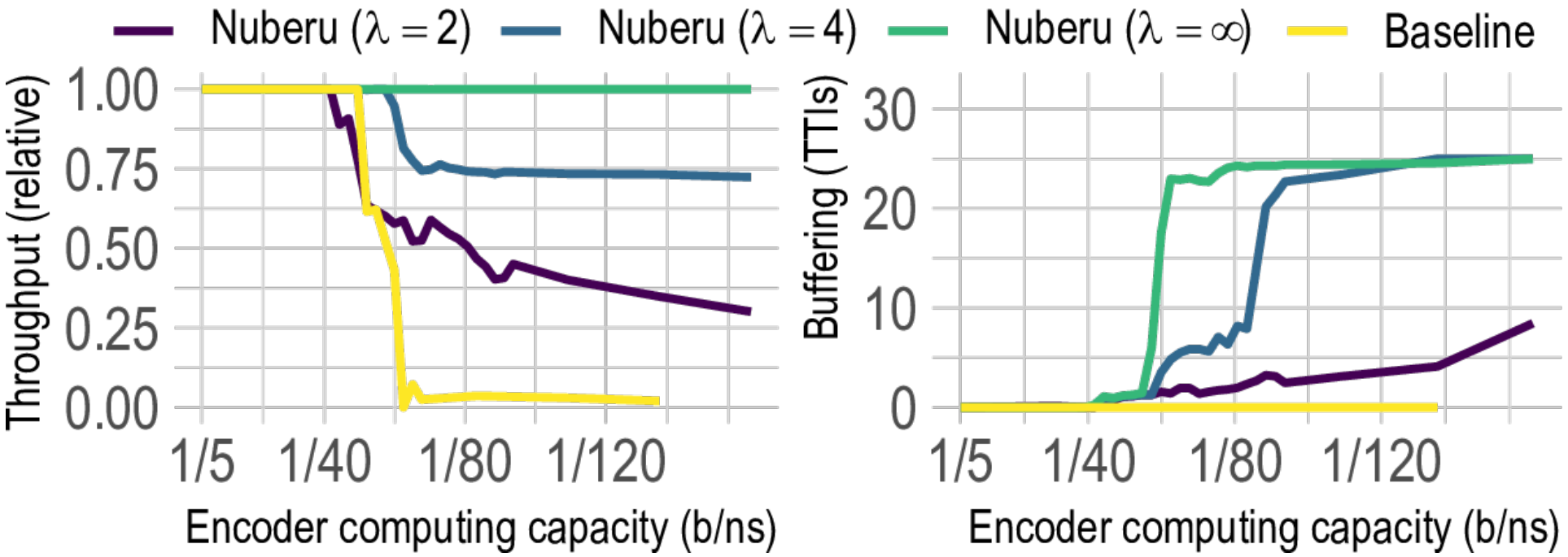


[9] P. Serrano et al. "The path towards a cloud-aware mobile network protocol stack", Trans. Emerging Telecomm. Tech., 2018

Three key mechanisms

- Tight deadline control: data processing tasks are decoupled into separate threads with a hard MVSF deadline
  - If data workers do not finish in time, the MVSF is transmitted anyway, preserving synchronization
- Predictive HARQ (E-HARQ): rather than waiting, predict whether a transport block will decode successfully
  - Reducing compute requirements by up to 35x for some MCS
- Congestion control: a rate limiter prevents the PHY buffer from growing unboundedly
  - Throughput vs. stability/delay

Results


Nuberu (λ = 2)
Nuberu (λ = 4)
Nuberu (λ = ∞)
Baseline
Throughput (relative)
1.00
0.75
0.50
0.25
0.00
1/5
1/40
1/80
1/120
Encoder computing capacity (b/ns)
Buffering (TTIs)
30
20
10
0
1/5
1/40
1/80
1/120
Encoder computing capacity (b/ns)

SUMMARY AND OPEN PROBLEMS

Summary

- 6G elastic compute is a hard problem: reliability walls, energy pressure, and multi-service slicing cannot be addressed independently
- We developed a toolchain spanning the full stack:
  - **Design:** minimum-cost server farm configuration given an SLA and traffic profile
  - **Adapt:** A3S closed-loop control that outperforms reinforcement learning in convergence speed
  - **Offload:** BiQui and Min-Swap for near-optimal edge-cloud splitting with minimal CPU switching
  - **RAN edge:** a redesigned DU pipeline that achieves near-optimal spectrum efficiency where the baseline loses connectivity
- These foundations are the foundations towards a serverless mobile architecture

Open Problems and Future Directions

- **Heterogeneous server farms across slices:** the models optimize a single service in isolation (apart from simple cases). Extending to shared elastic pools serving multiple slices simultaneously, with coupled reliability constraints, remains open
- **End-to-end latency modeling:** the server farm and RAN pipeline models are currently separate. Challenge: a unified model integrating compute delay, radio scheduling, and transport latency is needed to make end-to-end URLLC guarantees
- **O-RAN integration:** the redesigned DU pipeline and elastic provisioning algorithms need to be validated on real O-RAN platforms. This requires testbed infrastructure and industry partnership
- **AI/ML-assisted orchestration:** machine learning can complement analytical guarantees rather than replace them, e.g., hybrid approaches that use ML for prediction and control theory for stability are a promising direction
- **Additionally:** use of AI to harden these systems against security threats and enable stakeholders to cooperate (sharing insights, not raw data) through the Horizon Europe project PAISES-6G



https://paises-6g.eu

Elastic and Reliable Compute for 6G: From Server Farm Design to Virtualized RAN

Thank you!
Questions?

Pablo Serrano
pablo@it.uc3m.es

Huawei Future of Wireless Workshop 2026

uc3m | Universidad Carlos III de Madrid

References

[1] M. Gramaglia et al. “The case for serverless mobile networking”. IFIP Networking 2020

[2] J. Perez-Valero et al. “Performance Trade-offs of Auto Scaling Schemes for NFV with Reliability Requirements”. Elsevier Computer Networks 2023

[3] C. Marquez et al., “How should I slice my network? A multi-service empirical evaluation of resource sharing efficiency,” ACM MobiCom 2018, New Delhi, India

[4] J. Perez-Valero et al. “Minimum-Cost Design of Auto-Scaling Server Farms Providing Reliability Guarantees”. IEEE OJCOMS, 2025

[5] J. Perez-Valero et al. “Energy-Aware Adaptive Scaling of Server Farms for NFV with Reliability Requirements”. IEEE Trans. Mobile Computing, 2023

[6] L. E. Chatzieleftheriou et al. “Sustainable Provision of URLLC Services for V2N: Analysis and Optimal Configuration”, ACM MobiHoc 2024

[7] L. E. Chatzieleftheriou et al. “Optimal Scaling and Offloading for Sustainable Provision of Reliable V2N Services in Dynamic and Static Scenarios”, IEEE TNSM 2025

[8] G. Garcia-Aviles et al. “Nuberu: Reliable RAN Virtualization in Shared Platforms”, ACM MobiCom 2021

[9] P. Serrano et al. “The path towards a cloud-aware mobile network protocol stack”, Trans. Emerging Telecomm. Tech., 2018