ITEE Ph.D — Information Technology Electrical Engineering

Stefano Rosiello

Tutor: prof. Domenico Cotroneo

XXXI Cycle - III year presentation

Autonomic Overload Management for Large-Scale Virtualized Network Functions

Università degli Studi di Napoli Federico II

My Background

- **Master of Science:**
  - In **Ingegneria Informatica** at University of Naples - Federico II
- **DIETI group:**
  - Dependable Systems and Software Engineering Research Team (**DESSERT**)
- **Type of Fellowship:** PhD Student Grant
- **Industrial Collaboration:**
  - **Huawei Technologies Co. Ltd.** within an industrial research project with the aim to identify possible solutions to improve the reliability of NFV systems.

Network challenges

- In the past networks where challenged by **exceptional events**
  - The new-year
  - An earthquake …

- Now, **challenging events happens any time** and they cannot be considered exceptional anymore
  - A viral post on a social
  - An update of a popular app on the smartphone
  - A new episode of a popular TV-series in streaming …

Customers are constantly pushing for **new services** and they expect the same **level of quality**

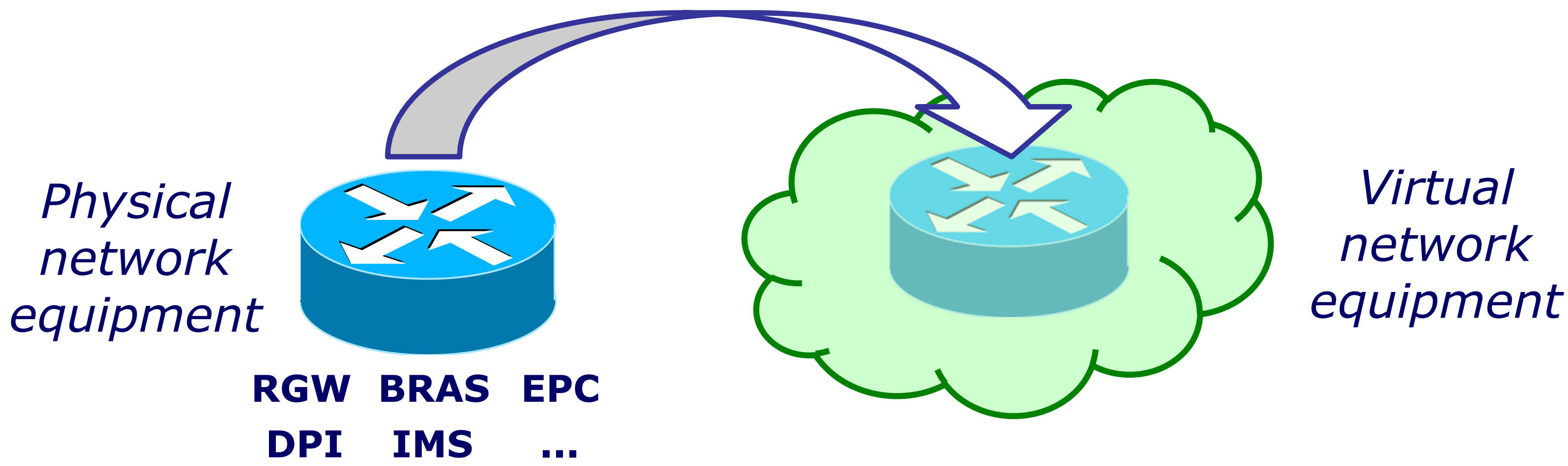
First live streaming exclusive of a sport match

- It is a predictable event
- Are the network systems prepared to this load?

- Unavailability
- Pause and buffering
- High delays
- Poor quality
- API overload errors

Network Function Virtualization

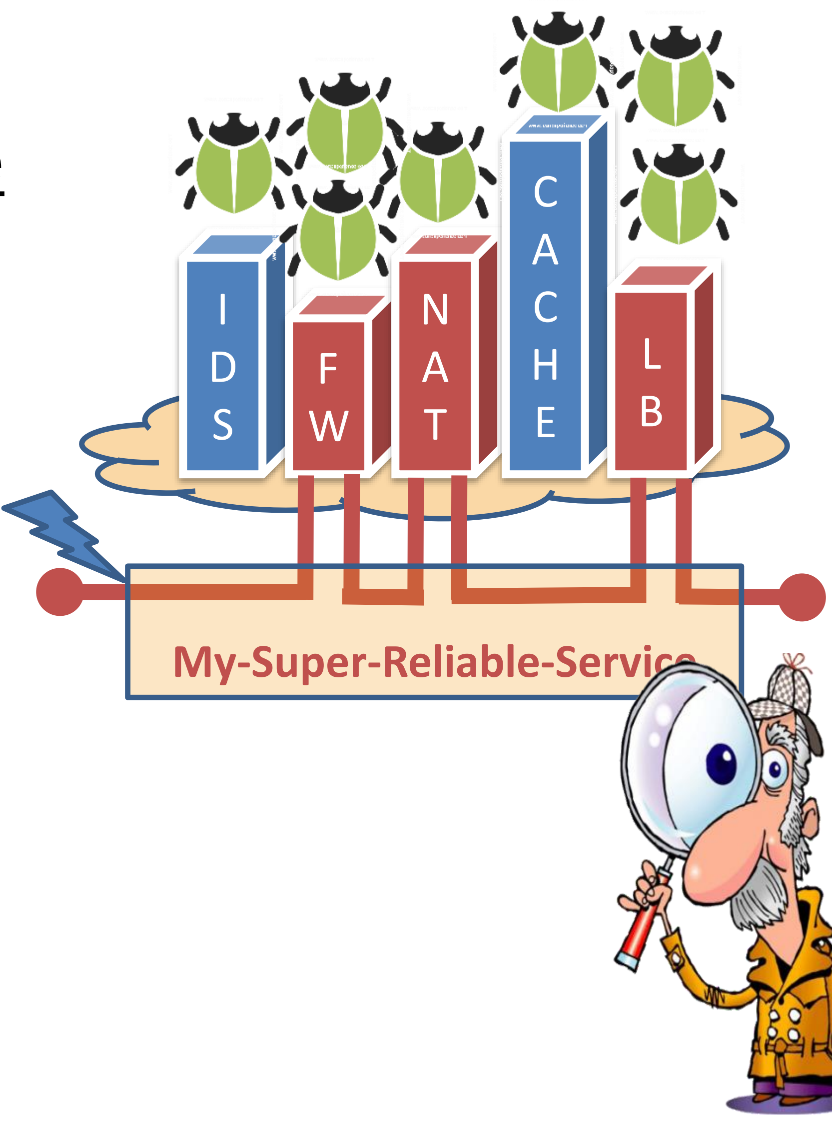
- To keep up with the high demand and staying profitable, Telcos are embracing the **Network Function Virtualization** (NFV) paradigm
  - Improve manageability of complex networks
  - Reduce time-to-market
  - are expected to support extremely large-scale architectures



The success depends upon the ability to **comply with carrier-grade requirements**

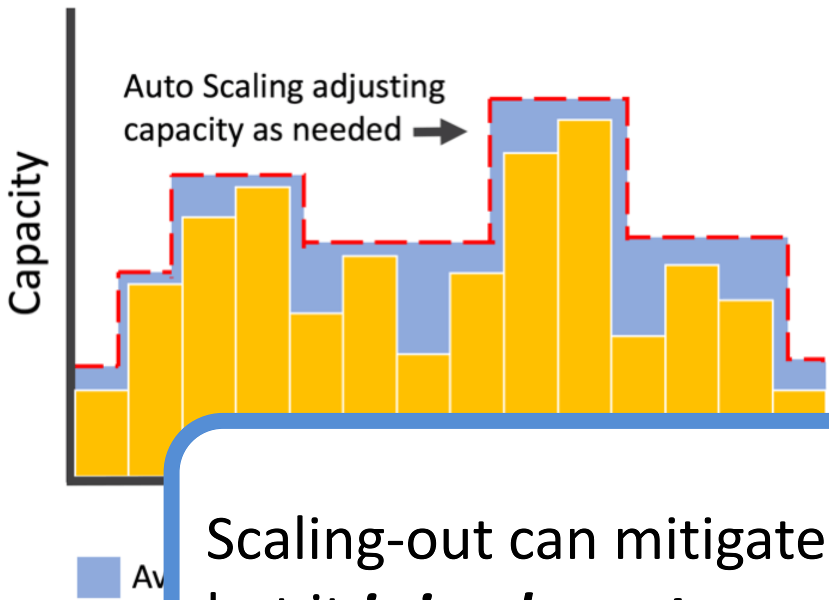
Network overload threats

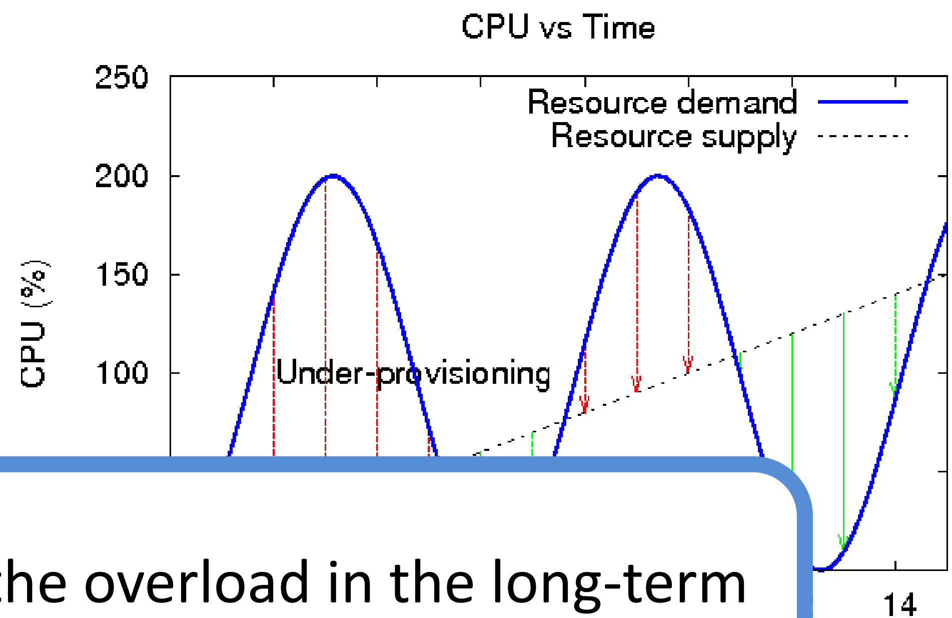
- Overload is **the main cause of cloud service** failures
- The incoming traffic exceeds the capacity of the system hitting some bottleneck
- Faults that impact on the QoS that reduce the capacity of the system:
  - ❖ Software bugs
  - ❖ Virtual machine / processes crashes
  - ❖ Physical resource contention
  - ❖ Poor load balancing
  - ❖ Misconfigurations



Is Cloud “Elasticity" an opportunity ?

- Capacity to adapt to workload changes by **provisioning and de-provisioning resources** in an autonomic manner



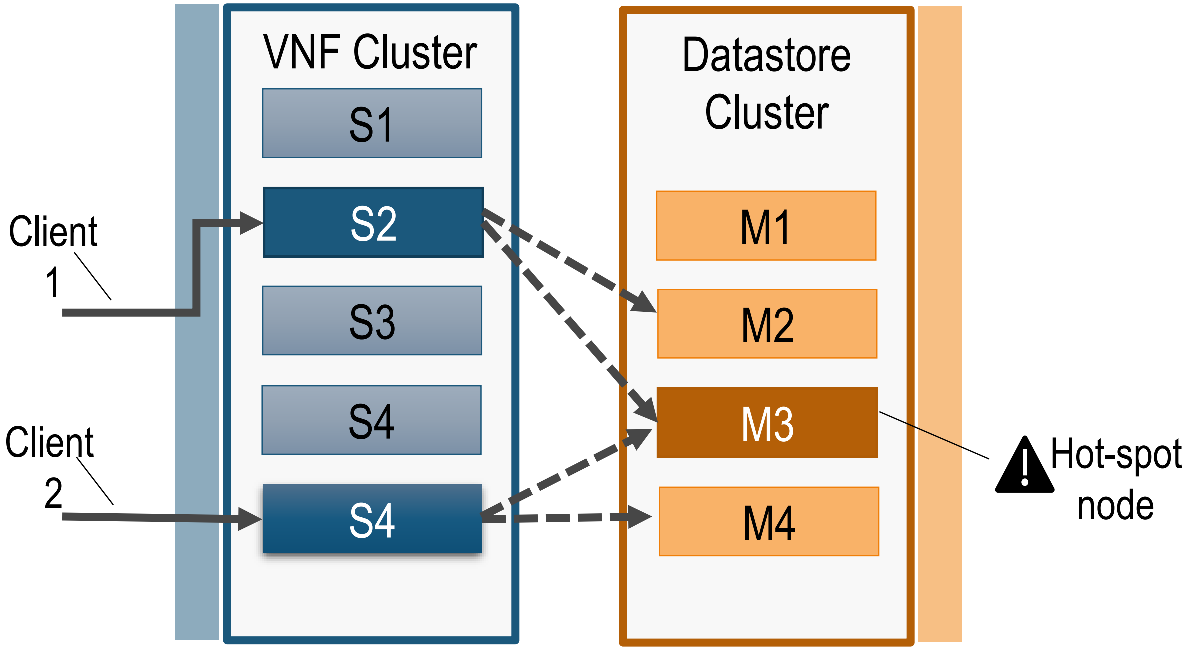


Scaling-out can mitigate the overload in the long-term but it ***is inadequate measure as short-term solution***.

Stateful Network Functions

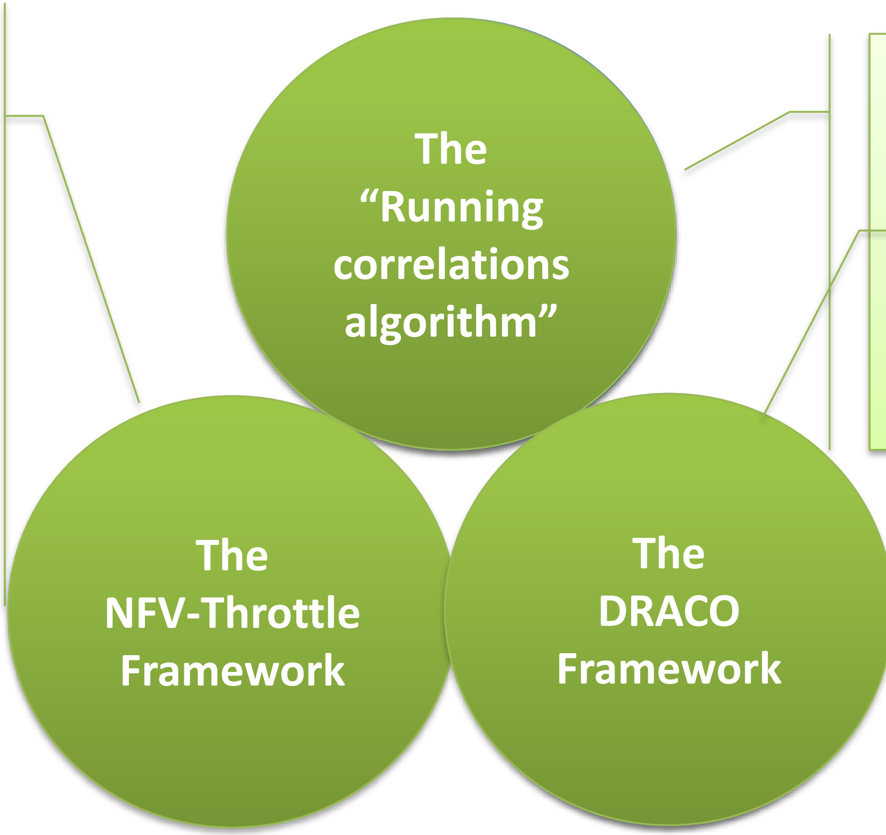
- Network Functions state is separated in a dedicate datastore cluster

- Datastore nodes storing the **most accessed data are more prone to overload**



Overload Management

- A traffic throttling approach based on a control-loop
- Mitigate physical resource contention
- Fits the cloud model services NFVIaaS and VNFaaS



- Detects performance

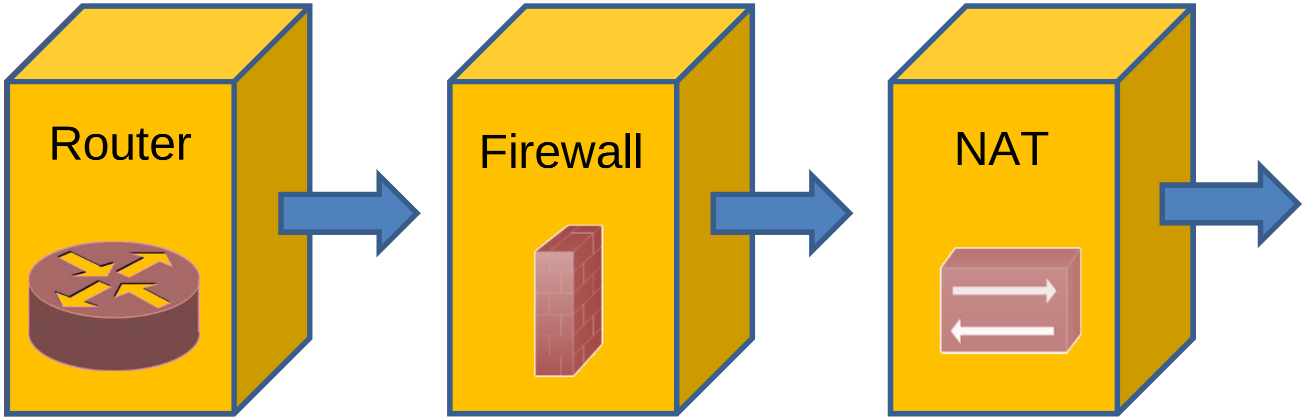
- Extends throttling approach to stateful (multi-tier) VNFs
- Mitigate datastore hot-spots
- Mitigate bottlenecks due to node heterogeneity

D. Cotroneo, R. Natella, **S. Rosiello** – “*A Fault Correlation Approach to Detect Performance Anomalies in Virtual Network Function Chains*”, IEEE 28th International Symposium on Software Reliability Engineering (ISSRE), October 24th, 2017, Toulouse, France

The overload detection approach

characterizing the normal behaviour

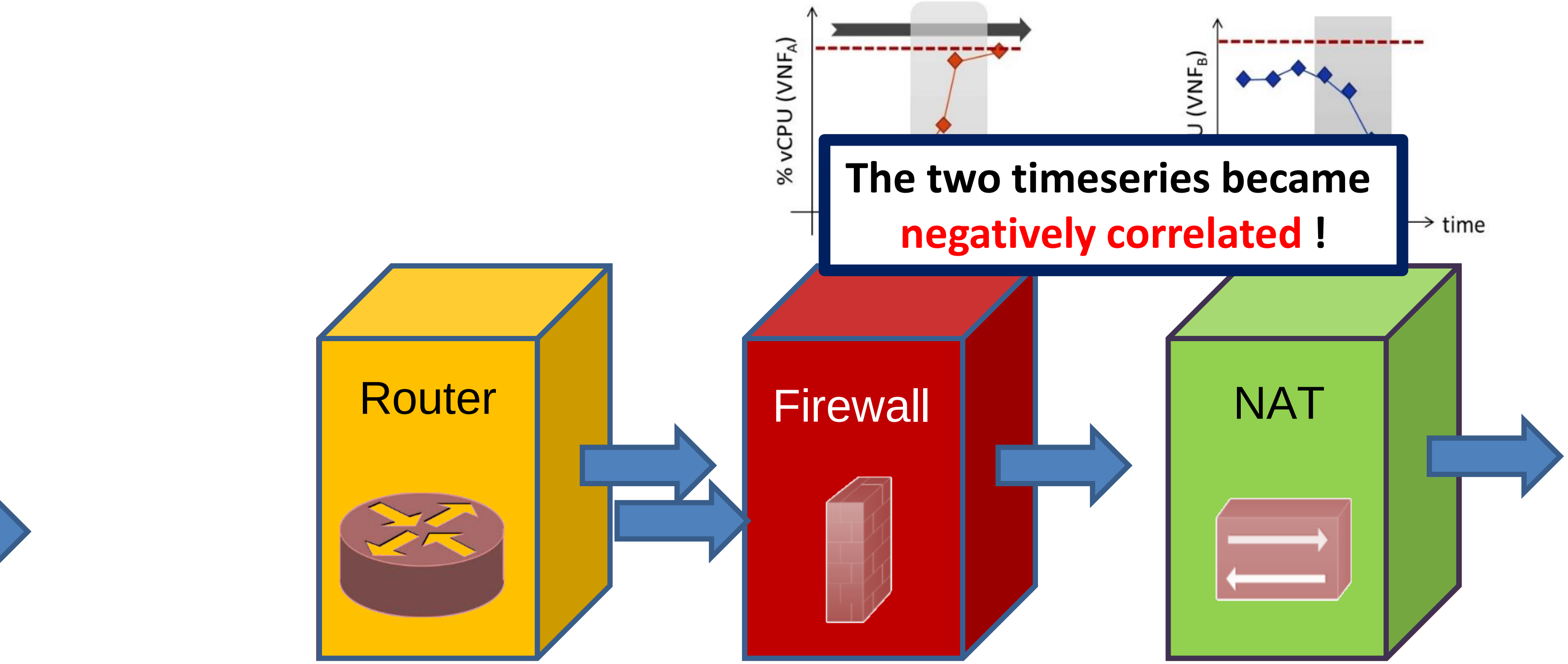
- A VNF chain **can be seen as a multistage pipeline,**
  - The output of a network function is the input for the next one.



> ➢ The load metrics of each stage are **strongly correlated** !

The driving idea

detecting an anomaly in the chain



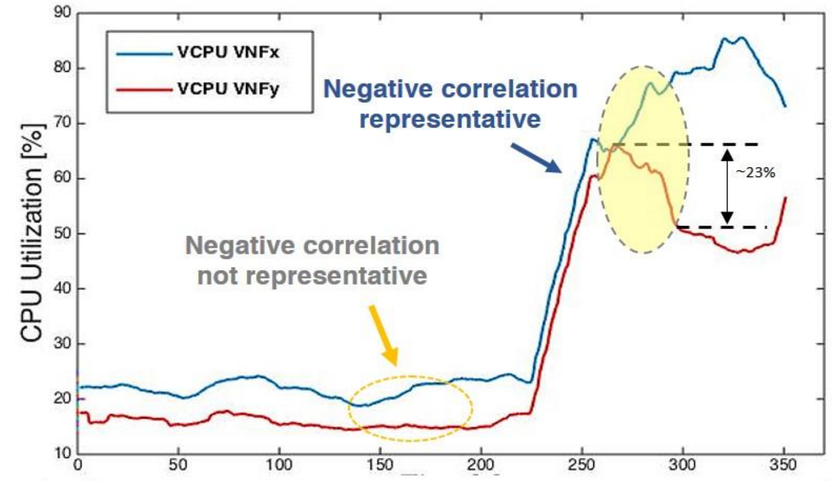
Bottleneck overloaded

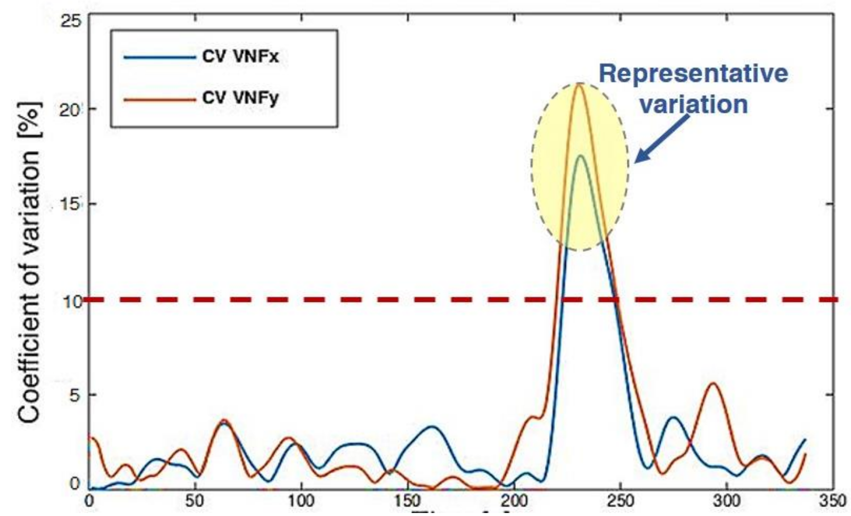
Lower traffic

The approach take into account the topology and consider this negative correlation a symptom of a performance anomaly

Avoid false alarms

- By chance, one of the time series may slightly increase due to **random fluctuations**, and at the same time the other time series may decrease.
- **compute the coefficient of variation** on a window of samples
  - A **correlation is taken into account if the cv is non negligible** (above 10%)





Workload surges detection outcomes

Overload Subs. (MTN)	Window	Smooth	Detection Outcome	Detection Latency (seconds)
20%	10	RMM	Detected (4/5)	29.0
	20	RMM	Detected (4/5)	45.6
	30	RMM	Not Det. (1/5)	28.0
	10	RMA	Unrel. Det. (2/5)	37.0

1000%	20	RMA	Detected (4/5)	57.0
	30	RMA	Non Det. (0/5)	-
	10	EMA	Detected (5/5)	36.0
	20	EMA	Non Det. (2/5)	57.0
	30	EMA	Non Det. (0/5)	-

- We use a sampling period of 2s
- We study the

detection outcome and on latency

➢ By using a **10 samples window** and applying a **moving median** we have:

  ➢ **A detection coverage of all the considered scenarios.**

  ➢ **An average detection latency less than 32s**

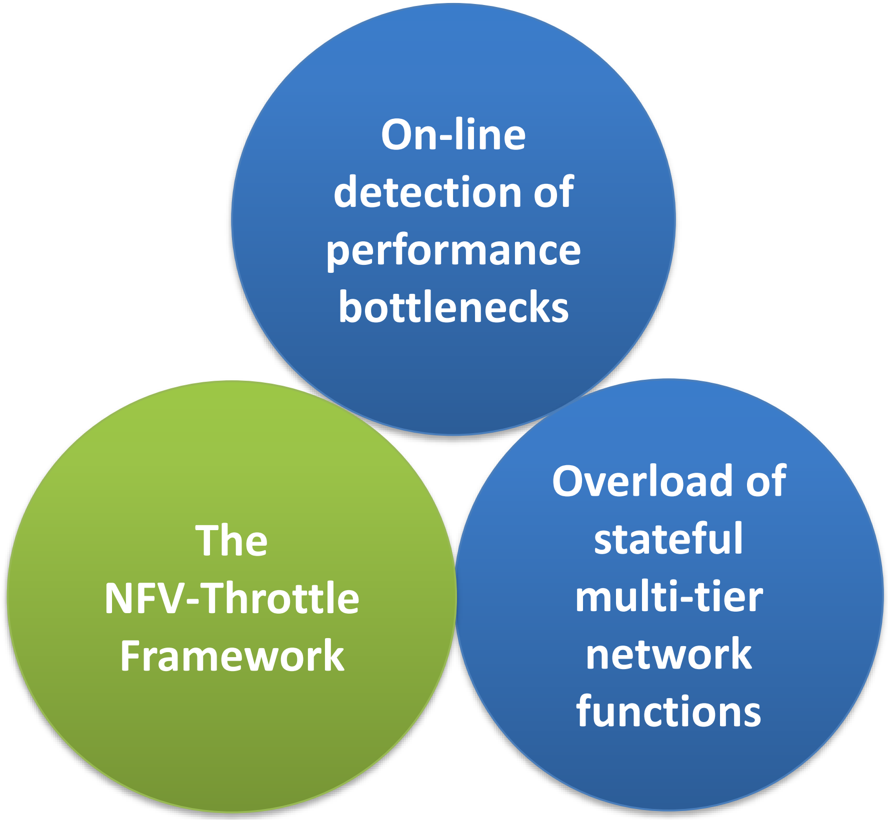
Performance-fault injection results

Failures	Window	Smooth	Detection Outcome	Detection Latency (seconds)
physical CPU contention	10	RMM	Detected (4/5)	13.0
S-CSCF crash	10	RMM	Detected (5/5)	24.0
P-CSCF failover	10	RMM	Detected (5/5)	18.0

Full detection coverage

Average Detection latency < 30 s

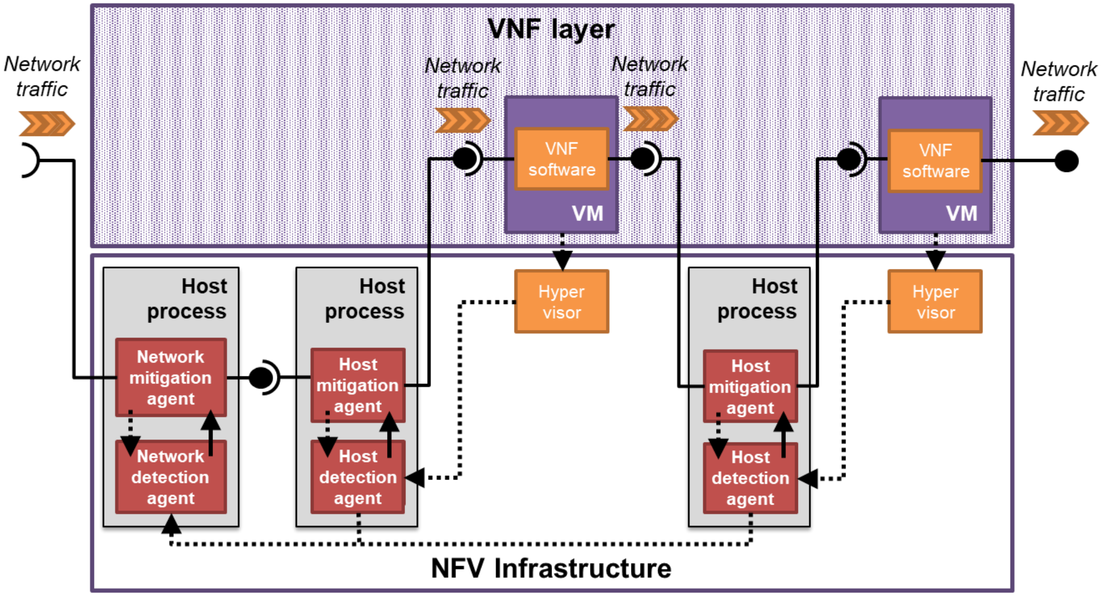
Overload Management



D. Cotroneo, R. Natella, **S. Rosiello** – “*NFV Throttle: An Overload Control Framework for Network Function Virtualization*” – IEEE Transaction on Network and Service Management

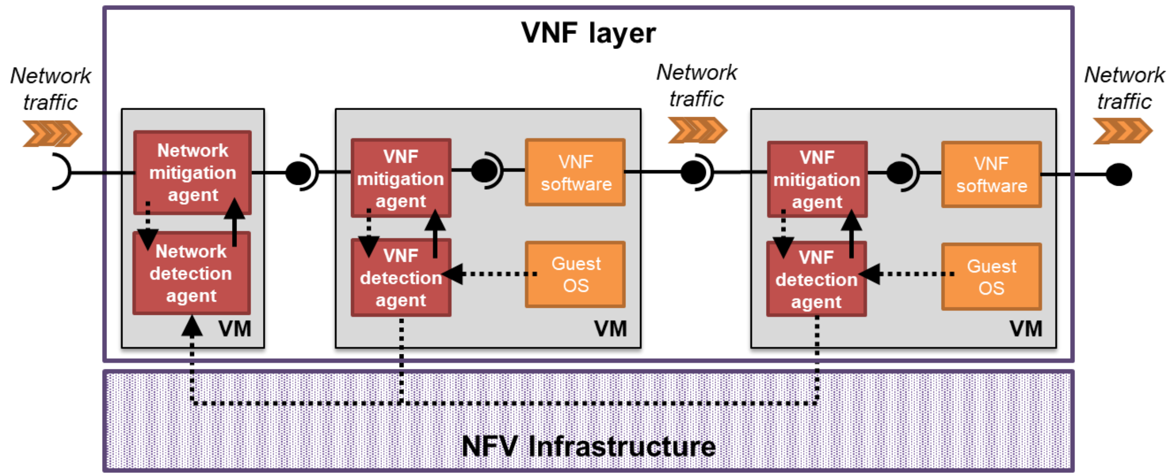
D. Cotroneo, R. Natella, **S. Rosiello** – “*Overload Control for Virtual Network Functions under CPU Contention*” – Future Generation Computer Systems, Elsevier (under review)

The framework components

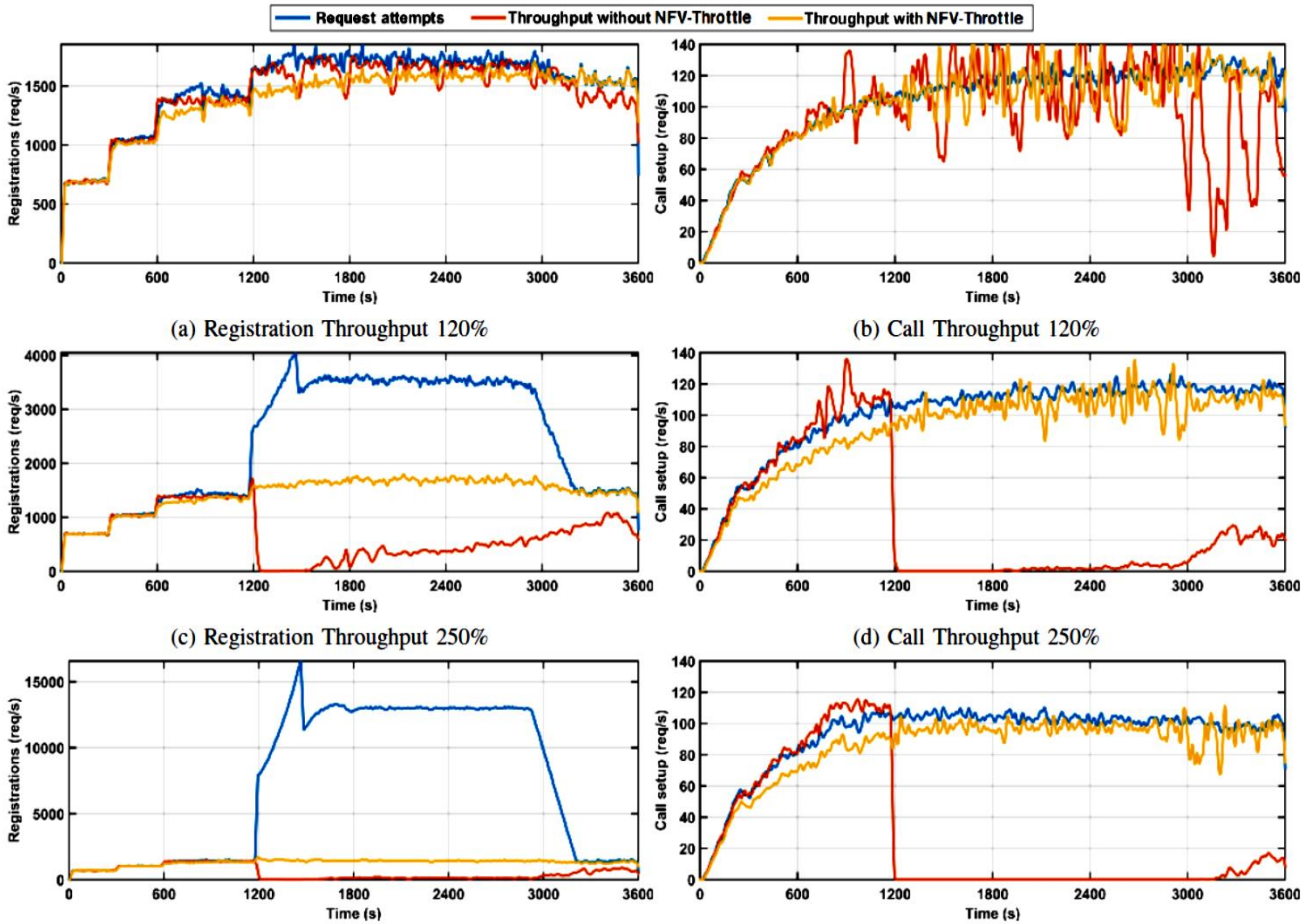


- **Framework components**
  - **VNF Detection agents:** estimate the actual capacity of a network function
  - **VNF Mitigation agents**: Inspect and throttle network traffic exceeding the capacity

- **Framework features**
  - **2 service models**: NFVIaaS and VNFaaS
  - **3 levels of protection**: guest, host and network



Experimental results on a vIMS



(a) Registration Throughput 120%

(b) Call Throughput 120%

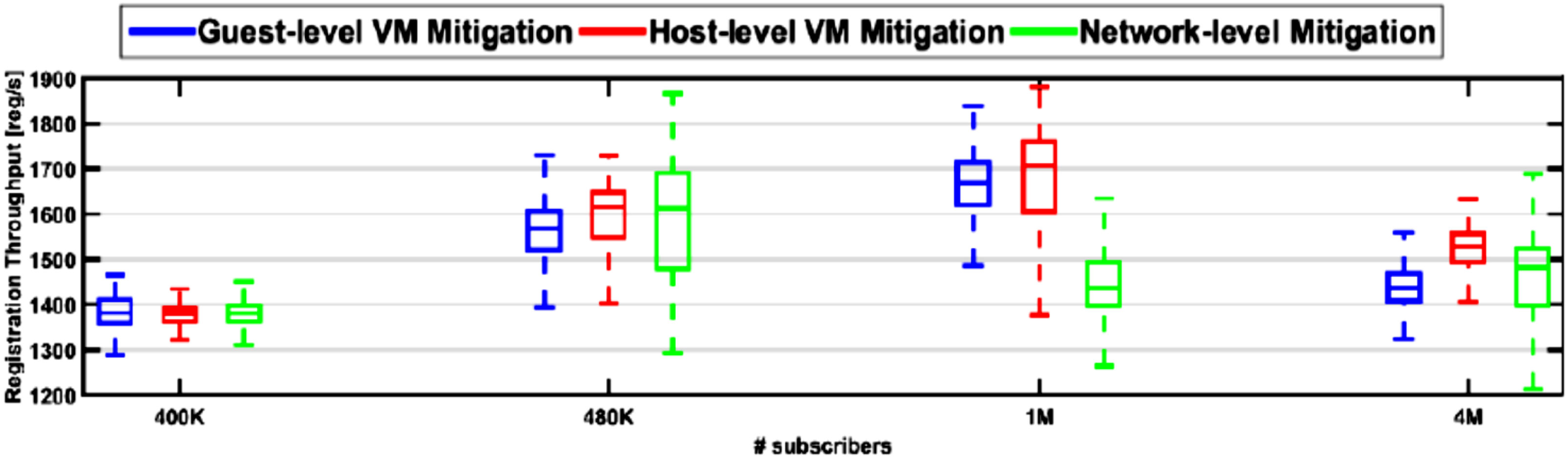
(c) Registration Throughput 250%

(d) Call Throughput 250%

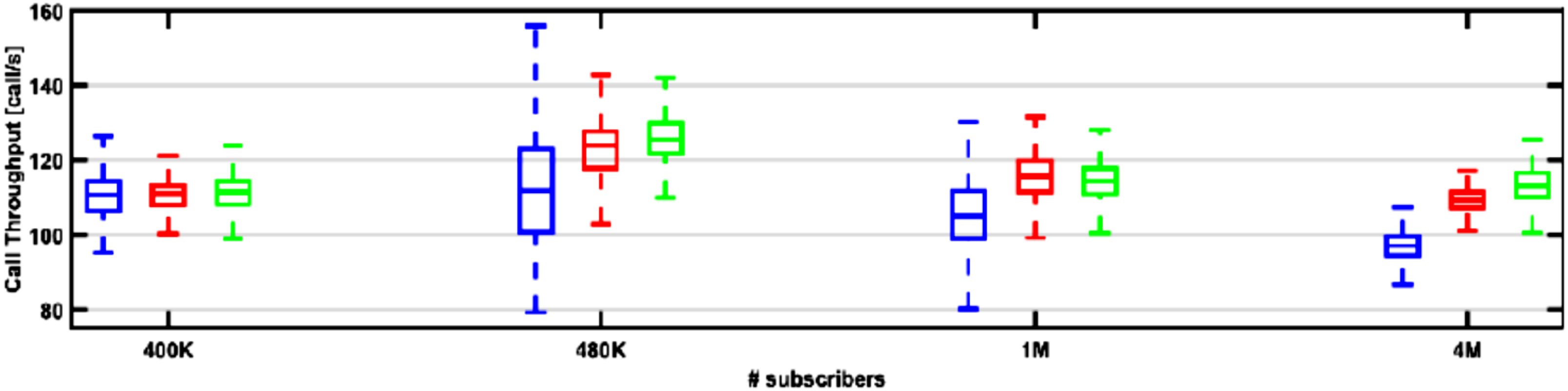
(e) Registration Throughput 1000%

(f) Call Throughput 1000%

Levels of protection

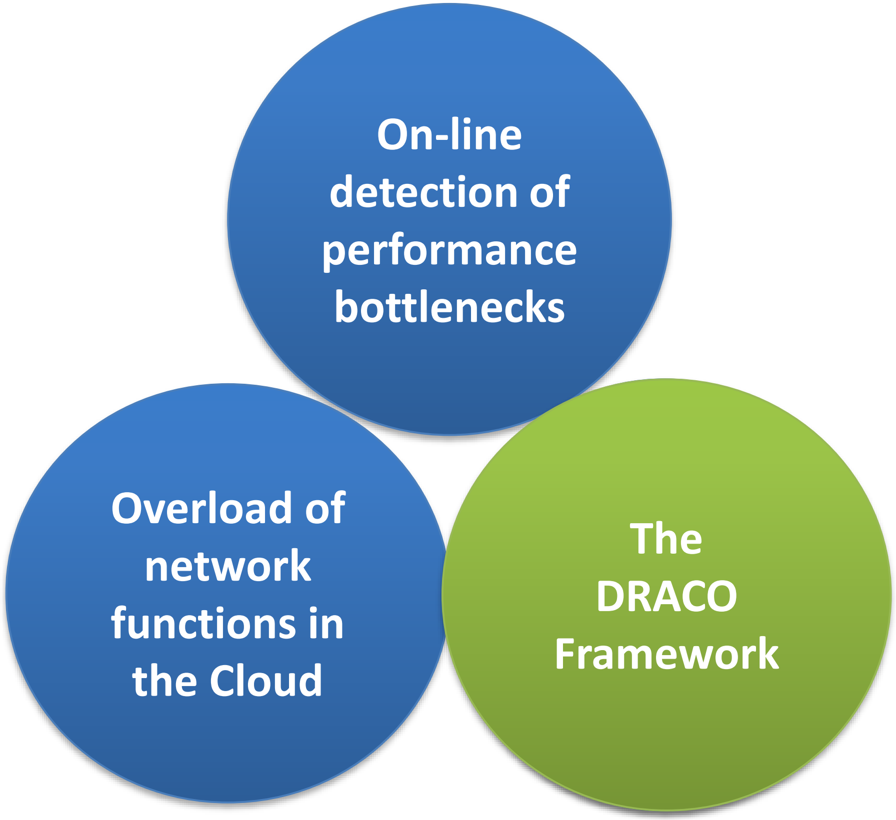


(a) Registration Throughput



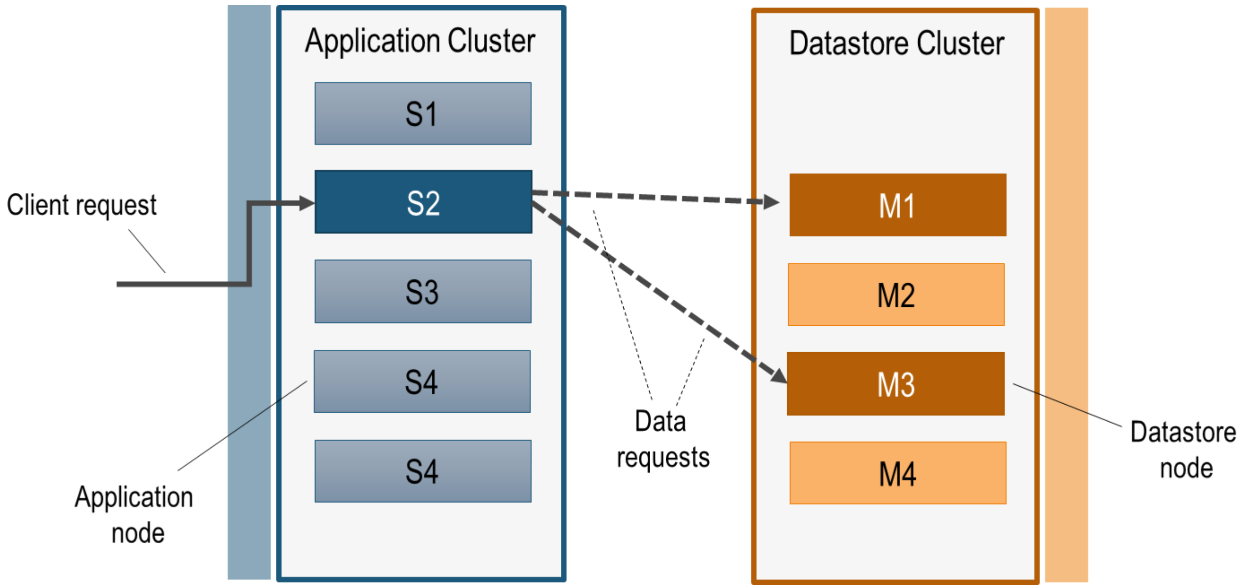
(b) Call Throughput

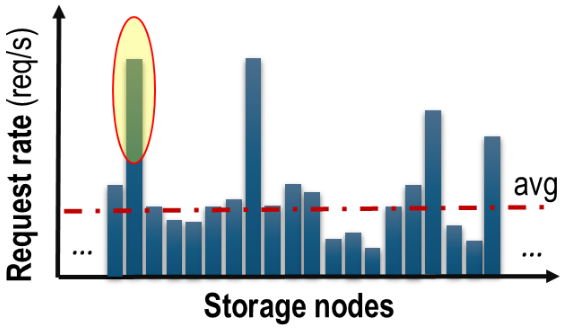
Overload Management

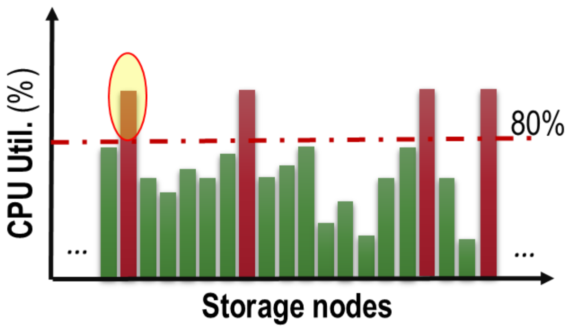


D. Cotroneo, R. Natella, **S. Rosiello** – "DRACO: Distributed Resource-aware Admission Control for Large Scale, Multi-tier systems" – ACM Transactions on Computer Systems, ACM (**under review**)

The unbalanced load problem



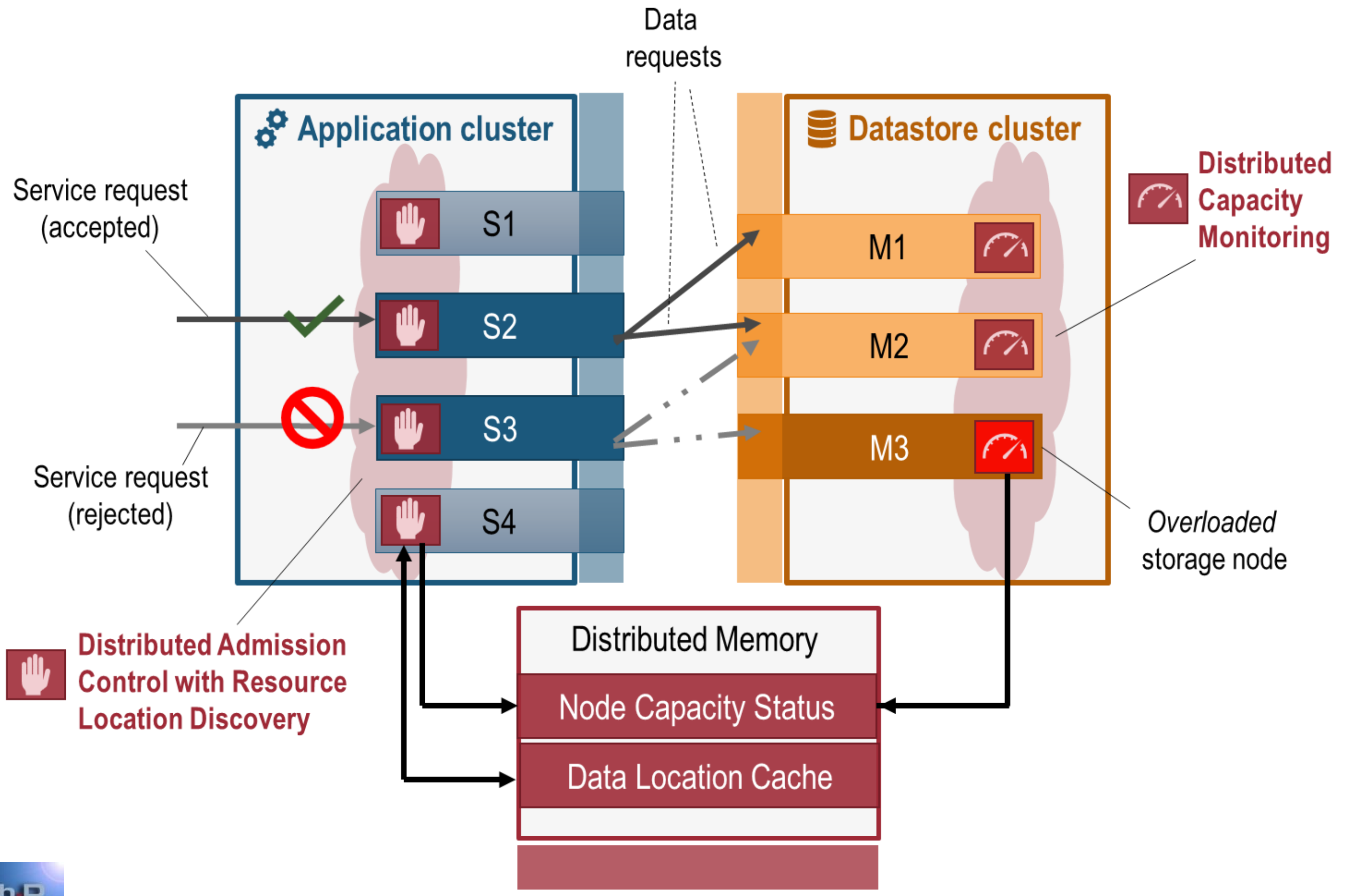




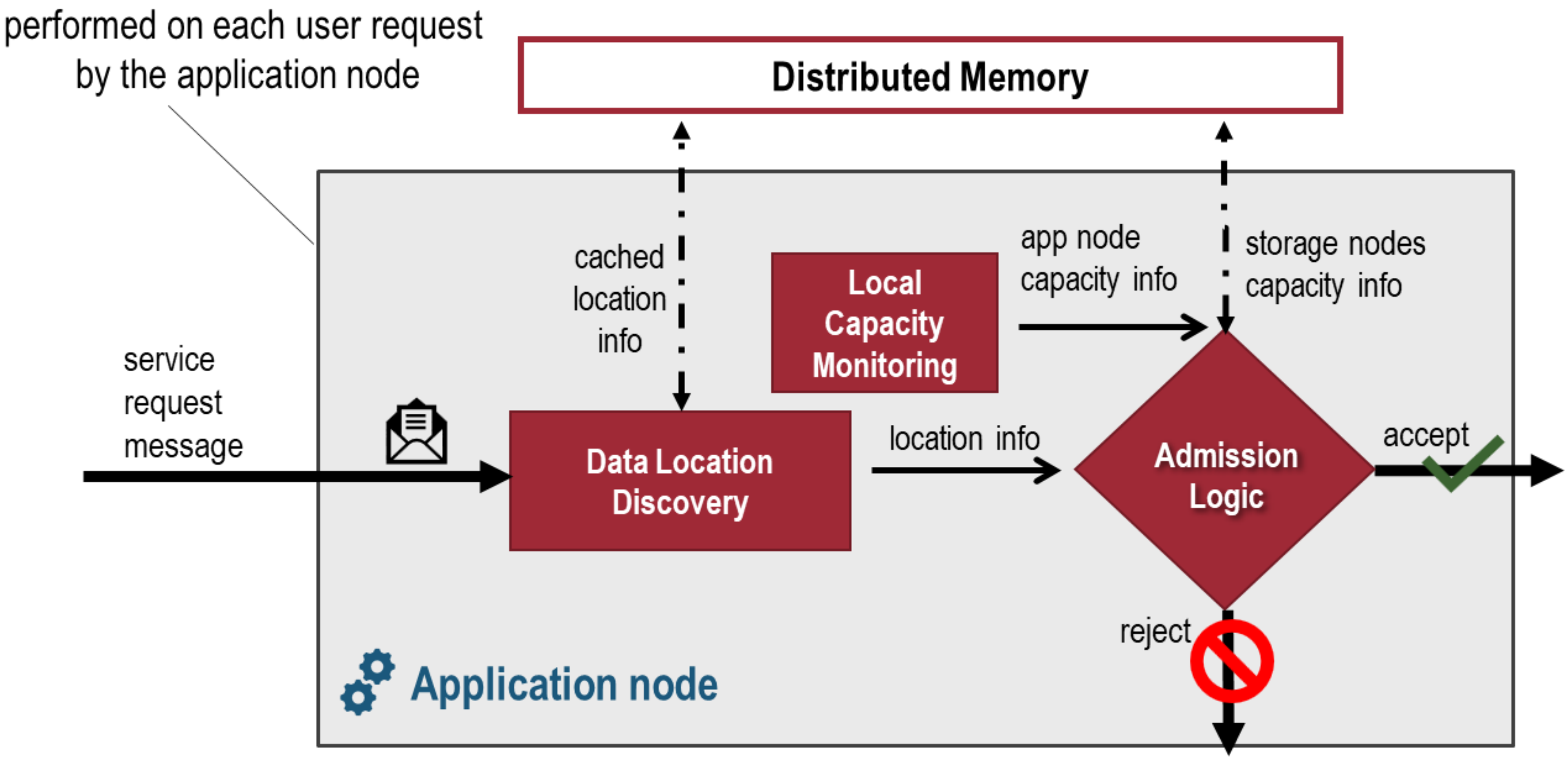
- Stateless application tiers access distributed state in a **separate key-value datastore**
- Hash-based node access
  – Limited load balancing
- **Throttling datastore requests is not possible to avoid consistency issues.**

DRACO overview

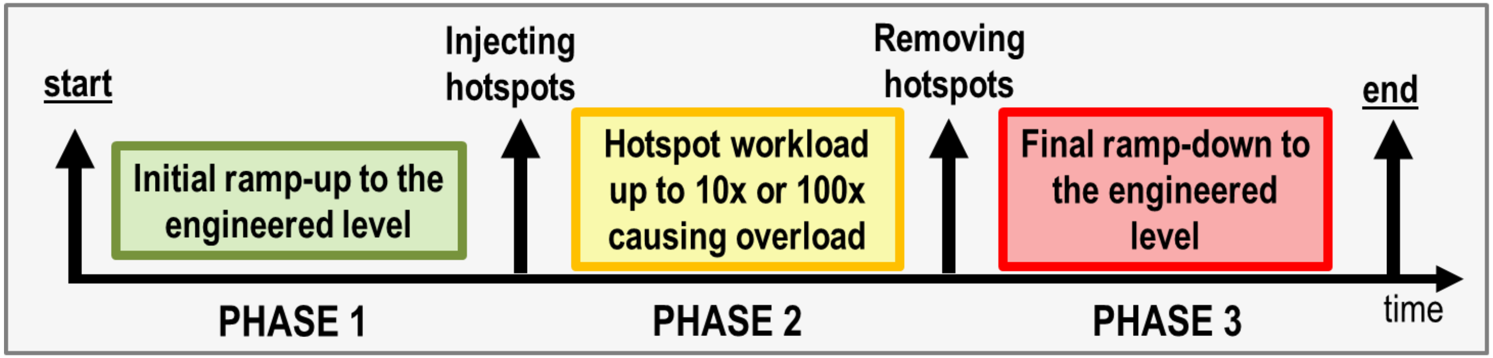
- **Distributed approach**: separate admission control and capacity monitoring agents.

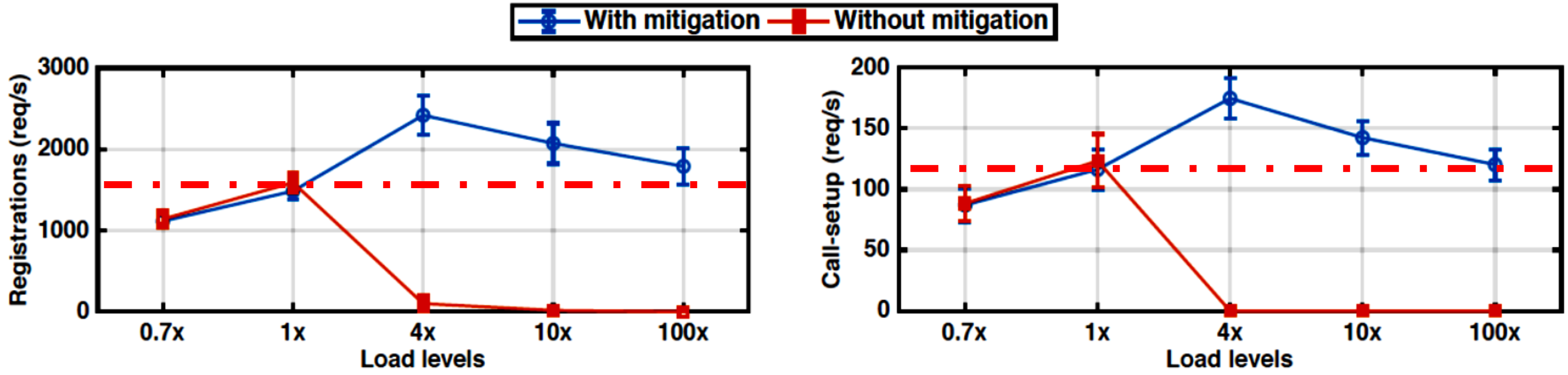


Fine-graded admission control



Experimental results on the vIMS





Conclusions

How can a service provider detect and mitigate overload conditions in the short-term ?

The “Running correlations algorithm”

The NFV-Throttle Framework

The DRACO Framework

Products

International Journal

D. Cotroneo, R. Natella, **S. Rosiello** – "*NFV Throttle: An Overload Control Framework for Network Function Virtualization*" – IEEE Transaction on Network and Service Management, September 2017, ISSN: 1932-4537, IEEE Computer Society Press

D. Cotroneo, R. Natella, **S. Rosiello** – "*Overload Control for Virtual Network Functions under CPU Contention*" – Future Generation Computer Systems, Elsevier (**under-review**)

D. Cotroneo, R. Natella, **S. Rosiello** – "DRACO: Distributed Resource-aware Admission Control for Large Scale, Multi-tier systems" – ACM Transactions on Computer Systems, ACM (**under-review**)

D. Cotroneo, A.K. Iannillo, R.Natella, **S. Rosiello**, "*Software Fault Injection for the Android Mobile OS*", IEEE Computers Magazine (**under-review**)

International Conferences

D. Cotroneo, R. Natella, **S. Rosiello** – "*A Fault Correlation Approach to Detect Performance Anomalies in Virtual Network Function Chains*", IEEE 28th International Symposium on Software Reliability Engineering, Tolouse, France

Credits summary

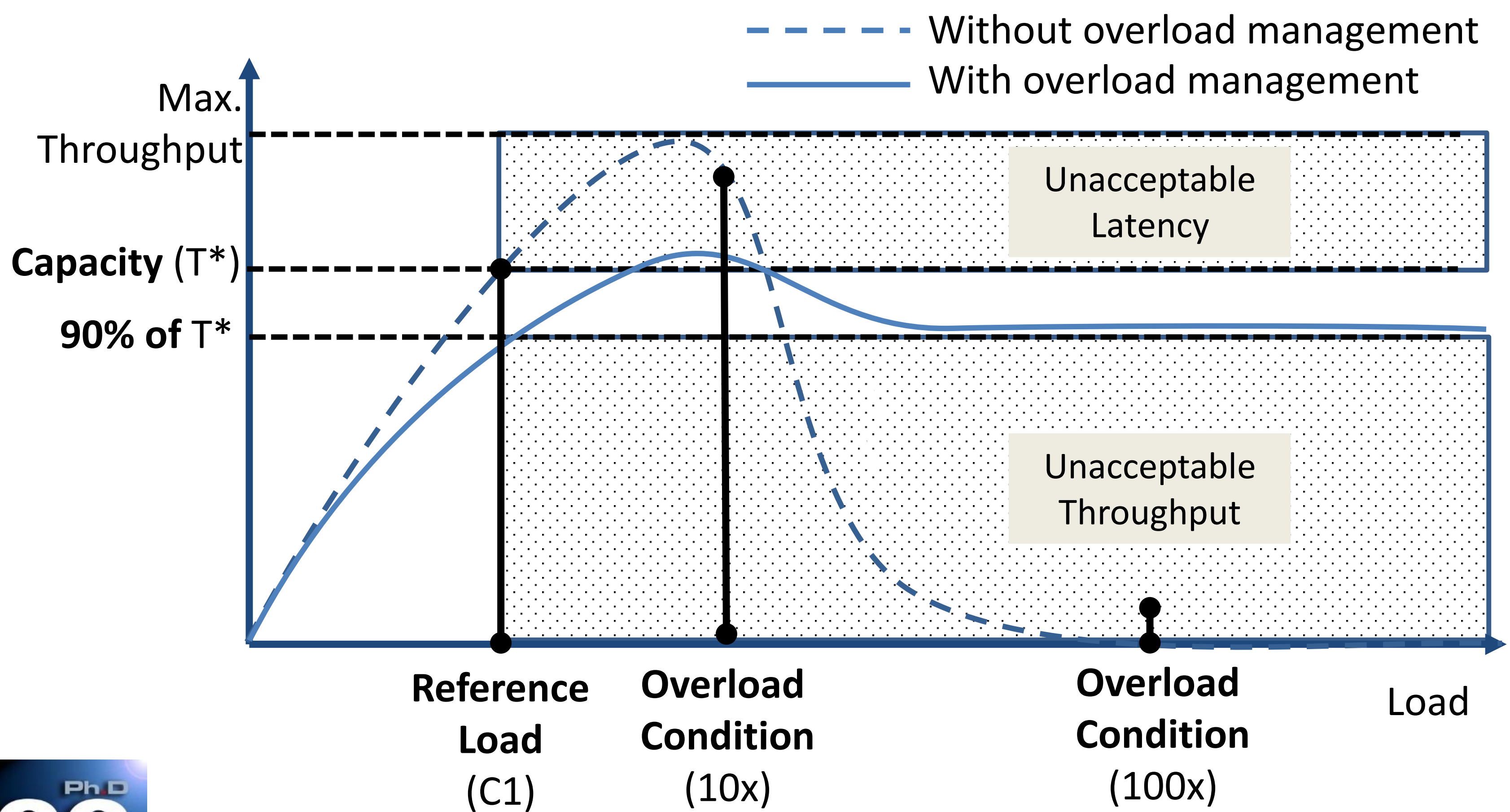
	Credits year 1								Credits year 2								Credits year 3									
		1	2	3	4	5	6			1	2	3	4	5	6			1	2	3	4	5	6			
	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	**Summary**	**Estimated**	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	**Summary**	**Estimated**	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	**Summary**	**Total**	**Check**
Modules	**20**	0	7	0	3	0	9	**19**	**10**	0	9	2	0	0	0	**11**	**0**	0	0	0	0	0	0	**0**	**30**	**30-70**
Seminars	**5**	0	0.8	0.8	1.2	0	0.5	**3.3**	**5**	1.8	0	2.6	0	0	1.4	**5.8**	**5**	0.4	0	1.2	0.4	0	2.4	**4.4**	**14**	**10-30**
Research	**35**	10	2	9	6	10	1	**38**	**40**	8	1	5	10	10	9	**43**	**55**	9.6	10	8.8	9.6	10	7.6	**56**	**137**	**80-140**
	60	10	9.8	9.8	10	10	11	**60**	**55**	9.8	10	9.6	10	10	10	**60**	**60**	10	10	10	10	10	10	**60**	**180**	**180**

Thank you !

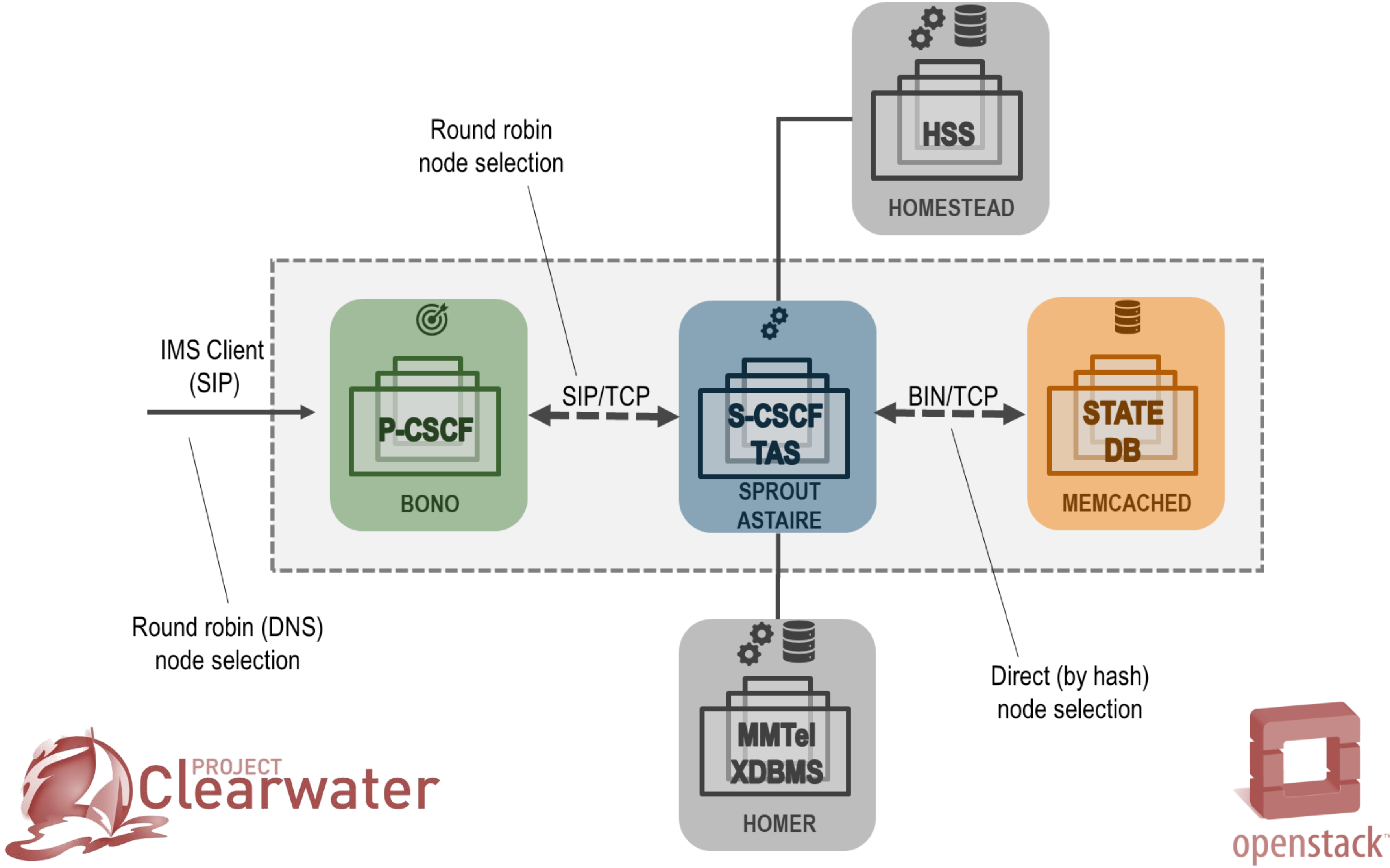
BACKUP SLIDES

The overload problem

- Is the main cause of cloud service failures: **external load exceeds the capacity of the system hitting some bottleneck**



The Case-Study: Clearwater IMS



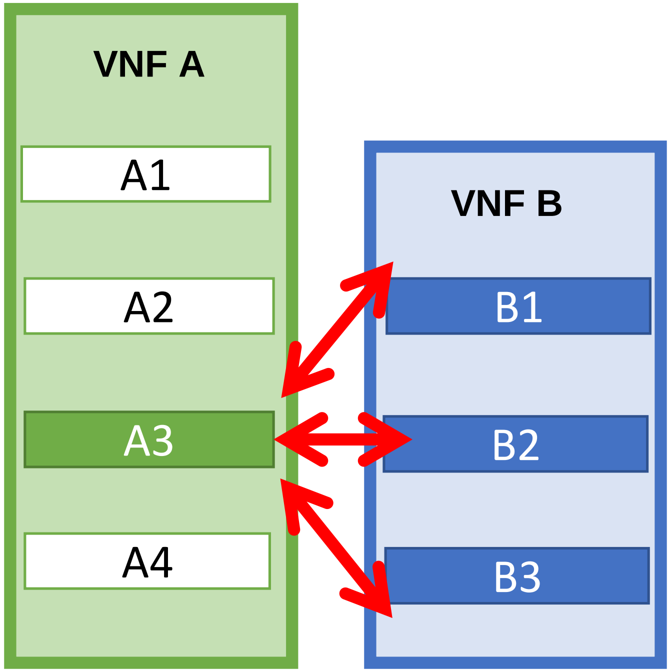
Running correlations: Approach

- In real deployments, the load is balanced across a big number of **VNF active replicas**.

➢ We **compute Pearson's correlation indexes** (to find symptoms of anomalies) between each couple of replicas in two **adjacent tiers**.

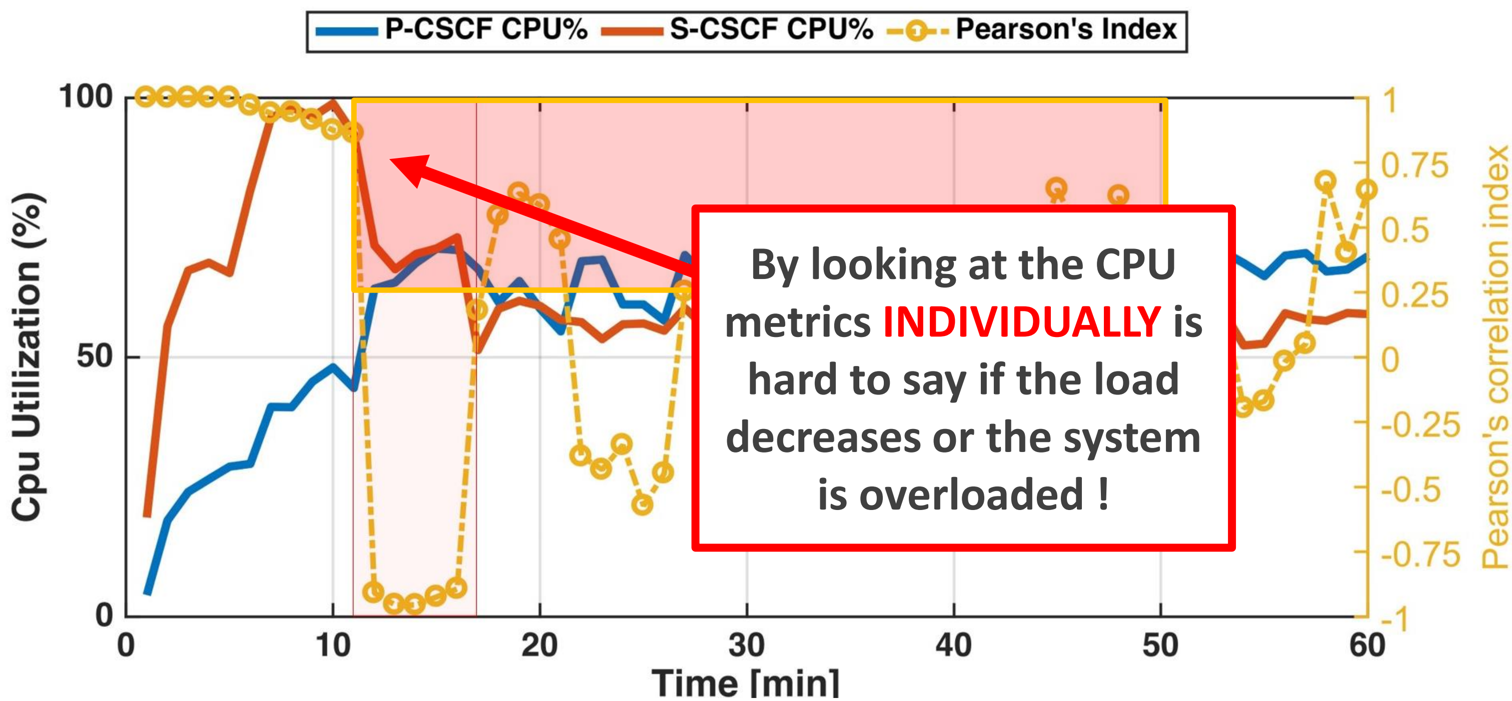
➢ We raise an alert in a tier only if the anomaly is **detected by a majority of nodes**.

  ❖ This prevents false alarms due to random load fluctuation due to poor load balancing.



$$\rho = \frac{COV(W_A, W_b)}{\sigma_A \ \sigma_B}$$

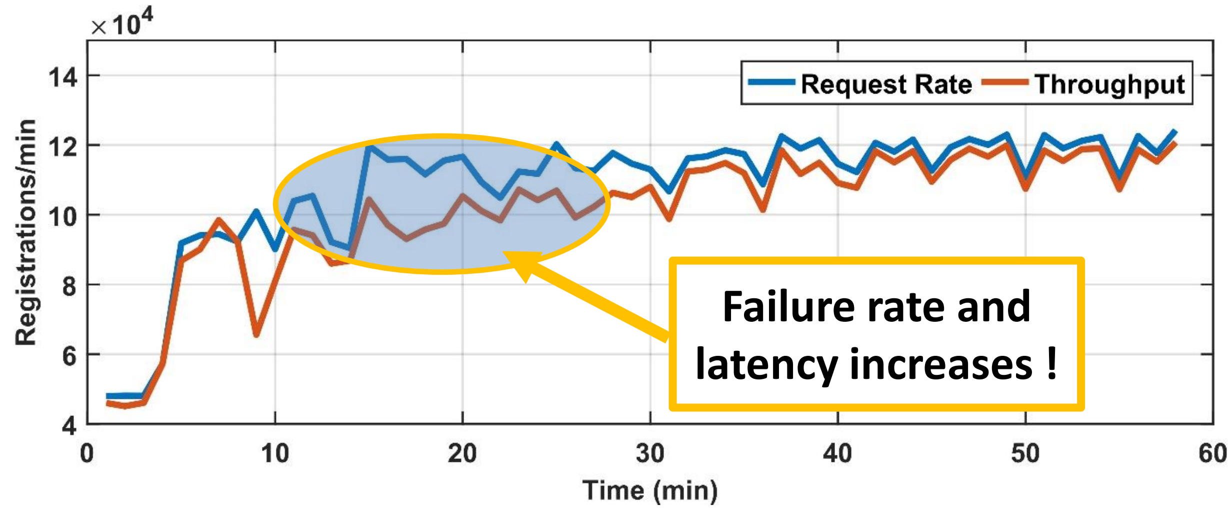
Analyzing the correlation between CPU time-series in Clearwater



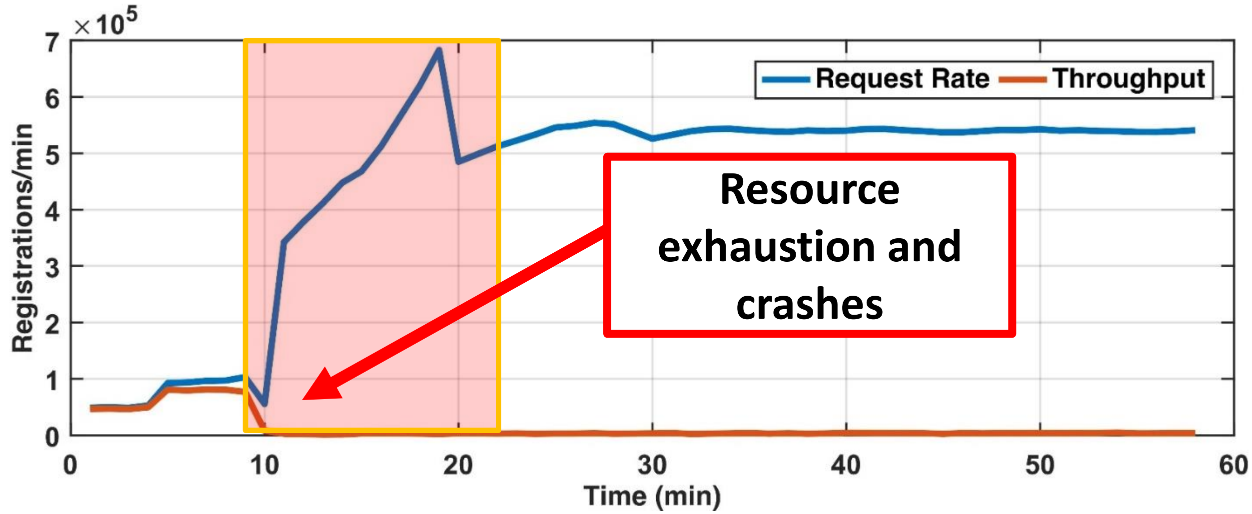
Workload surges

hitting the system bottleneck

- ❑ 20 % More than the engineered capacity

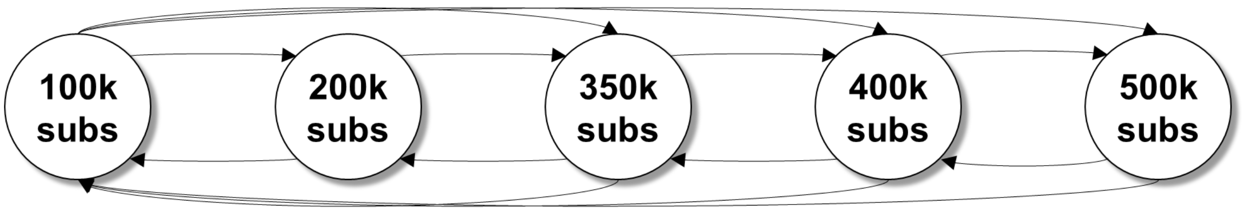


- ❑ 1000 % more than the engineered capacity



Anomaly free, long run experiments

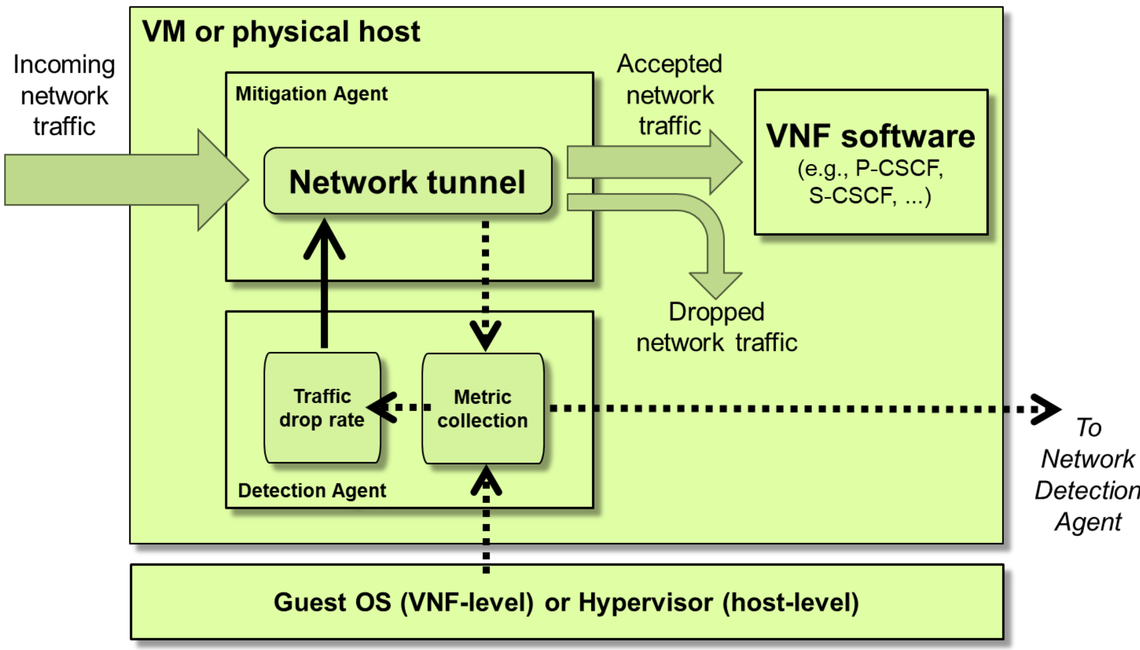
- We performed **anomaly free experiments** to assess false alarms raised by the algorithm
  - Constant workload below the engineered capacity
  - Varying the number of the subscribers each 20 min between 100k and 500k



➢ **No false alarms** were raised during these experiments !

  ➢ require **strong correlations**
  ➢ require **feedbacks from a majority of nodes**
  ➢ **use the COV** to discard non representative correlations
  ➢ apply a **smoothing function to discard outliers** in each sampling window

VNF-level / Host-Level protection
Detect locally Act locally

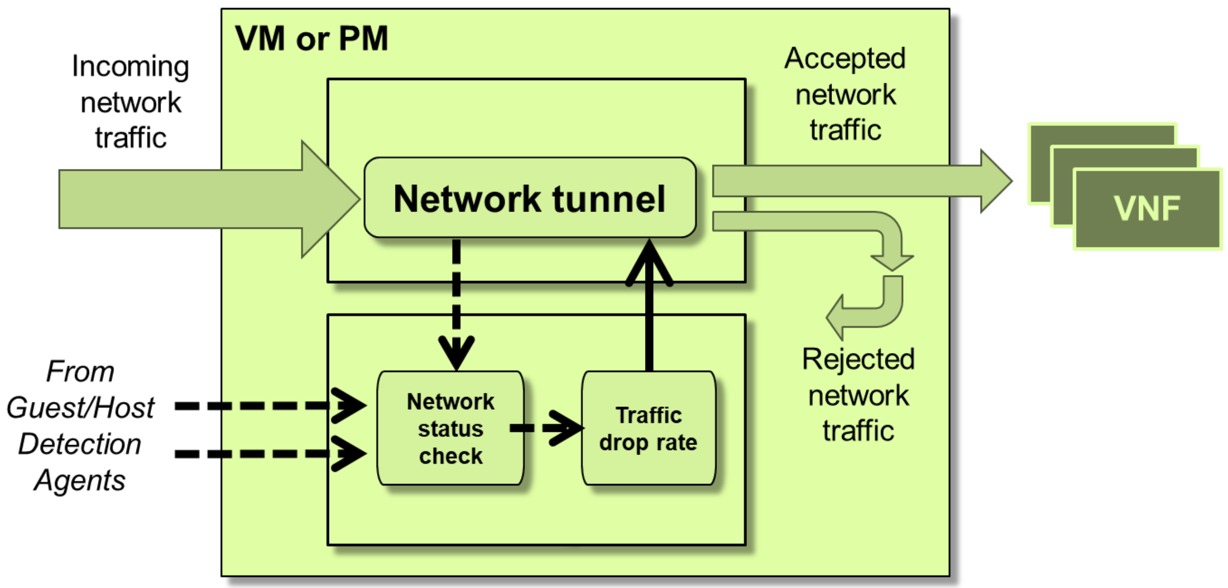


$$\text{capacity} = \frac{\text{MEAN}[\text{accepted\_traffic}[1 \ldots N]]}{\frac{\text{MAX}[\text{cpu\_usage}[1 \ldots N]]}{\text{reference\_cpu\_usage}}}$$

$$\text{drop\_rate} = 100 \cdot \left(1 - \frac{\text{capacity}}{\text{incoming\_traffic}[N]}\right)$$

- Estimate the current capacity based on
  - The incoming network traffic
  - The VNF resource consumption
- The network traffic is **intercepted by the mitigation module** which performs admission control

Network-level protection (Detect globally Act globally)

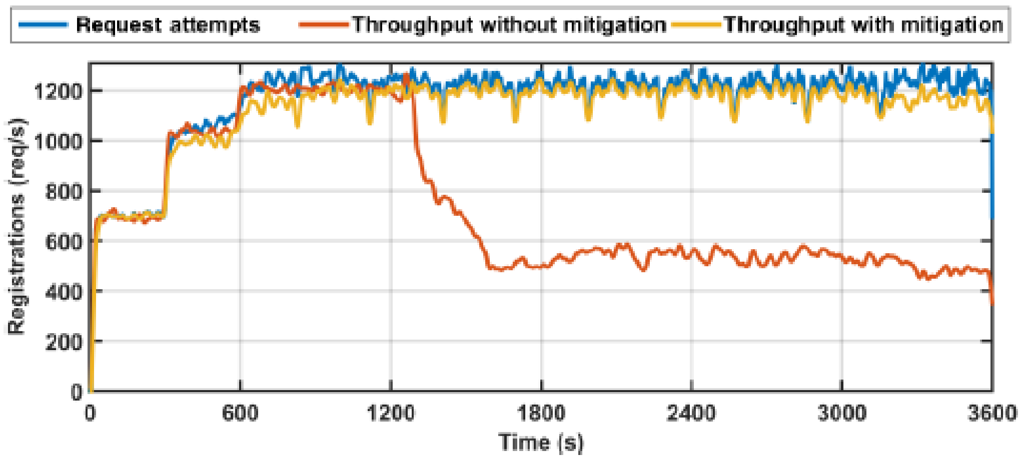


$$\text{capacity} = \begin{cases} \text{capacity}/(\alpha+\gamma) & \text{if overloaded} \\ \text{capacity}\cdot(\beta-\gamma) & \text{otherwise} \end{cases}$$

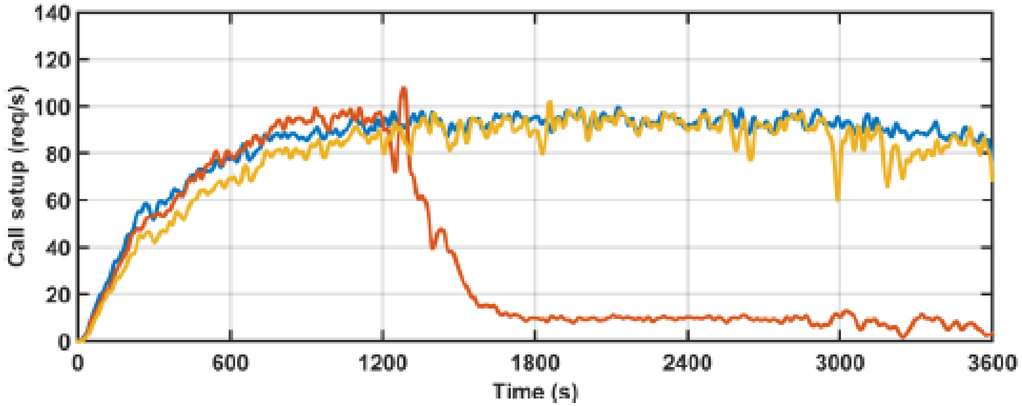
$$\text{reject\_rate} = 100 \cdot \left(1 - \frac{\text{capacity}}{\text{incoming\_traffic}[N]}\right) [\%]$$

- Additional level of protection
- **Global view** of all the node capacity
- Admission control performed at **network edges**
- Can **notify user-agents** to reduce the load due to the overload

Host level CPU contention

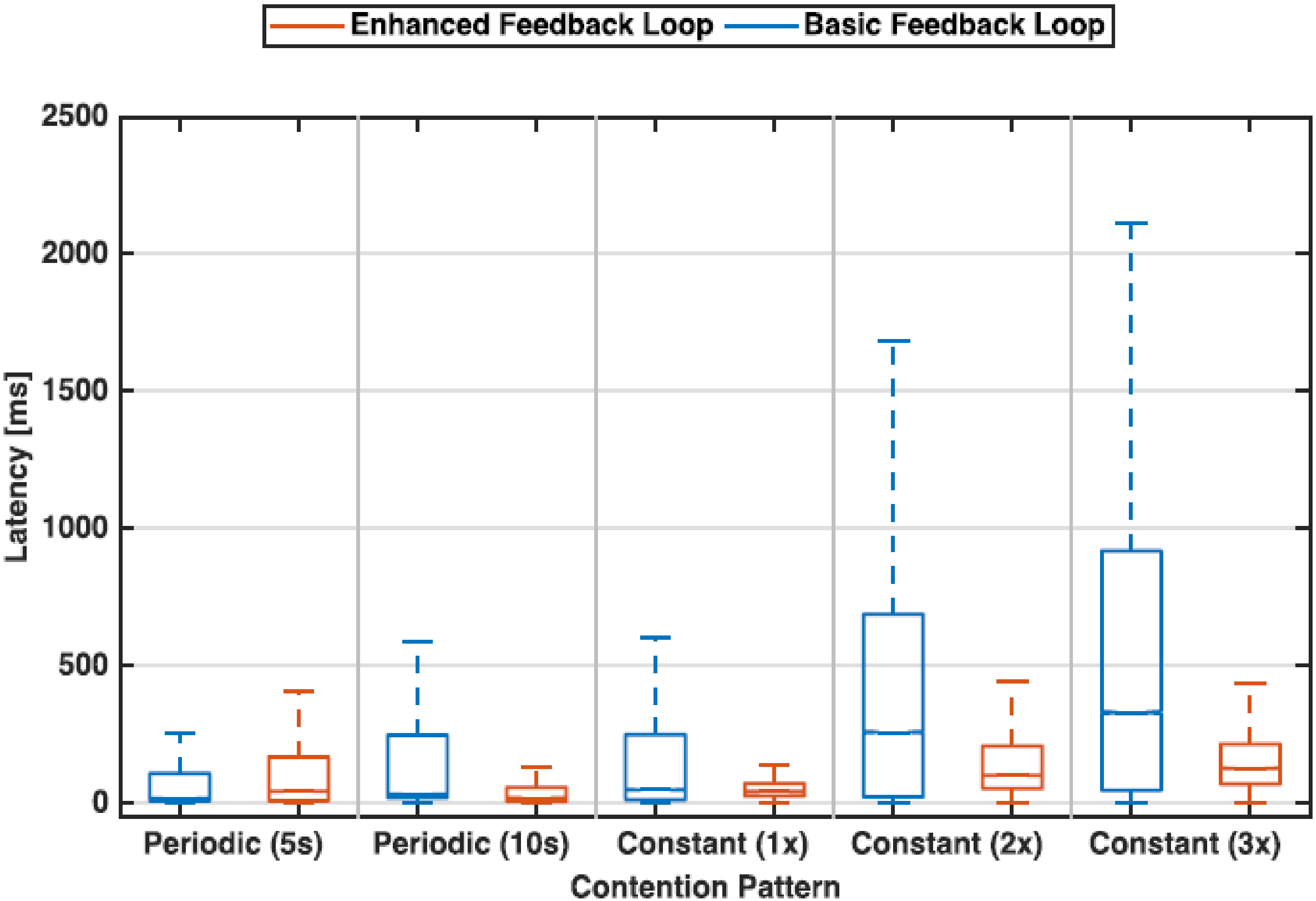


(a) Registration Throughput

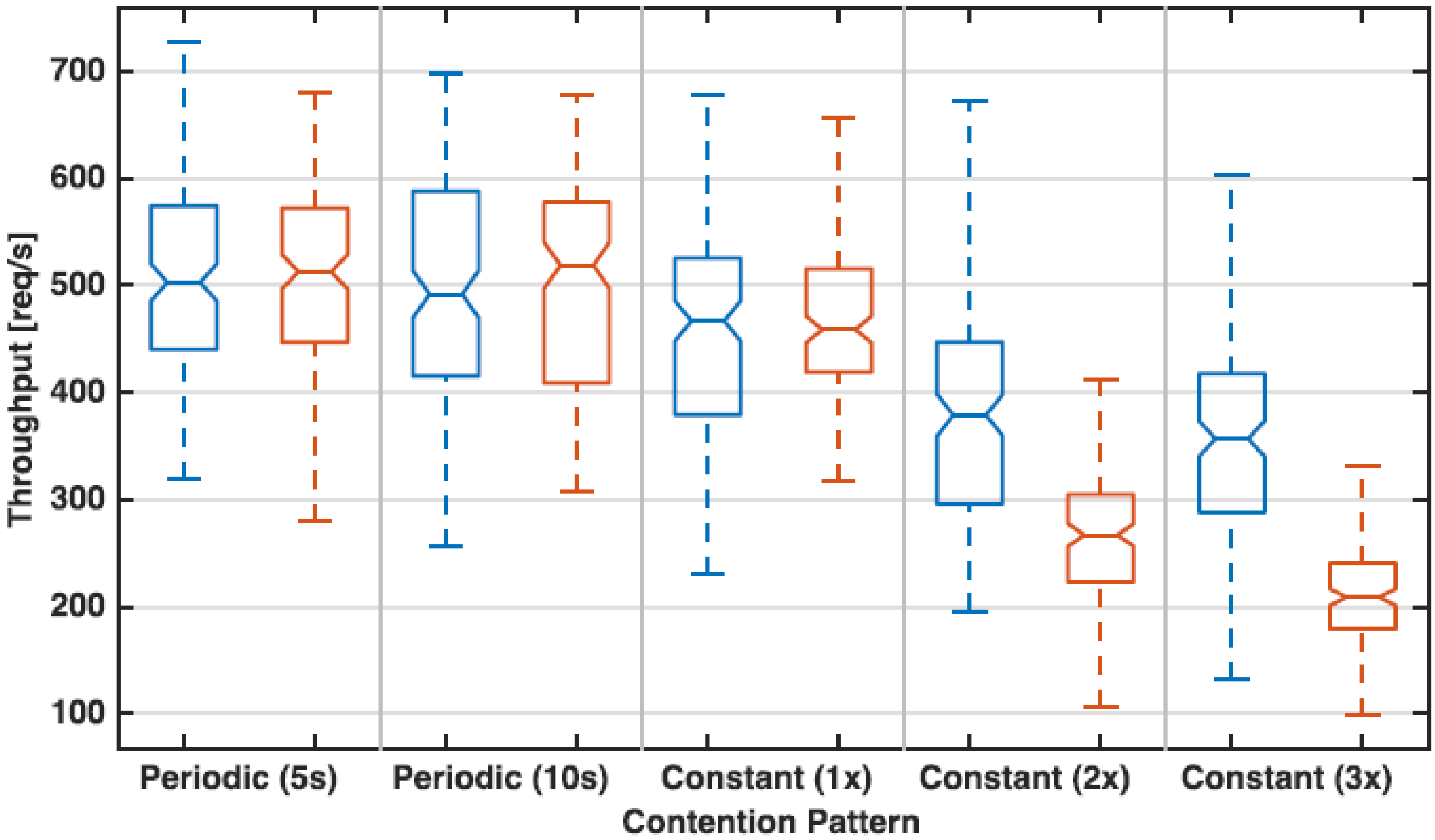


(b) Call Setup Throughput

Latency during contention



Throughput during contention



(b) IMS Throughput

Stateful Network Functions

- Even when the load on the datastore is balanced overload can happen due to server heterogeneity

- **Nodes with lower capacity acts as a bottleneck for the whole tier**