5G Network service resilience estimation under short time scale traffic variation

Chair Risk and Resilience of Complex Systems

Annual Scientific Seminar

Presenter:

Rui Li (starting date: 02/11/2020)

Supervisors:

Bertrand Decocq (Orange Innovation)

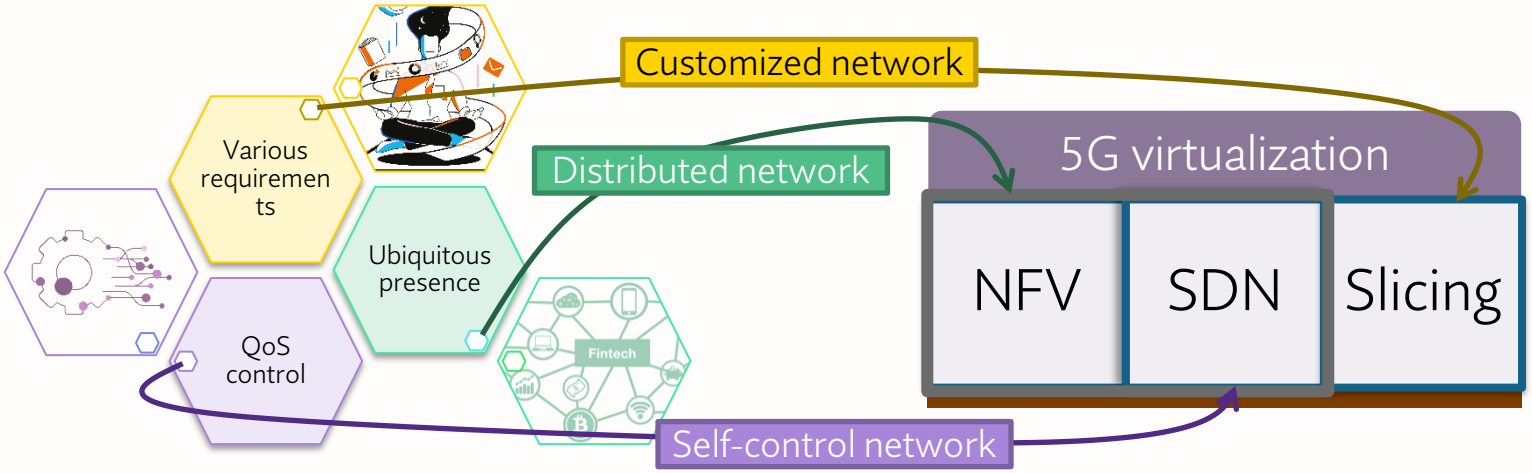
Anne Barros, Yiping Fang, Zhiguo Zeng (CentraleSupélec)

17 November 2022

Table of contents

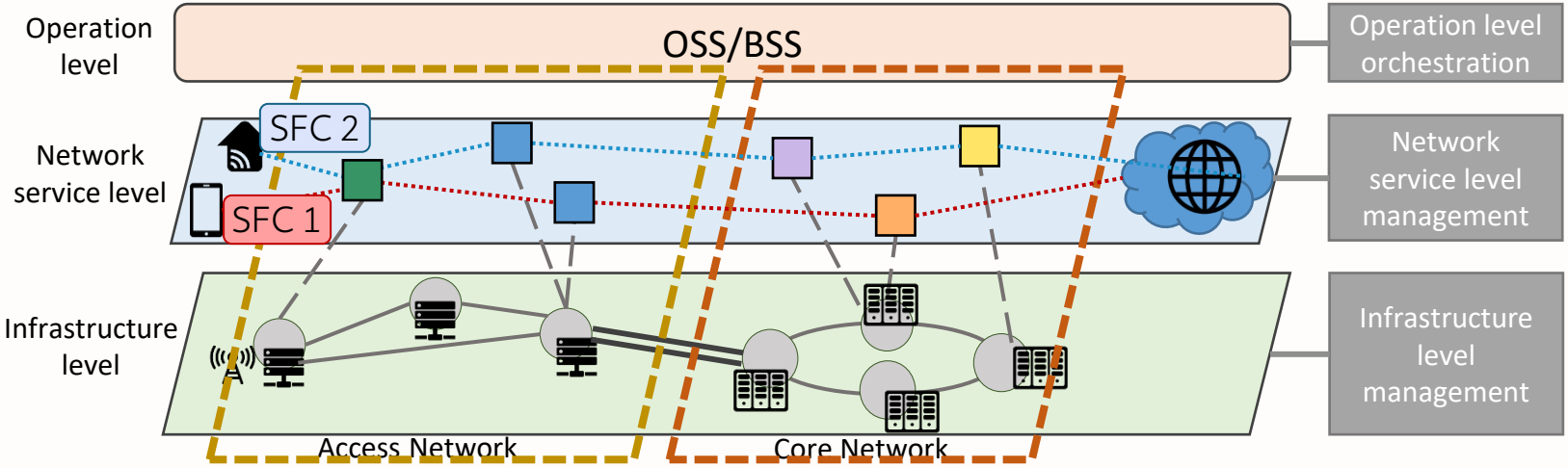
Main challenges for 5G system

5G ecosystem/verticals



5G evolution

5G service delivery



Chronology	Service delivered by
Before 5G	Network Functions (e.g., Firewall, loadbalancing)
5G and beyond	Virtual Network Functions

Where, when and how to deploy these applications ?
A flexible and real-time decision

Applications on servers

5G and Resilience(continued)

Resilience: maintain an acceptable level of service facing various incidents.

- availability
- latency
- communication reliability

Video Conference

Very low latency

Smart Factory

Availability

Entertain ment

Low packet loss

Transport

Communication reliability

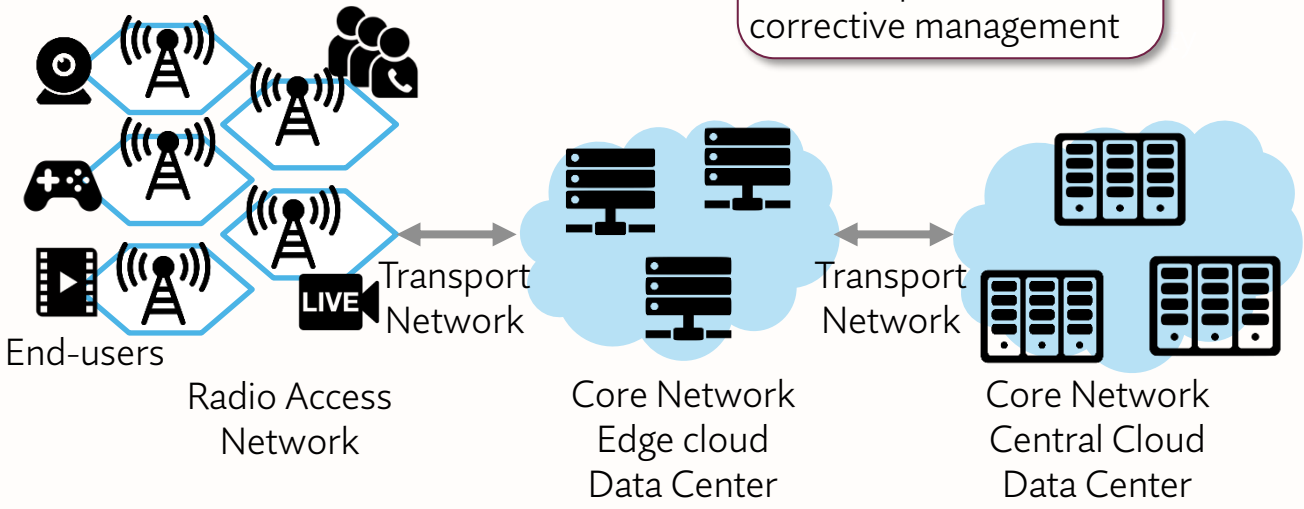
5G vertical engagements

5G networks are facing risks from RAN, TN, CN. Different vertexical industry may require different SLA.

5G and Resilience(continued)

Resilience: maintain an acceptable level of service facing various incidents.

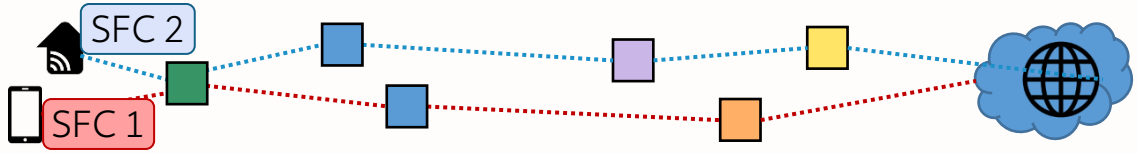
Focus on preventive and corrective management

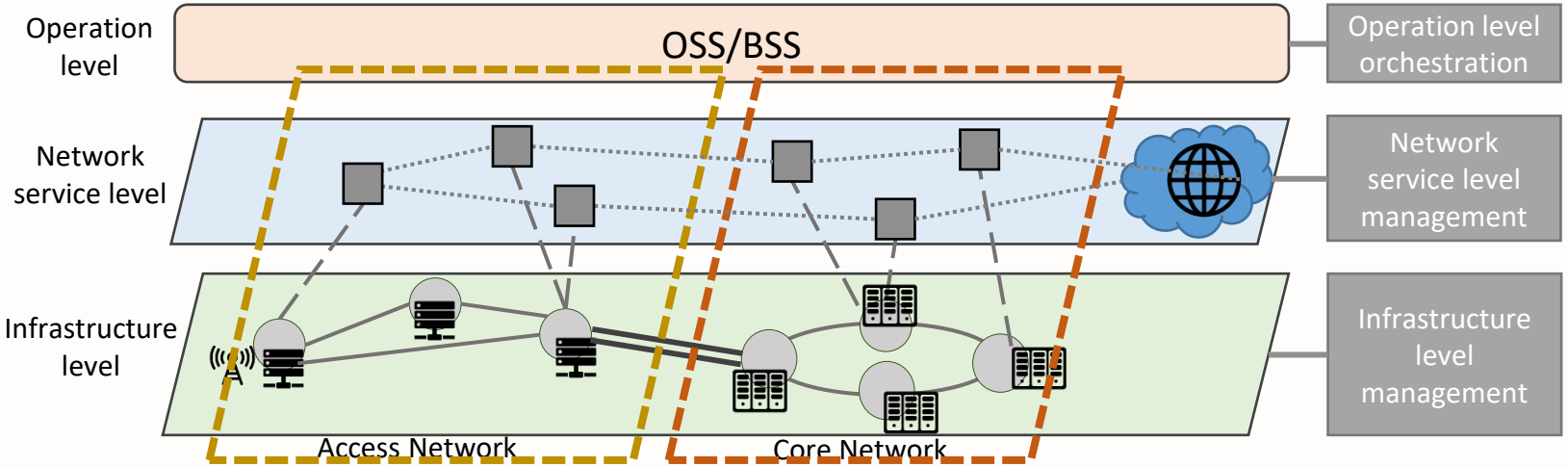


5G networks are facing risks from RAN, TN, CN. Different vertical industry may require different SLA.

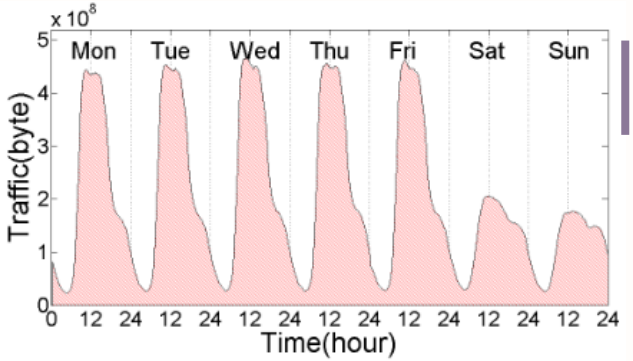
Different risks of a 5G network

- External risks:
  - **Traffic change**, attacks…
- Failure on different layer:
  - Damage of switches, servers or physical links or failure on virtual switch, VNF, virtual link…
- Failure propagation
  - Direct: from lower level to a higher level; Indirect: e.g. a rerouting action may congest other entities





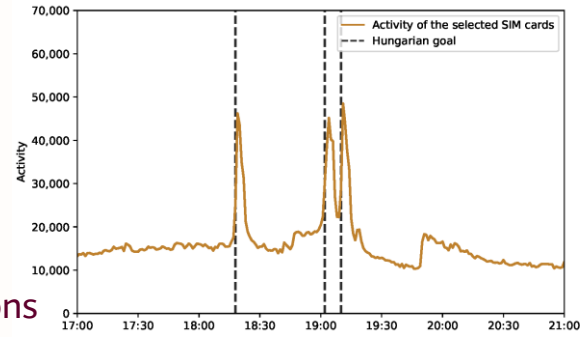
Traffic change, one of the most frequent risks



Predictable

Gracefully change

Mobile traffic in office area during a week[1]



Almost unpredictable

Short period variation causes sudden congestions

Mobile activity during a football match[2]

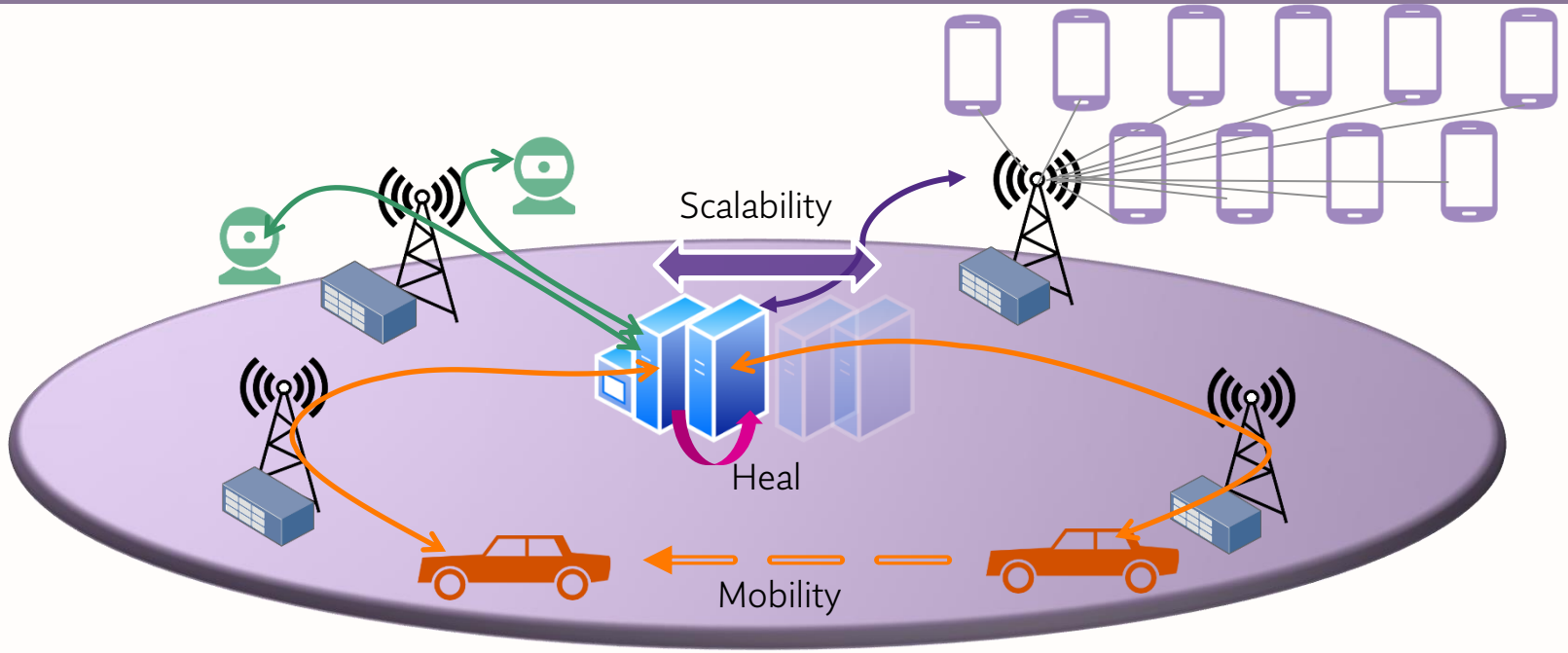
Flash events

Attack

Simultaneous work (maintenance/upgrade)

[1] Xu F, Li Y, Wang H, Zhang P, Jin D. Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment. *IEEE/ACM Transactions on Networking*. 2017; 25,(2) :1147-1161. doi: 10.1109/TNET.2016.2623950.
[2] Pintér G, Felde I. Analyzing the Behavior and Financial Status of Soccer Fans from a Mobile Phone Network Perspective: Euro 2016, a Case Study. *Information*. 2021; 12(11):468. doi:10.3390/info12110468

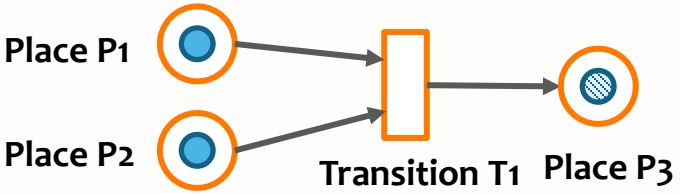
5G – flexible and dynamic system



5G network is a dynamic system. It has many dynamic behaviors.
Scalability: change the capacity according to current load.
Healing: repair automatically the failed component (virtual layer).
Mobility: connection (path, anchor point…) evolves with time.

Network modeling based on Petri Net

A Petri net, also known as a place/transition net



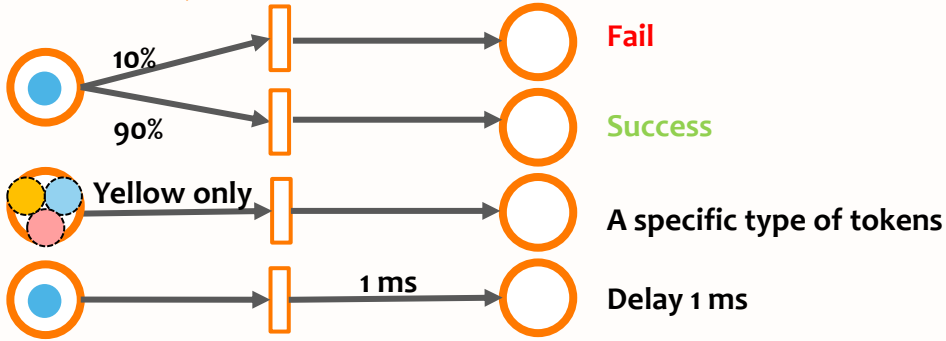
5-tuple: (P, T, F, W, M_0)

- P is a finite set of places
- T is a finite set of transitions
- F is a finite set of arcs with $F \subseteq (P \times T) \cup (T \times P)$ connecting places with transitions
- W is a multiset of arcs $(P \times T) \cup (T \times P) \rightarrow \mathbb{N}$, it assigns the weight
- M_0 is the initial marking of the Petri net graph

Classical Petri Net is not directly applicable to telecommunication systems.

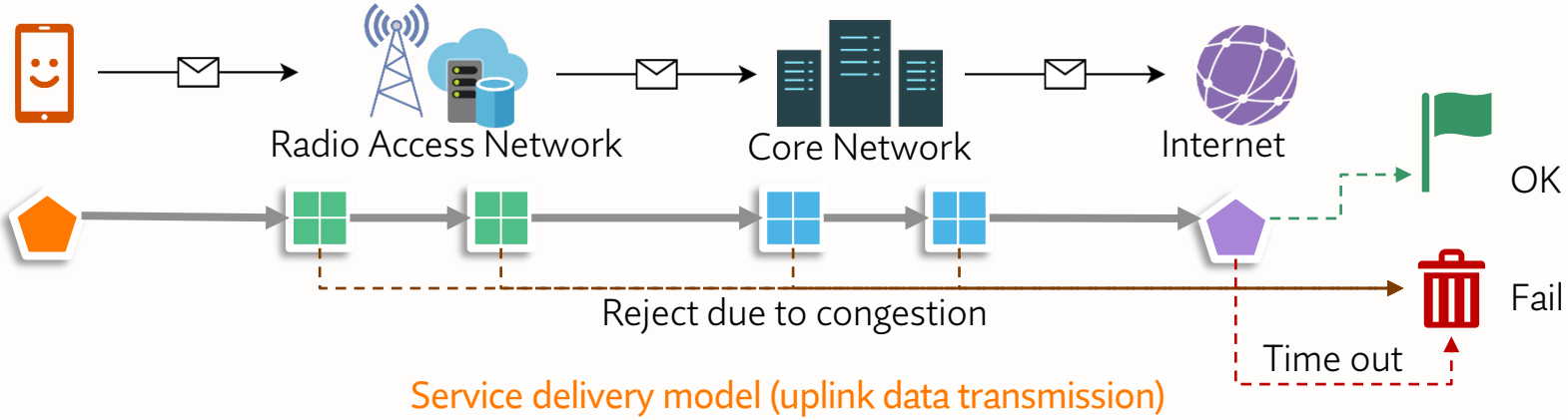
Some extensions:

- Stochastic Petri Nets --failure process
- Colored Petri Nets --token selection
- Timed Petri Nets --delay



- We consider a multi-tuple, Timed Stochastic Colored Petri Net, TSCPN=$(P, T, F, W, M_0, C, E, I, R, D)$

Service delivery model Petri Net



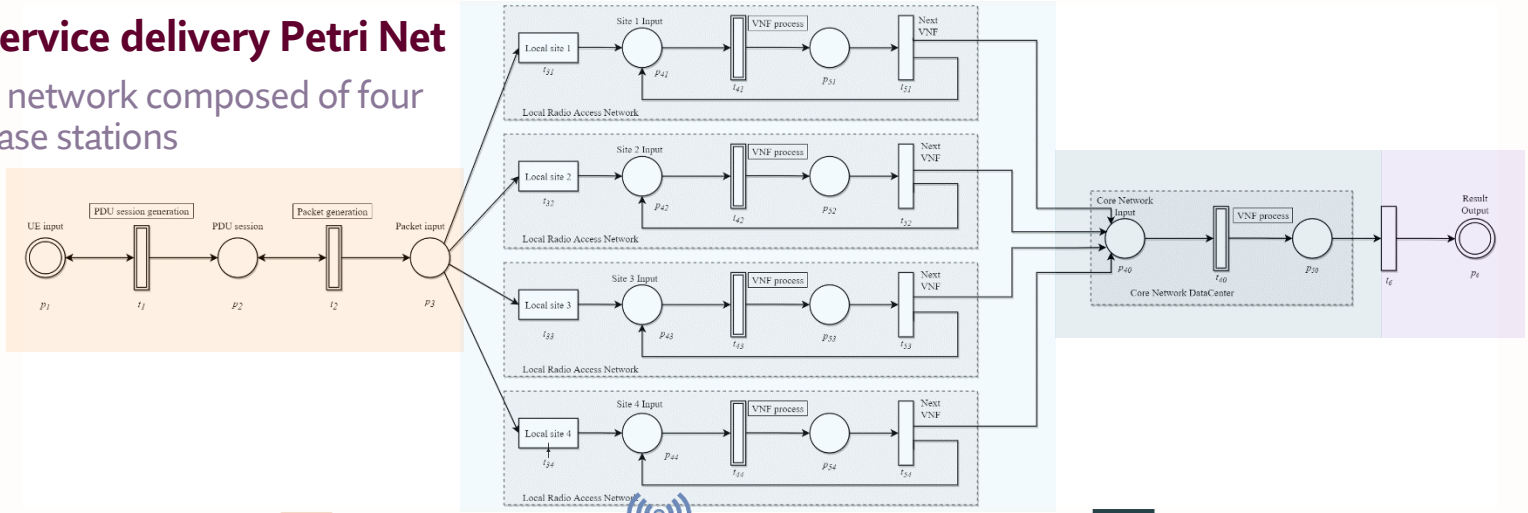
Service delivery model (uplink data transmission)

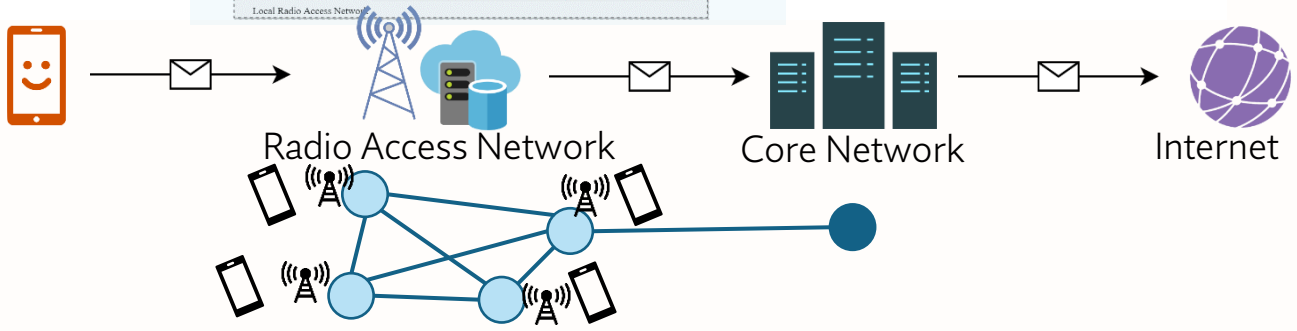
Service is delivered by following an acyclic service function chain consisting of Virtual Network Functions (VNFs);
Each VNF is containerized and contains subfunctions in form of containers/pods.

Congestion--queueing model

Service delivery Petri Net

A network composed of four base stations





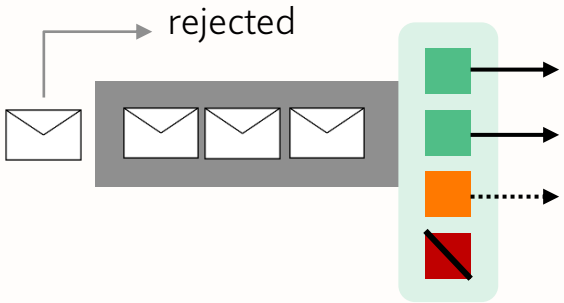
Congestion--queueing model (continued)

MS queue Petri Net

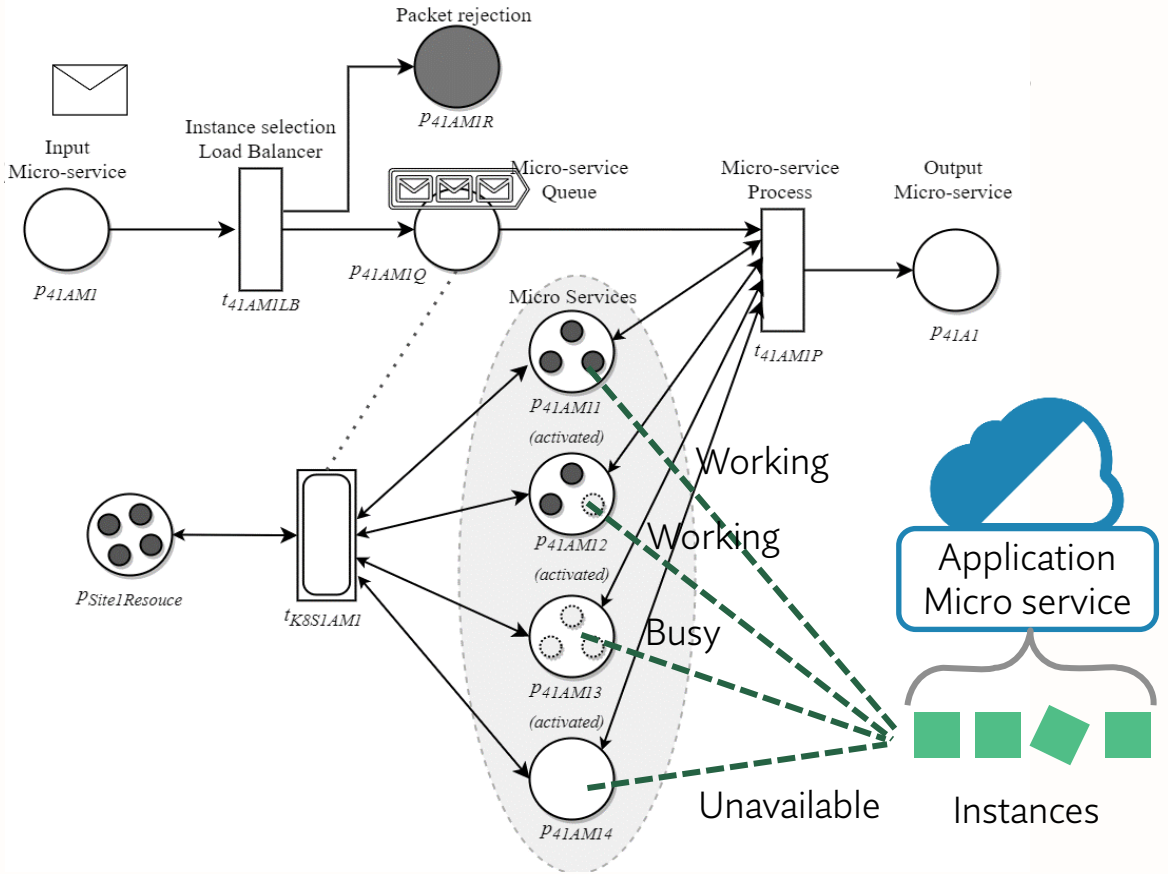
A Service Function Chain = a set of VNF and paths

A VNF = a set of subfunctions

A subfunction(microservice) = a set of instances containers (pods)



Queueing model

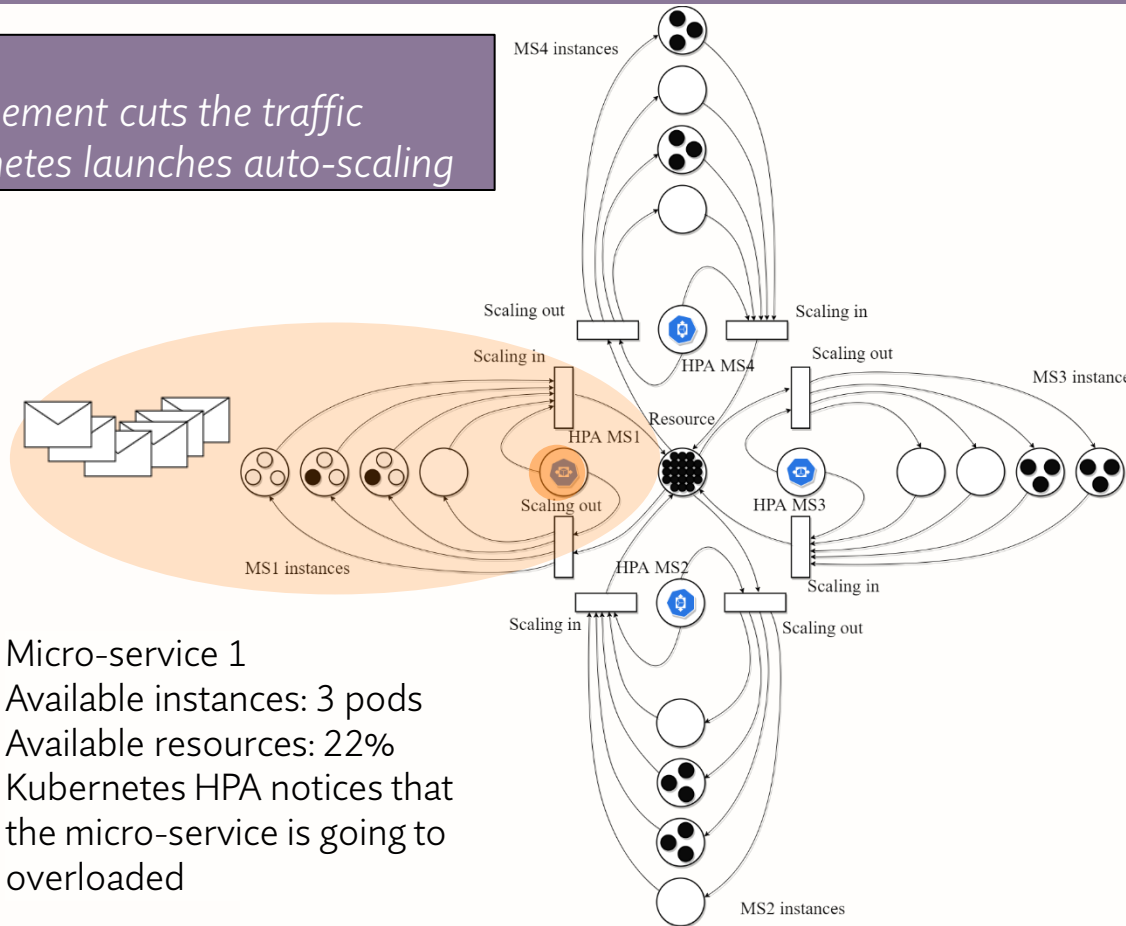


Congestion--auto-scaling

To cope with network congestion:

- ***Reduce traffic**: 5G network level management cuts the traffic*
- ***Increase capacity**: in 5G MANO, Kubernetes launches auto-scaling*

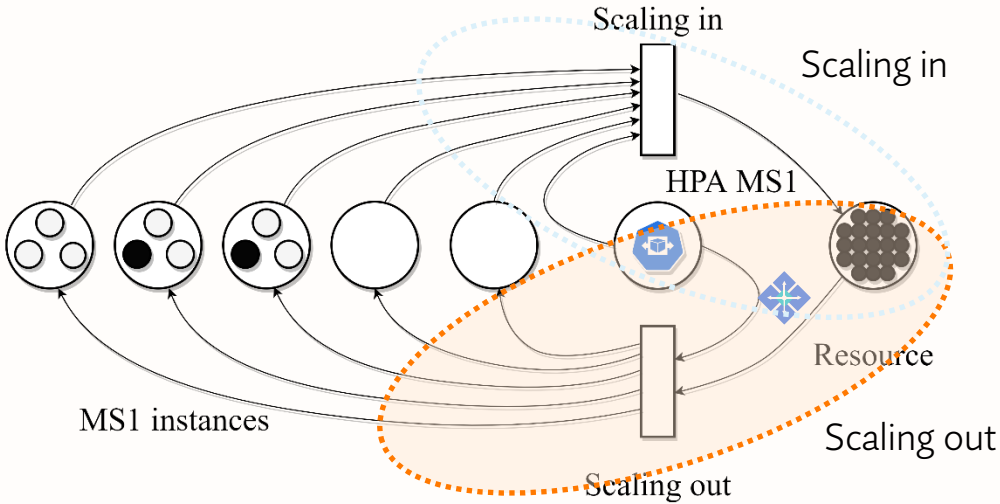
Kubernetes , the container management system, collects metrics (e.g., CPU usage…) values intermittently.
After analyzing these metrics, Kubernetes knows if the system (in level of VNF components) is overloaded or idle. Then it decides to create new instances or remove the existing ones.



Micro-service 1
Available instances: 3 pods
Available resources: 22%
Kubernetes HPA notices that the micro-service is going to overloaded

Kubernetes Auto-scaling out

Kubernetes , the container management system, collects metrics (CPU usage, memory usage …) values with a certain frequency.
After analyzing these metrics, Kubernetes knows if the system (in level of VNF components) is overloaded or too idle. Then based on this, it decides to create new instances or remove the existing ones.

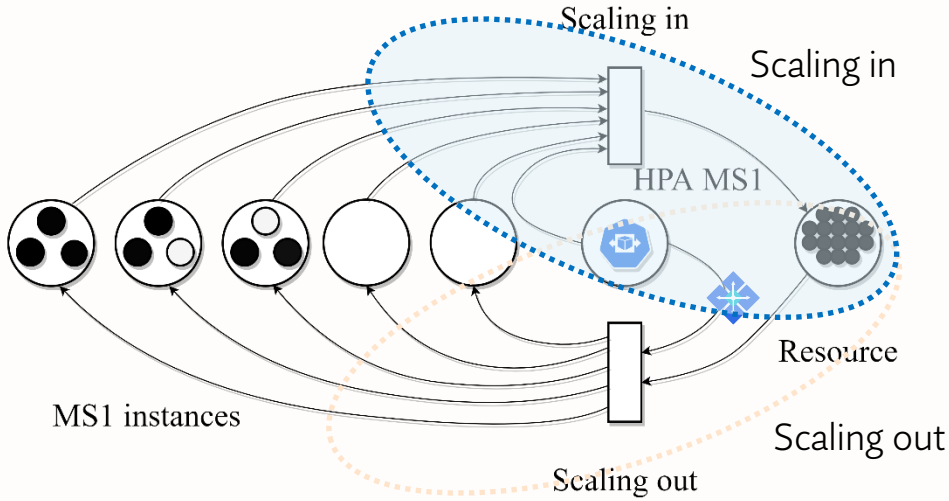


The suto-scaling process modeled by Petri Net

Desired CPU usage := 50%
Threshold_UP:= 60%
Threshold_DOWN := 30%
Current CPU usage is 78 %
Current CPU usage is 22 %
If Metrics > Threshold_UP:
    New_scale := Current CPU usage / Desired CPU usage [0.78/0.5*3 = 4.67 pods →5 pods]
    Deploy(New_scale)
# The network instances will be doubled
Elseif Metrics < Threshold_DOWN:
    New_scale := Current CPU usage / Desired CPU usage [0.22/0.5*3 = 1.33 pods 2 pods]
    Deploy(New_scale)
# The network instances will be halved

Kubernetes Auto-scaling in

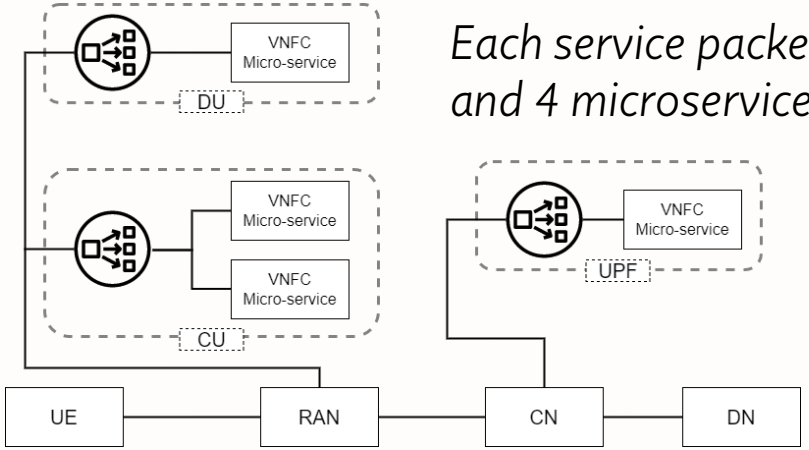
Kubernetes , the container management system, collects metrics (CPU usage, memory usage …) values with a certain frequency.
After analyzing these metrics, Kubernetes knows if the system (in level of VNF components) is overloaded or too idle. Then based on this, it decides to create new instances or remove the existing ones.



The suto-scaling process modeled by Petri Net

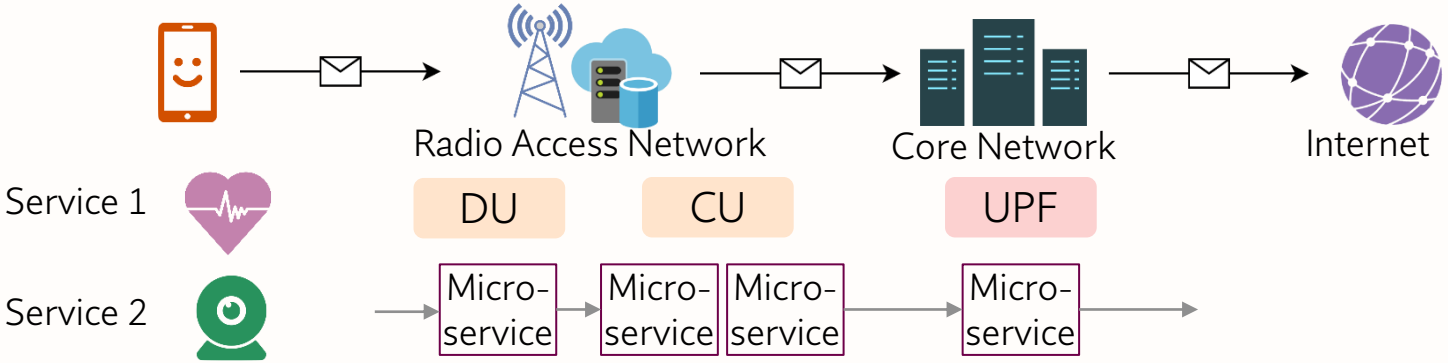
Desired CPU usage := 50%
Threshold_UP:= 60%
Threshold_DOWN := 30%
Current CPU usage is 78 %
Current CPU usage is 22 %
If Metrics > Threshold_UP:
New_scale := Current CPU usage / Desired CPU usage [0.78/0.5*3 = 4.67 pods →5 pods]
Deploy(New_scale)
# The network instances will be doubled
Elseif Metrics < Threshold_DOWN:
New_scale := Current CPU usage / Desired CPU usage [0.22/0.5*3 = 1.33 pods →2 pods]
Deploy(New_scale)
# The network instances will be halved

Network services



Each service packet is delivered via three VNF and 4 microservices

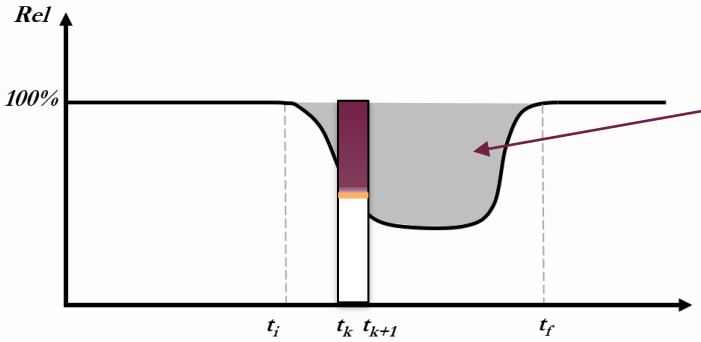
Service	Packet size	Frequency	Latency requirement
1. Critical	*Short*	*High*	*10 ms*
2. IoT	*Long*	*Low*	*50 ms*



5G network resilience performance evaluation

Resilience: maintain an acceptable level of service facing various incidents.

Reliability[1] in the context of network layer packet transmissions: percentage value of the packets successfully delivered to a given system entity within the time constraint required by the targeted service out of all the packets transmitted.



Not rejected — *Packet loss*

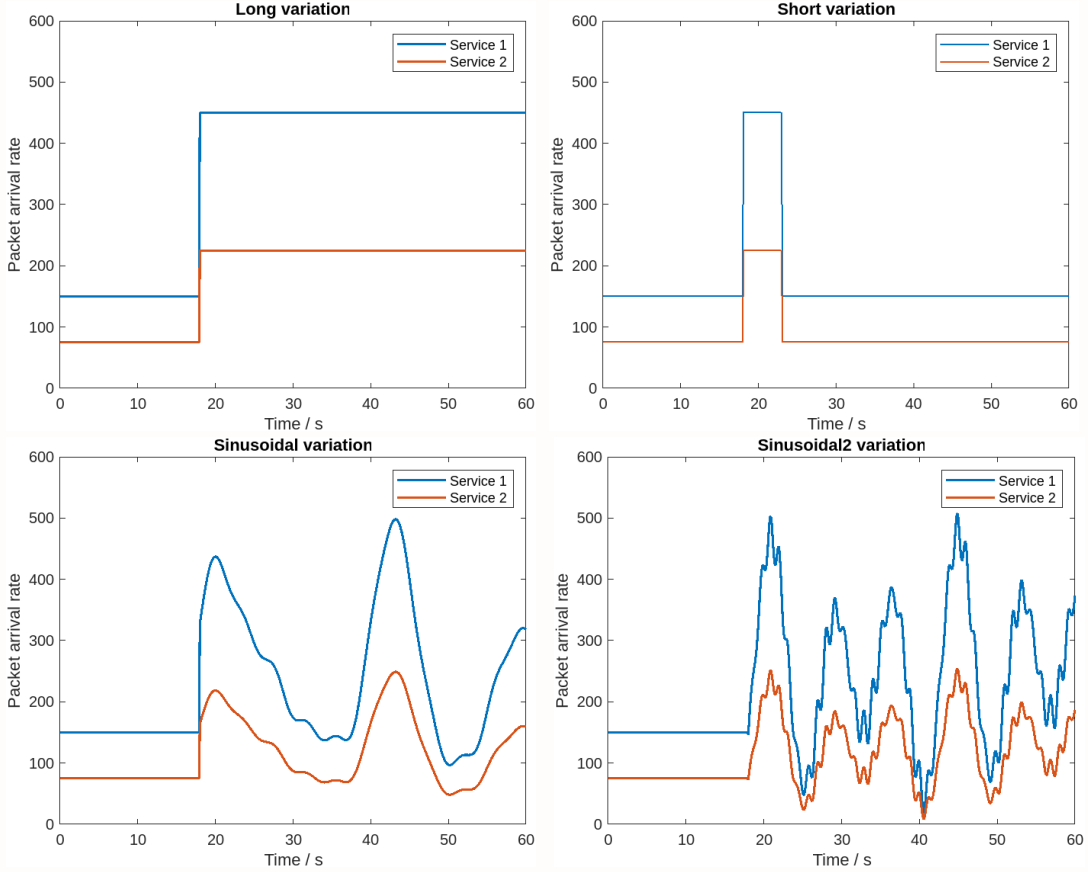
Within delay limit — *Latency*

$$R = \int_{t_i}^{t_f} [1 - Rel(t)]dt = \sum_{t=t_1}^{t_K} [100\% - SR(t_k)]\Delta T$$

$SR(_{tk})$: Discretized service reliability
Percentage of packets delivered to the internet within time limit during time slot $[t_k, t_{k+1}[$.

[1]Third Generation Partnership Project (3GPP), "Management and orchestration; 5G performance measurements (Release 16)," 3GPP TS 22.261 version 16.14.1 Release 16, July 2022.

Network traffic variation



From 0 to 18 seconds:

preparation/ Launch PDU sessions

arrival rate:

150 packet/PDU·s for service 1 or 75 packets/PDU·s for service 2

From 18 to 60 seconds:

Traffic change with different patterns

Entropy LV = 0.0108
Entropy SV = 0.0207
Entropy S1 = 0.1019
Entropy S2 = 0.3676

Network traffic variation (continued)

3 strategies for scaling

Nothing

- Wait until traffic variation stops

Autoscaling – intermittence = 5 s

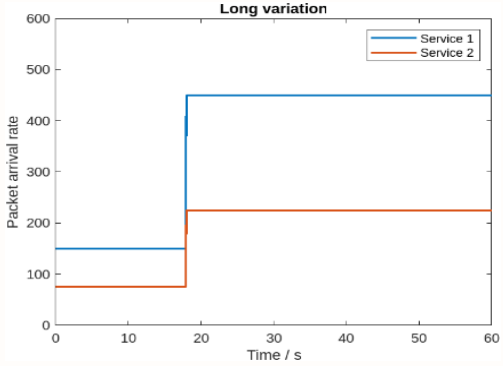
- Take actions based on the traffic load
- Check every 5 seconds

Autoscaling – intermittence = 5 s, windows = 15 s

- Take actions based on the decision made during last 15 seconds
- Check every 5 seconds

Scaling out	Scaling out	Scaling in	Scaling out
T-15	T-10	T-5	T

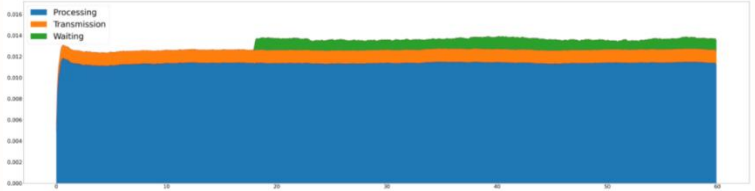
Network traffic variation – traffic pattern 1



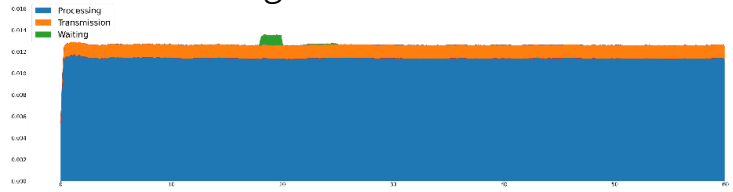
Simulation result of traffic pattern 1

Strategy		Average latency (ms)	Resilience loss (second)	Resource cost (CPU·s)
Long variation				
No AS	Serv. 1	10.437(±0.014)	24.082(±0.300)	12000
	Serv. 2	18.863±0.014)	1.460(±0.044)	
Basic AS	Serv. 1	9.742(±0.003)	1.476(±0.037)	20067(±70)
	Serv. 2	18.195(±0.005)	0.073(±0.005)	
Win. AS	Serv. 1	9.889(±0.007)	6.584(±0.141)	17934(±42)
	Serv. 2	18.337(±0.008)	0.445(±0.026)	

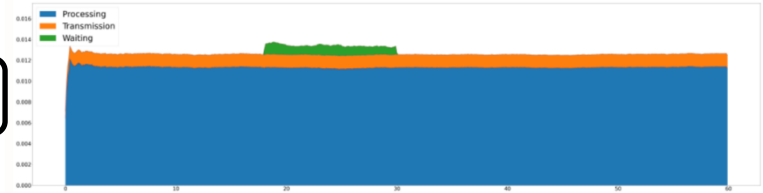
No scaling



Basic auto-scaling

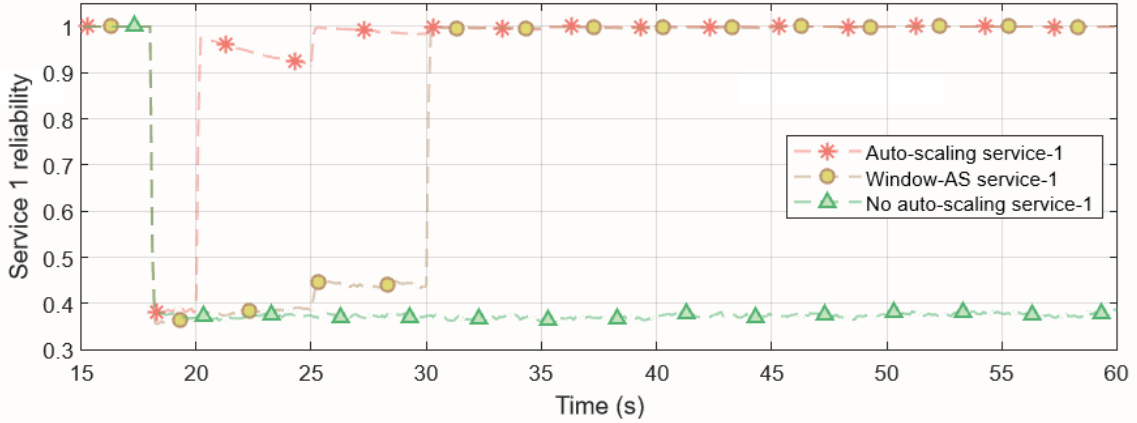


Window based auto-scaling

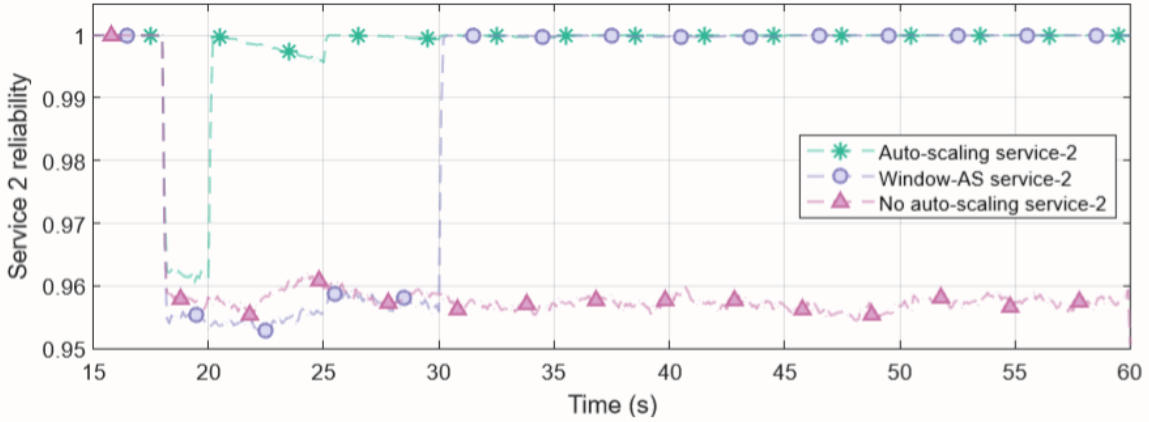


Long term traffic interruption

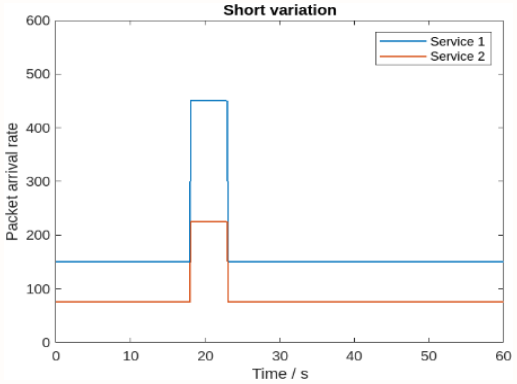
Service 1
Reliability evolution



Service 2
Reliability evolution



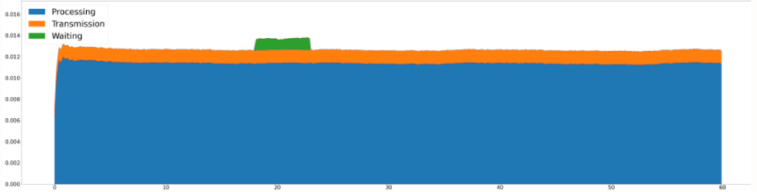
Network traffic variation – traffic pattern 2



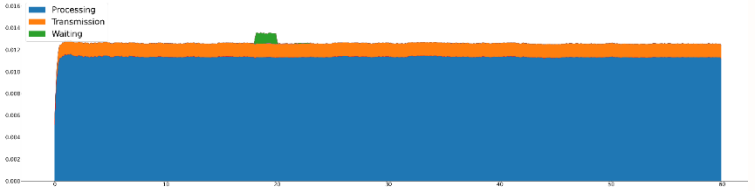
Simulation result of traffic pattern 2

Strategy		Average latency (ms)	Resilience loss (second)	Resource cost (CPU·s)
Short variation				
No AS	Serv. 1	9.786(±0.004)	2.867(±0.070)	12000
	Serv. 2	18.244(±0.006)	0.172(±0.011)	
Basic AS	Serv. 1	9.738(±0.003)	1.404(±0.033)	15364(±55)
	Serv. 2	18.194(±0.005)	0.070(±0.005)	
Win. AS	Serv. 1	9.785(±0.004)	2.830(±0.070)	12275(±16)
	Serv. 2	18.240(±0.005)	0.164(±0.010)	

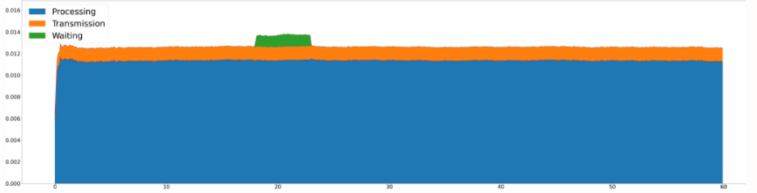
No scaling



Basic auto-scaling

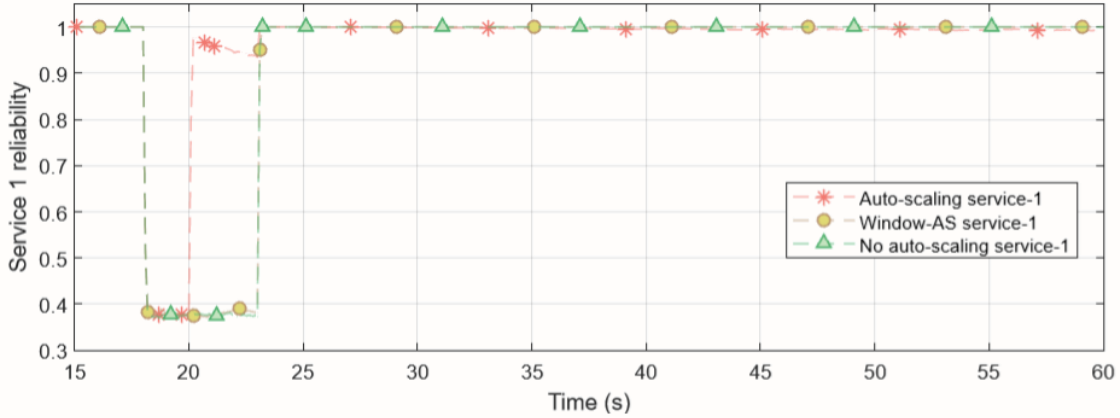


Window based auto-scaling

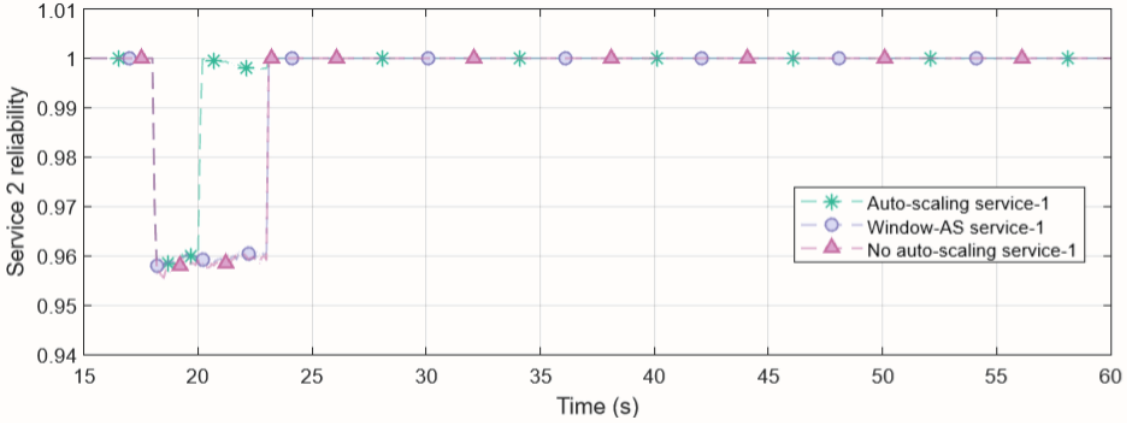


Short term traffic interruption

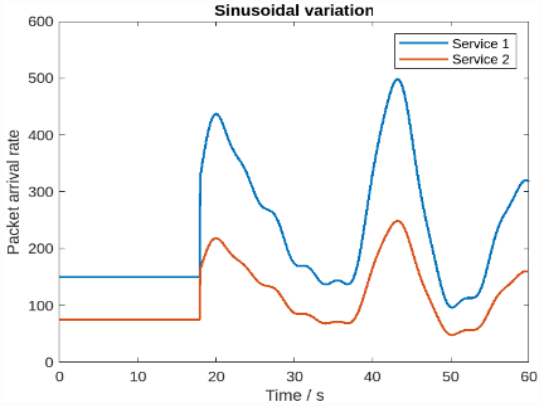
Service 1
Reliability evolution



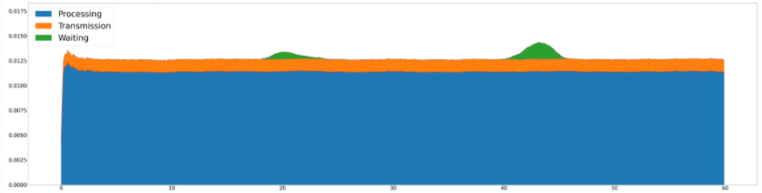
Service 2
Reliability evolution



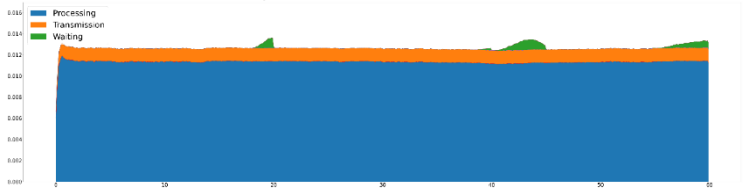
Network traffic variation – traffic pattern 3



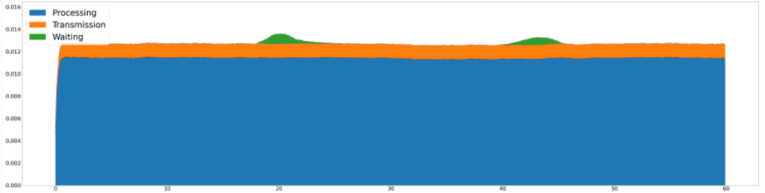
No scaling



Basic auto-scaling



Window based auto-scaling

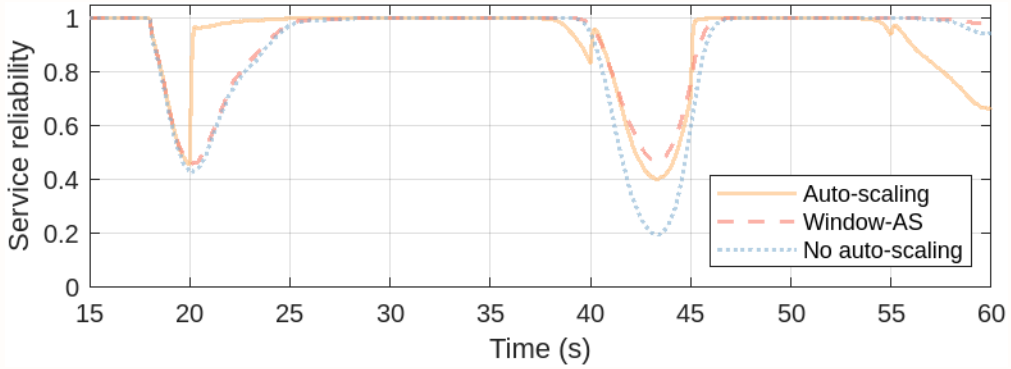


Simulation result of traffic pattern 3

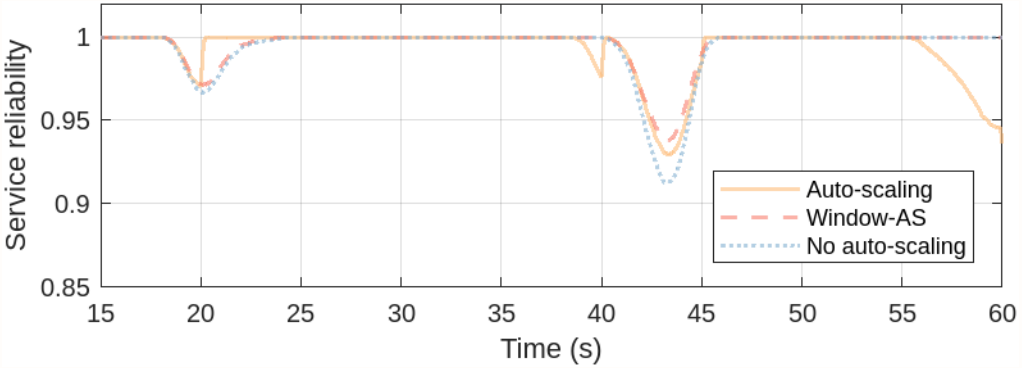
Strategy		Average latency (ms)	Resilience loss (second)	Resource cost (CPU·s)
Sinusoidal 1				
No AS	Serv. 1	9.832(±0.004)	4.640(±0.092)	12000
	Serv. 2	18.281(±0.006)	0.247(±0.011)	
Basic AS	Serv. 1	9.793(±0.004)	3.676(±0.092)	18320(±74)
	Serv. 2	18.252(±0.005)	0.310(±0.015)	
Win. AS	Serv. 1	9.780(±0.004)	3.394(±0.088)	13375(±28)
	Serv. 2	18.233(±0.005)	0.182(±0.010)	

Sinusoidal traffic interruption

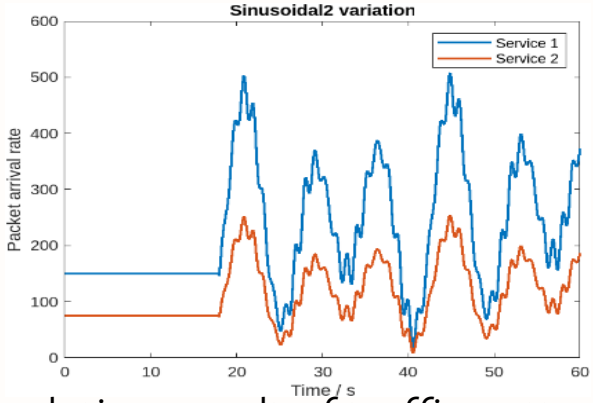
Service 1
Reliability evolution



Service 2
Reliability evolution



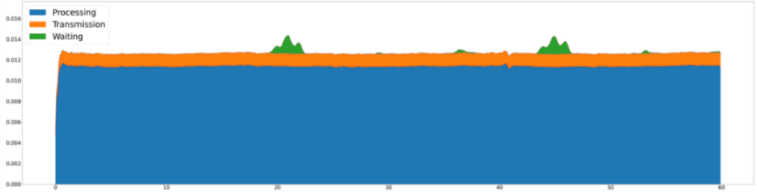
Network traffic variation – traffic pattern 4



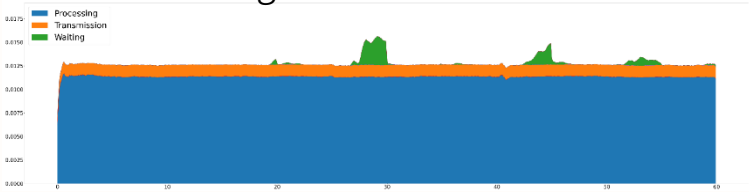
Simulation result of traffic pattern 4

Strategy		Average latency (ms)	Resilience loss (second)	Resource cost (CPU·s)
Sinusoidal 2				
No AS	Serv. 1	9.811(±0.004)	4.120(±0.077)	12000
	Serv. 2	18.264(±0.005)	0.200(±0.008)	
Basic AS	Serv. 1	9.911(±0.004)	5.946(±0.093)	19498(±81)
	Serv. 2	18.359(±0.006)	1.023(±0.026)	
Win. AS	Serv. 1	9.796(±0.037)	3.890(±0.074)	12249(±18)
	Serv. 2	18.255(±0.005)	0.187(±0.008)	

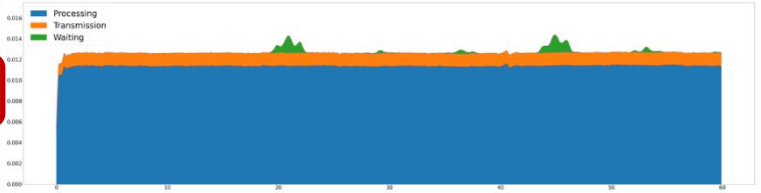
No scaling



Basic auto-scaling

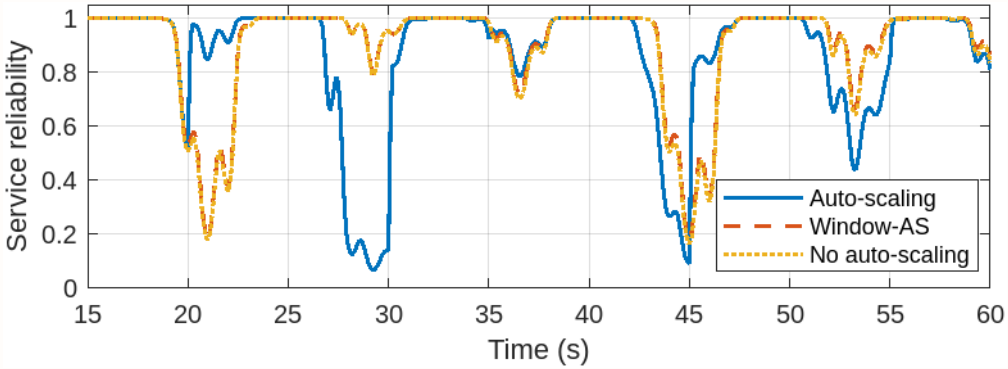


Window based auto-scaling

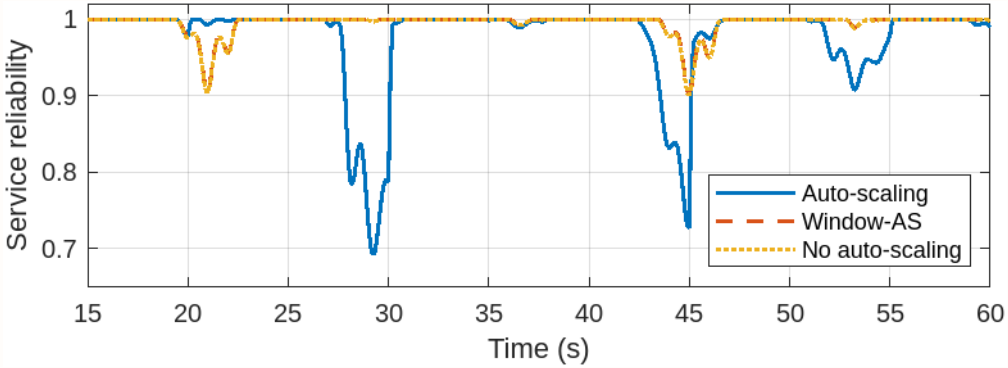


Sinusoidal traffic interruption 2

Service 1
Reliability evolution



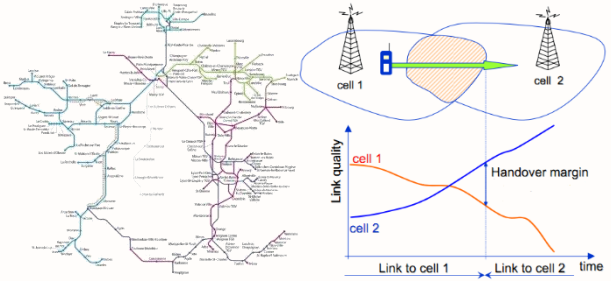
Service 2
Reliability evolution



Conclusion

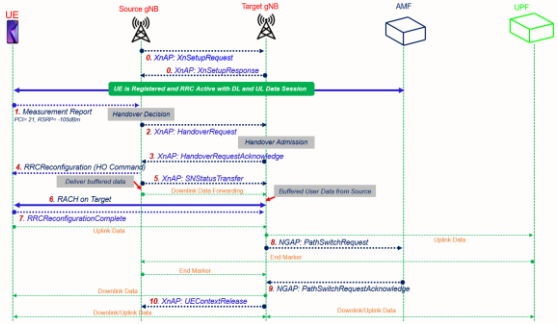
Conclusions

- A Petri Net-based model is constructed to represent the dynamics of 5G networks;
- Different management mechanisms are compared;
- 5G network service performance is evaluated from a resilience point of view.



Prospects

- AI-based management can be more usefull when the time series Entropy increases;
- More precise parameters to be collected from Orange experts / equipment suppliers;
- We will not stop in the data plane packet transfer but also a complete 5G architecture with signaling part.



Thank You

Orange Innovation

Rui Li

rui.li@orange.com

Bertrand Decocq

bertrand.decocq@orange.com

CentraleSupélec
Université Paris-Saclay

Anne Barros

anne.barros@centralesupelec.fr

Yiping Fang

yiping.fang@centralesupelec.fr

Zhiguo Zeng

zhiguo.zeng@centralesupelec.fr