THALES

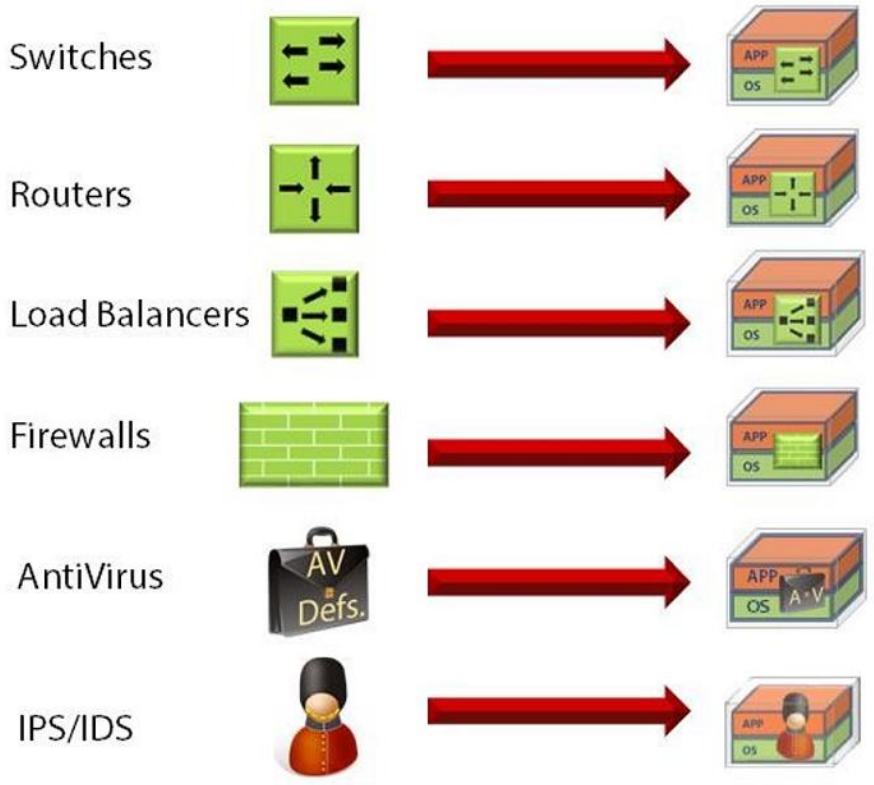
NFV orchestration

VANIA CONAN
WITH M. BOUET, M. OBADIA, N. TASTEVIN

JUNE 2017

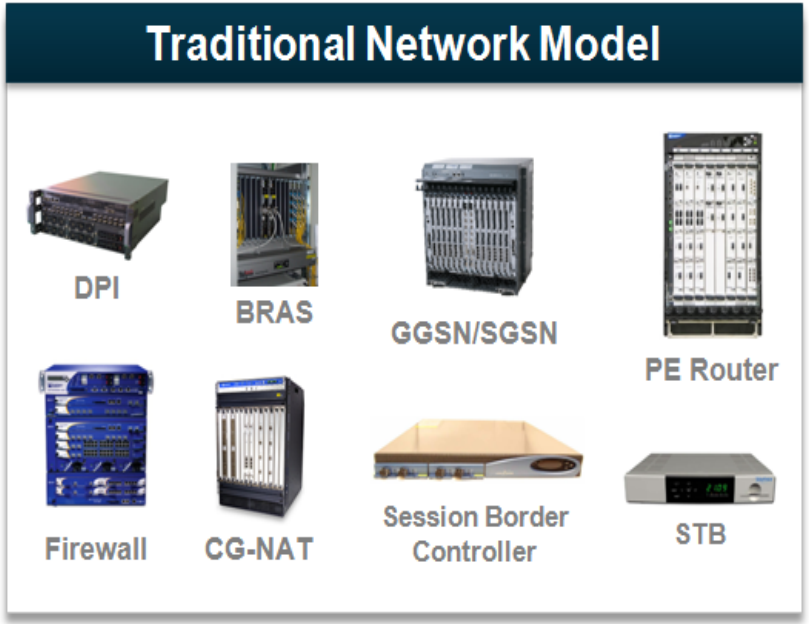
Virtual Network Function (VNF)

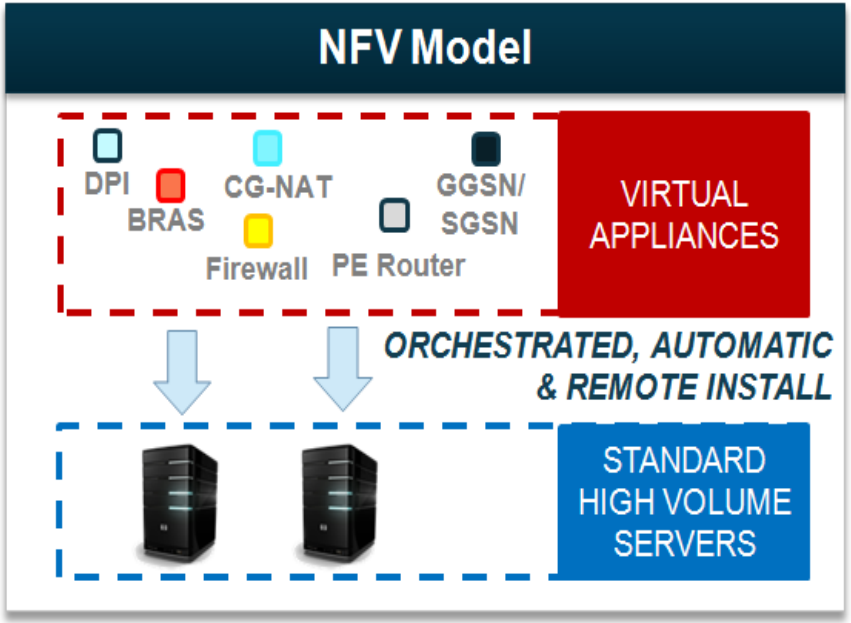
- Network equipment are provided as software components
- Functions are both basic networking functions (switching, routing) and middleboxes (firewalls, IDS, …)



THALES

Network Function Virtualisation (NFV) objectives





Removing tightly coupled network function's software from underlying hardware

THALES

NFV History

NFV: an operator-led initiative with direct impact on the Telecommunications industry business model

ETSI World Class Standards

THALES

NFV principles

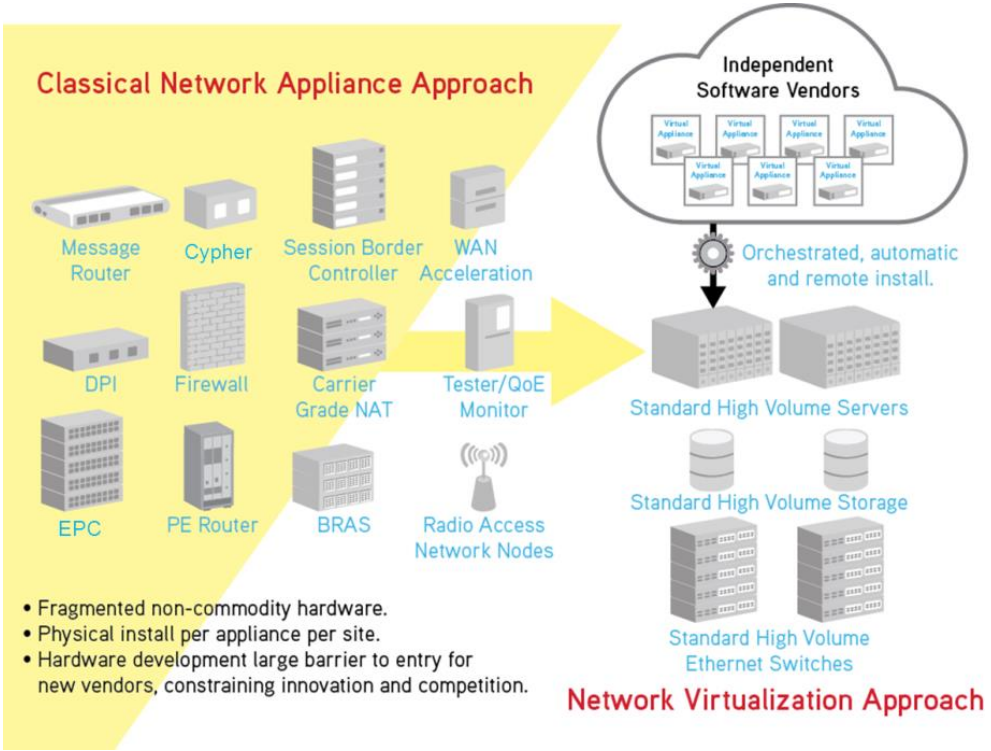
Convergence of IT and Network

Service-oriented multi-tenant systems

- pay as you go, on demand…

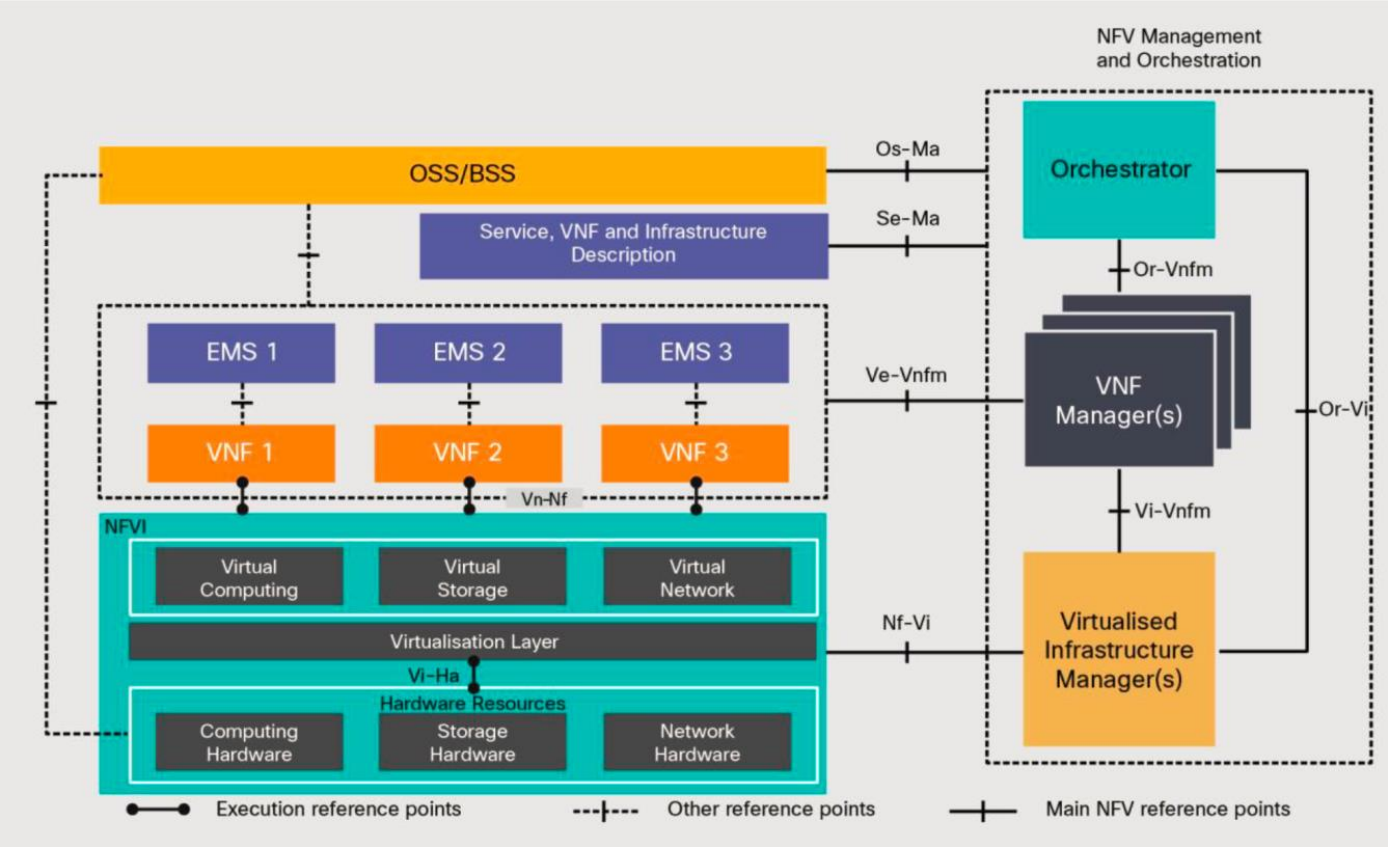
Orchestration

- programmability, virtualization, automation…



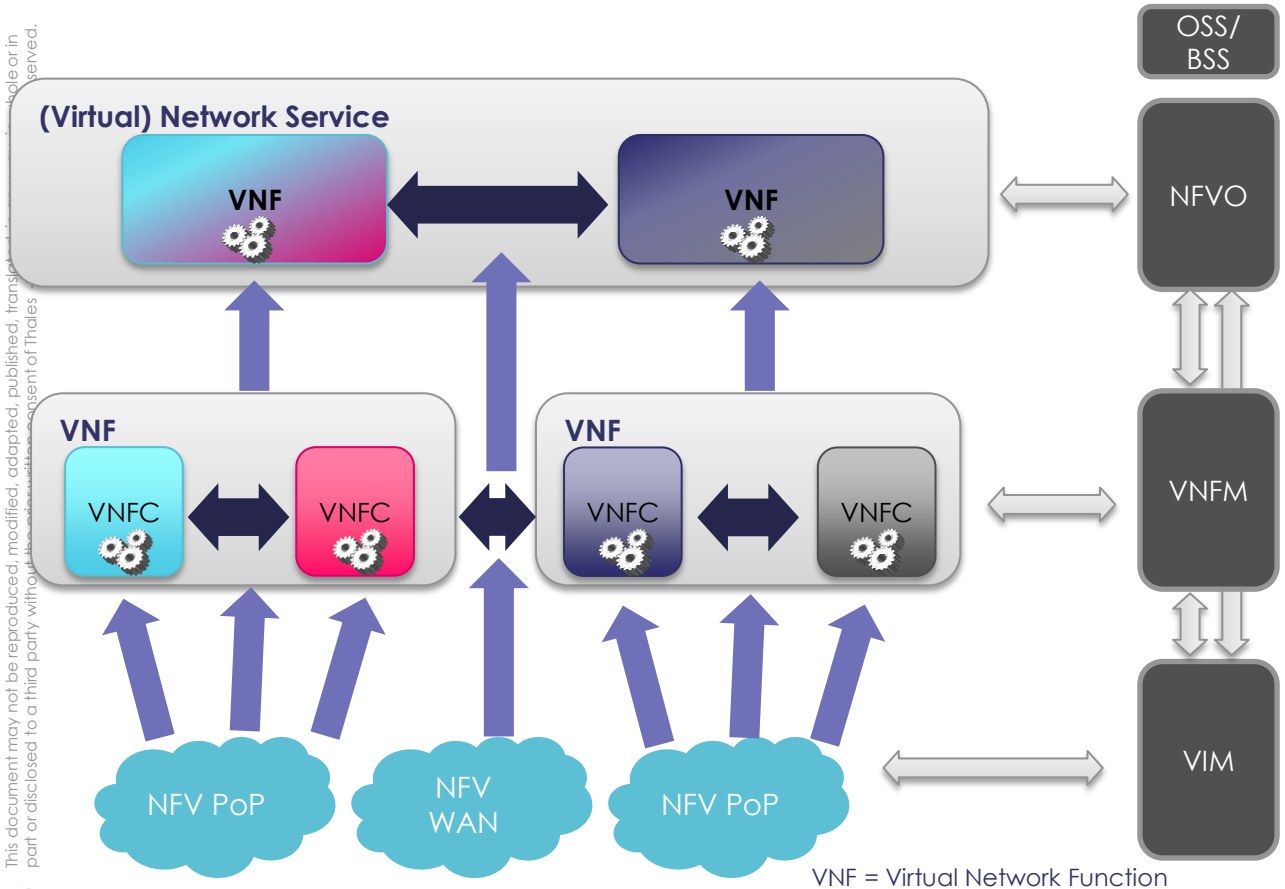
THALES

NFV Management and orchestration



THALES

ETSI NFV Reference Architectural Framework



VNF = Virtual Network Function
VNFC = VNF Container
PoP = Point of Presence

NFV Orchestrator:

- on-boarding of new Network Service (NS), VNF-FG and VNF Packages
- NS lifecycle management (including instantiation, scale-out/in, performance measurements, event correlation, termination)
- global resource management, validation and authorization of NFVI resource requests
- policy management for NS instances

VNF Manager:

- lifecycle management of VNF instances
- overall coordination and adaptation role for configuration and event reporting between NFVI and the E/NMS

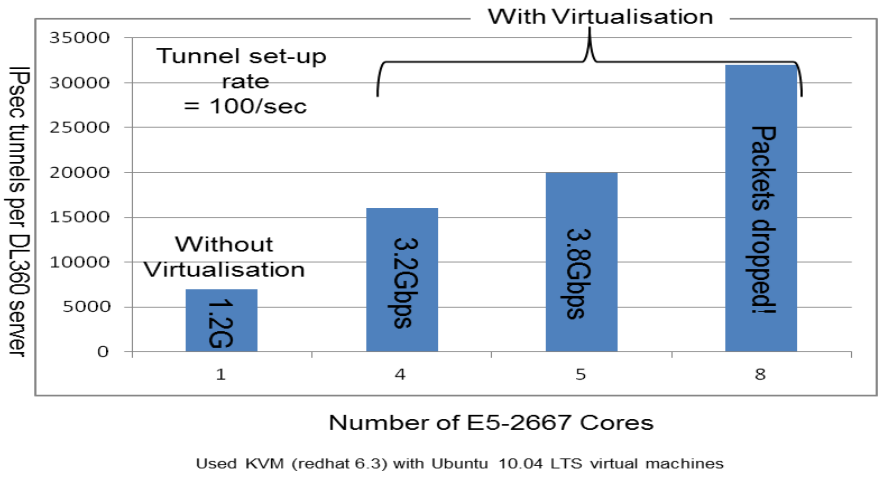
Virtualized Infrastructure Manager (VIM):

- control and manage the NFVI compute, storage and network resources
- collection and forwarding of performance measurements and events

THALES

NFV is on its way

- network functions can operate at the level of several millions of packets per sec, per CPU core
- standard high volume servers have sufficient processing performance to cost-effectively virtualize network appliances
- The hypervisor need not be a bottleneck
- The OS need not be a bottleneck



- Total Cost of Ownership advantages are a huge driver – e.g., energy savings
- A concerted industry effort is underway to accelerate this vision by encouraging common approaches which address the challenges for NFV

THALES

NFV and orchestration

- Networks are made of software : they need to be engineered, designed, planned and operated as such
- Major change in network engineering
- Why it matters and what does it change for system design : Thales business example

THALES

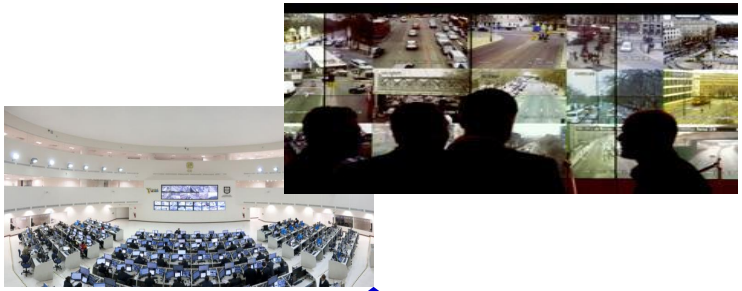
Mexico – largest urban security system

- Mexico – 22 Million people
- Mexico City's urban security system comprises: two mobile C2s, four MUAV drones, 210 number plate recognition cameras, integrated communications, 20000 advanced video surveillance cameras, centralisation of incidents and reporting

Multi-agencies coordination system
Video surveillance- > 20K cameras
Crisis management system
Emergency response time cut by three
Crime in the metro has fallen by 80%
Car thefts reduced by 10%
Criminal activity down by 35%.

THALES

System architecture



Central supervision

- Video surveillance control room

Communications network

Up/Downlink Video Traffic

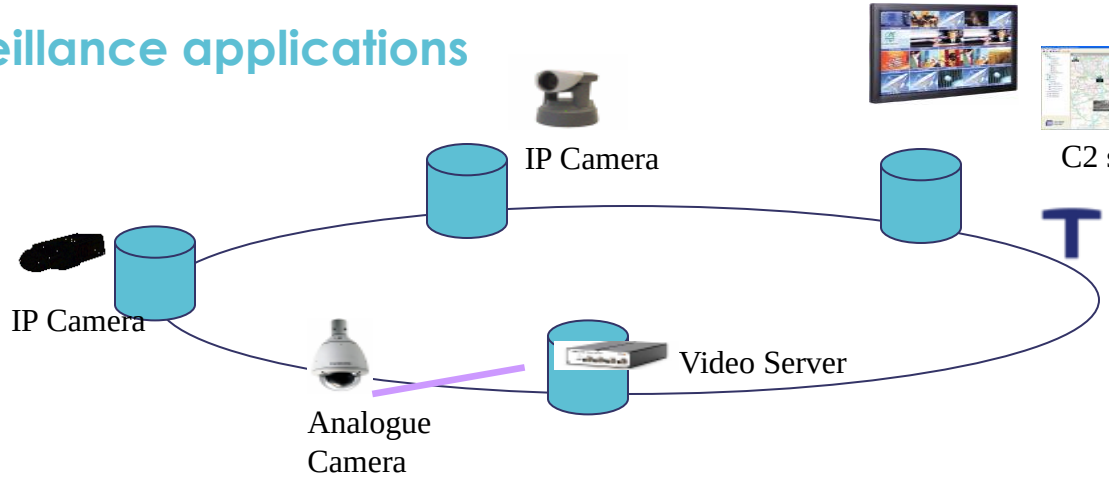
Uplink Video Traffic

Downlink Video Traffic

Group Call, Individual Call, Emergency Call

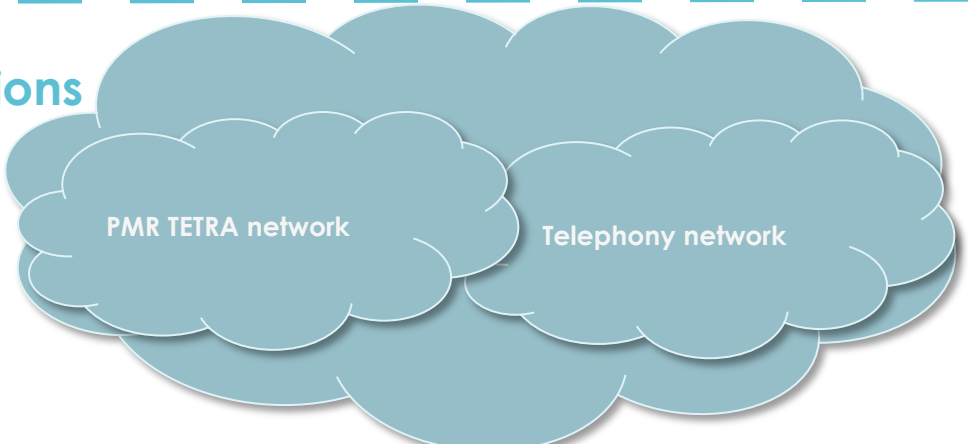
Mexico surveillance business split

Video surveillance applications



THALES

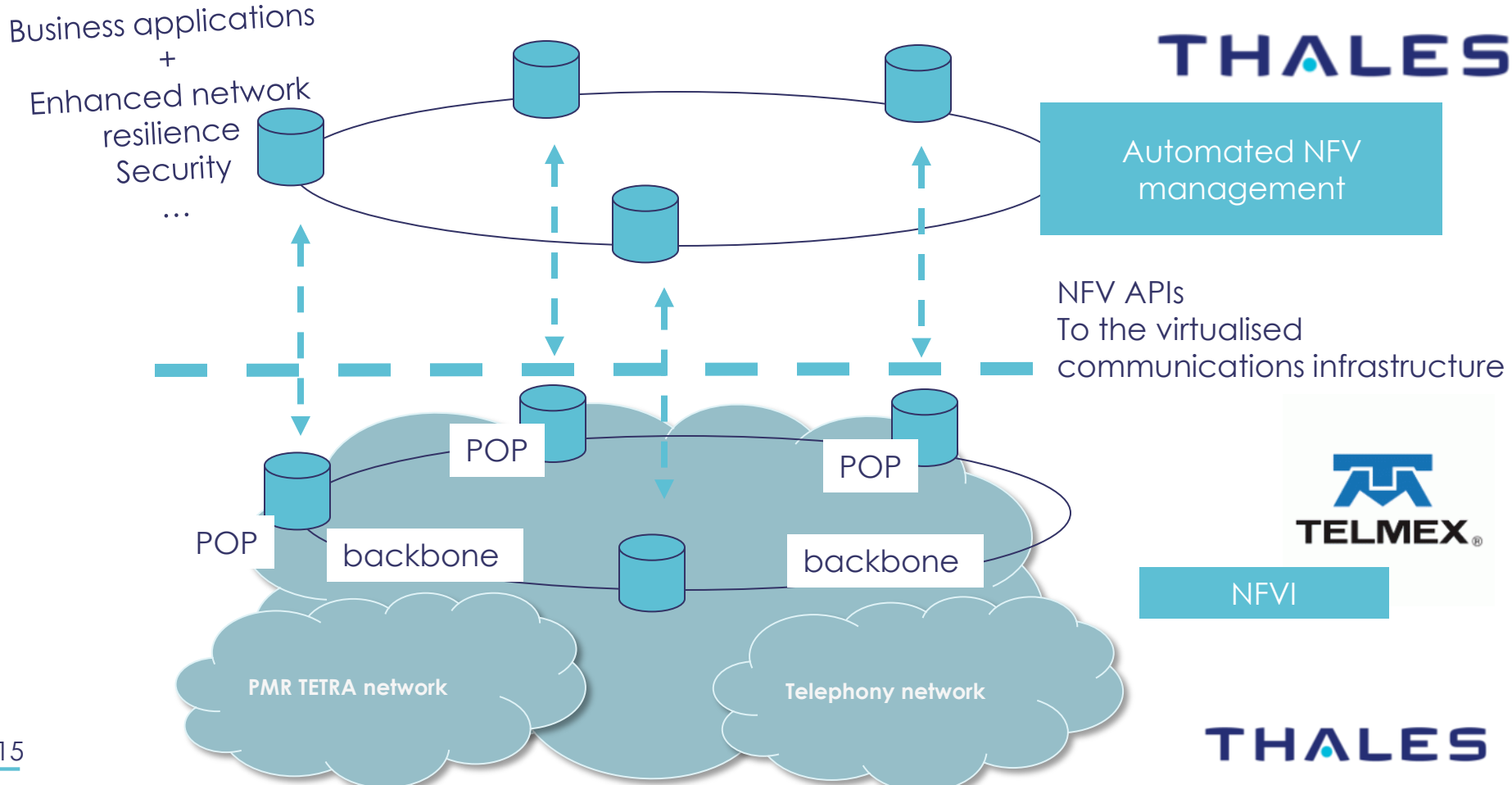
Telecommunications network



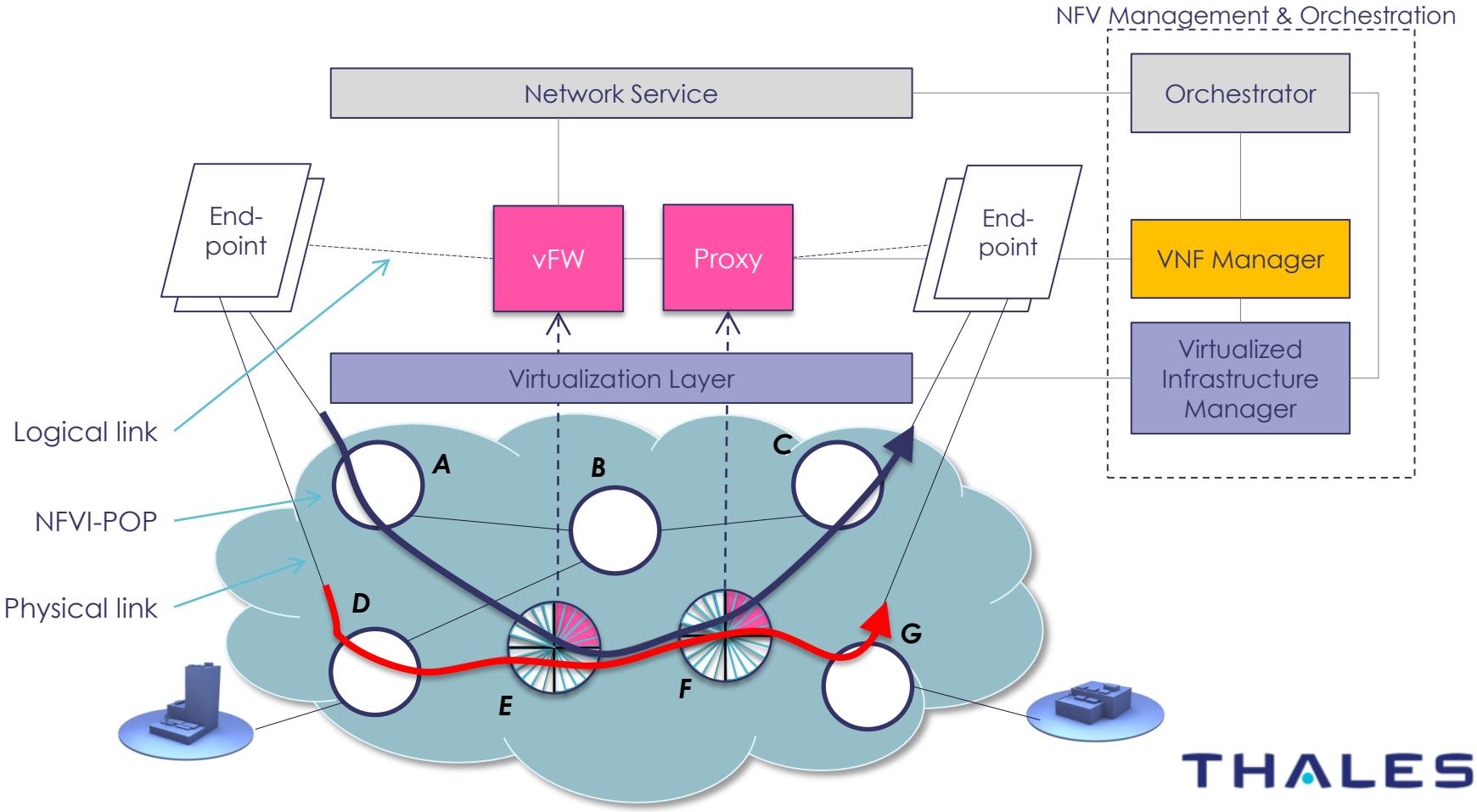
TELMEX®

THALES

NFV based virtualised videosurveillance solution



NFV orchestration: placement and chaining challenge



THALES

NFV orchestration

New for network design and operation: network build, run

New challenges of service orchestration and chaining

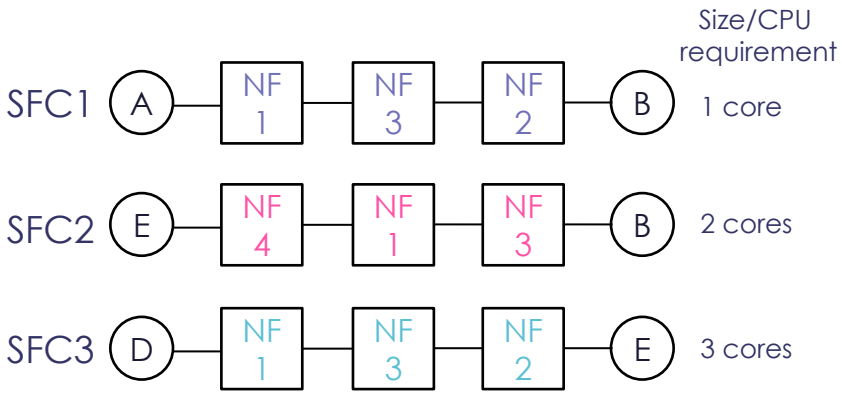
- Network build: Off-line (system provisioning)
- Network run: On-line (system operation)

THALES

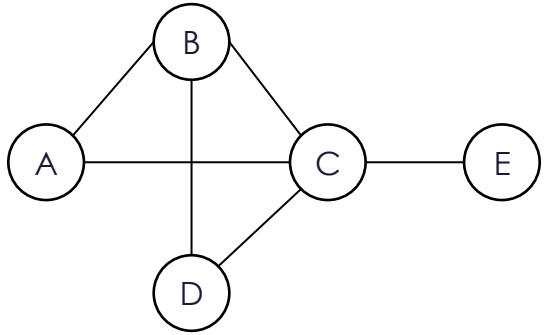
Off-line placement and chaining

THALES

Challenge overview



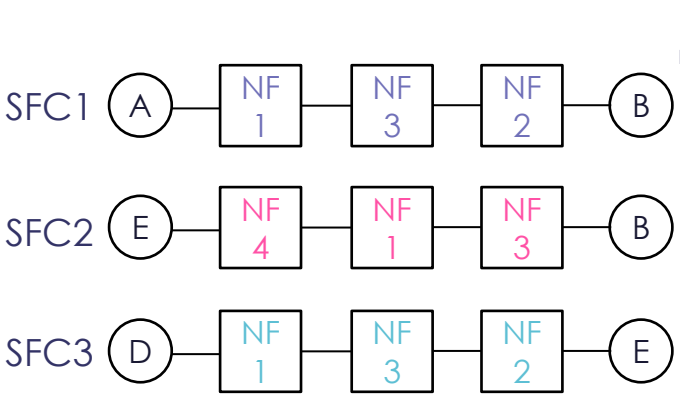
Service Function Chains



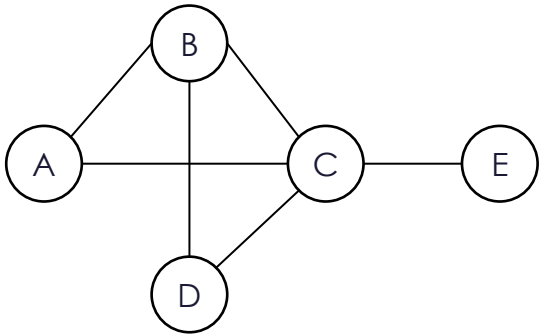
NFV Infrastructure

THALES

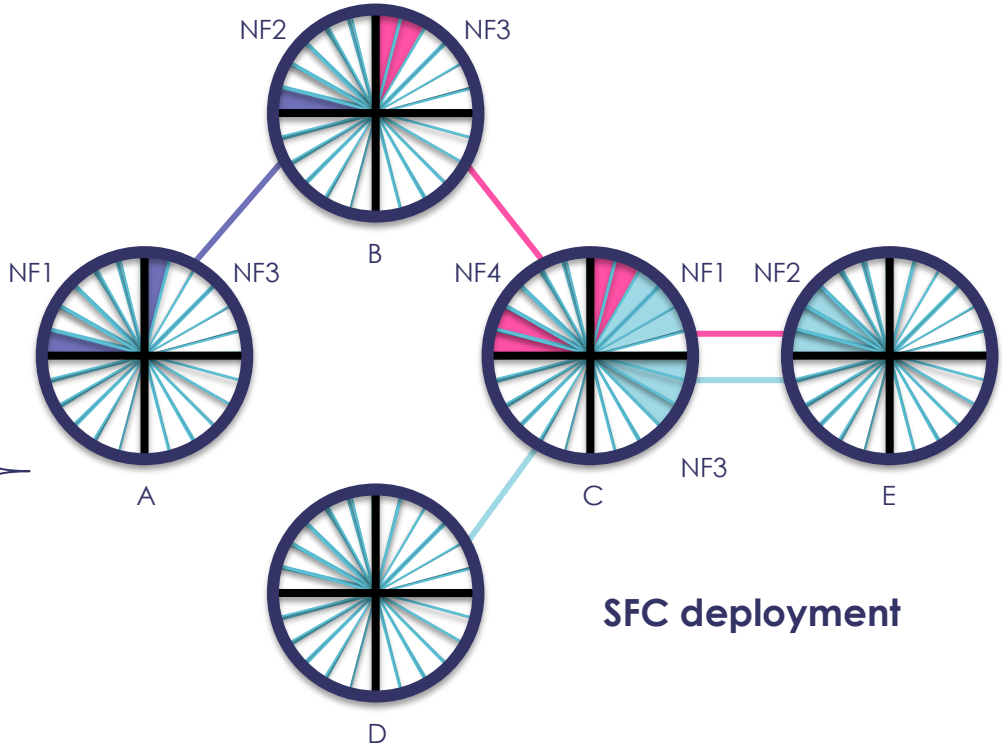
Challenge overview



Service Function Chains



NFV Infrastructure



SFC deployment

Trade-off between the number of VNF instances and service requests redirections (path stretch)

Our service chaining problem

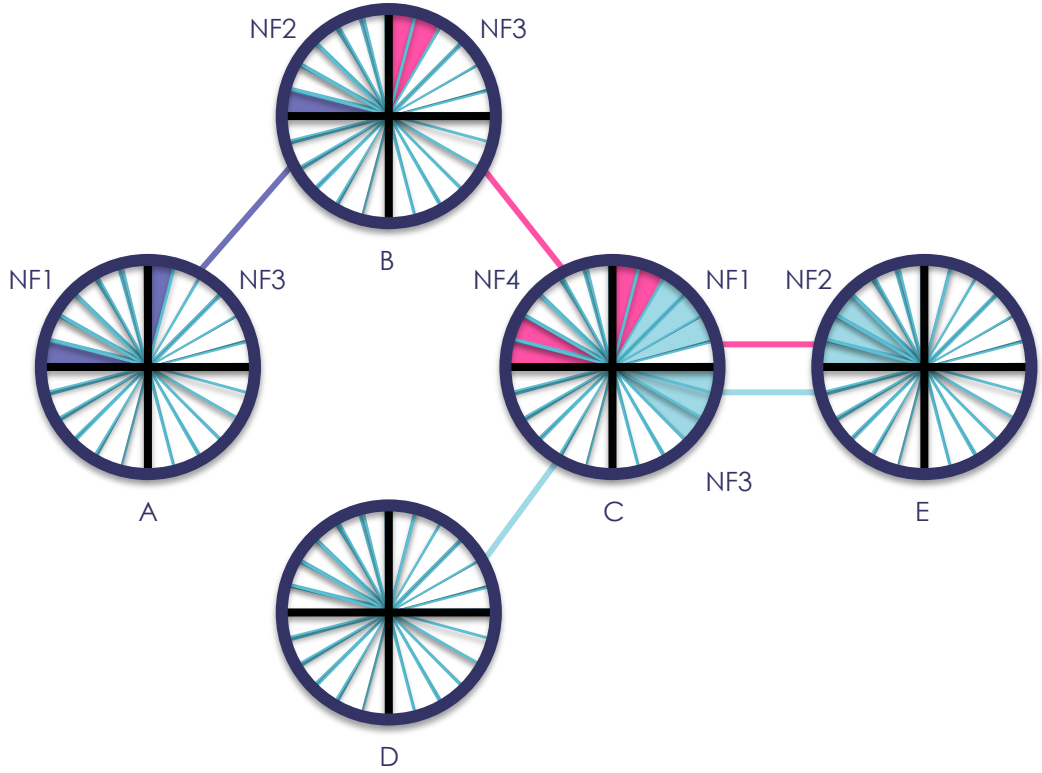
Goal: Minimize total deployment cost

Node cost: if at least one network function is assigned to a node (PoP), this node has a cost c

Network cost: proportional to the number of hops and the CPUs used by the NFs

Hypotheses:

- No SLO penalties
- Link resources are "infinite"
- Only limitation on nodes' (PoP) resources



THALES

ILP formulation

Objective:

$$\min\Big(\sum_{n\in N} o_n \cdot N_c \quad + \sum_{n_1\in N, n_2\in N} a_{n_1,n_2} \cdot \sum_{r\in R, l\in L} y_{r,l,n_1,n_2} \cdot L_c\Big) \tag{1}$$

Subject to:

$$x^c_{r,v_g,n,c} \le x_{r,v_g,n} \qquad \forall n\in N, v_g\in V_g, c\in C, r\in R \tag{2}$$

$$\sum_{r\in R, v_g\in V_g} x^c_{r,v_g,n,c}\cdot s_r \le V_c \qquad \forall n\in N, c\in C \tag{3}$$

$$\sum_{v\in V, c\in C} z_{v,n,c} \le C_c \qquad \forall n\in N \tag{4}$$

$$\sum_{v\in V} z_{v,n,c} \le 1 \qquad \forall n\in N, c\in C \tag{5}$$

$$l > u_r+1 \Rightarrow y_{r,l,n_1,n_2} = 0 \qquad \forall r\in R, n_1\in N, n_2\in N, l\in L \tag{6}$$

$$v_g > u_r \Rightarrow x_{r,v_g,n} = 0 \qquad \forall r\in R, n\in N, v_g\in V_g \tag{7}$$

$$v_g > u_r \Rightarrow x^c_{r,v_g,n,c} = 0 \qquad \forall r\in R, n\in N, v_g\in V_g, c\in C \tag{8}$$

$$v_g \le u_r \Rightarrow x^c_{r,v_g,n,c} \le z_{f_{r,v_g},n,c} \qquad \forall r\in R, n\in N, c\in C, v_g\in V_g \tag{9}$$

$$z_{v,n,c} \le o_n \qquad \forall v\in V, n\in N, c\in C \tag{10}$$

$$v_g \le u_r \Rightarrow \sum_{n\in N, c\in C} x^c_{r,v_g,n,c} = 1 \qquad \forall r\in R, v_g\in V_g \tag{11}$$

$$l^g \le u_r \Rightarrow \sum_{n_2\in N} a_{n_1,n_2}\cdot y_{r,l^g,n_1,n_2} - \sum_{n_2\in N} a_{n_1,n_2}\cdot y_{r,l^g,n_2,n_1} = (x_{r,l^g-1,n_1} - x_{r,l^g,n_1})\cdot s_r \qquad \forall r\in R, n_1\in N, l^g\in L^g \tag{12}$$

$$\sum_{n_2\in N} a_{I_r,n_2}\cdot y_{r,1,I_r,n_2} - \sum_{n_2\in N} a_{I_r,n_2}\cdot y_{r,1,n_2,I_r} = (1 - x_{r,1,I_r})\cdot s_r \qquad \forall r\in R \tag{13}$$

$$n_1 \ne I_r \Rightarrow \sum_{n_2\in N} a_{n_1,n_2}\cdot y_{r,1,n_1,n_2} - \sum_{n_2\in N} a_{n_1,n_2}\cdot y_{r,1,n_2,n_1} = x_{r,1,n_1}\cdot s_r \qquad \forall r\in R, n_1\in N \tag{14}$$

$$\sum_{n_2\in N} a_{E_r,n_2}\cdot y_{r,u_r+1,E_r,n_2} - \sum_{n_2\in N} a_{E_r,n_2}\cdot y_{r,u_r+1,n_2,E_r} = (x_{r,u_r,E_r} - 1)\cdot s_r \qquad \forall r\in R \tag{15}$$

$$n_1 \ne E_r \Rightarrow \sum_{n_2\in N} a_{n_1,n_2}\cdot y_{r,u_r+1,n_1,n_2} - \sum_{n_2\in N} a_{n_1,n_2}\cdot y_{r,u_r+1,n_2,n_1} = x_{r,u_r,n_1}\cdot s_r \qquad \forall r\in R, n_1\in N \tag{16}$$

$$l \le u_r + 1 \Rightarrow a_{n_1,n_2}\cdot y_{r,l,n_1,n_2} \ge 0 \qquad \forall r\in R, n_1\in N, n_2\in N, l\in L \tag{17}$$

Our heuristic algorithm: presentation

3 steps:

- Step 1: Find the minimum number of nodes (*N_min*) needed to allocate all requests. We program a heuristic to do it.

- Step 2: Rank nodes in a decreasing order according to a modified betweenness centrality. Select the *N_min* nodes.

- Step 3: Allocate the SFC requests on the selected *N_min* nodes using an adapted Viterbi algorithm (multi-stage graph).

The last step is the one which dominates in terms of computational time.

Time complexity:

$$\Theta(r)$$

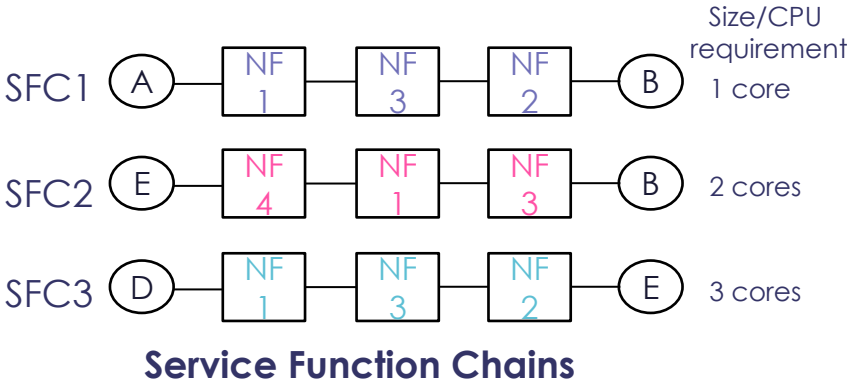
$$\Theta(r \cdot \frac{d+2}{2} \cdot n \cdot log(n))$$

$$\Theta(\frac{r^3}{16}(m-1) \cdot \frac{d+2}{2} \cdot n \cdot log(n))$$

- r number of SFCs
- d node degree
- n number of nodes
- m number of functions per SFC request

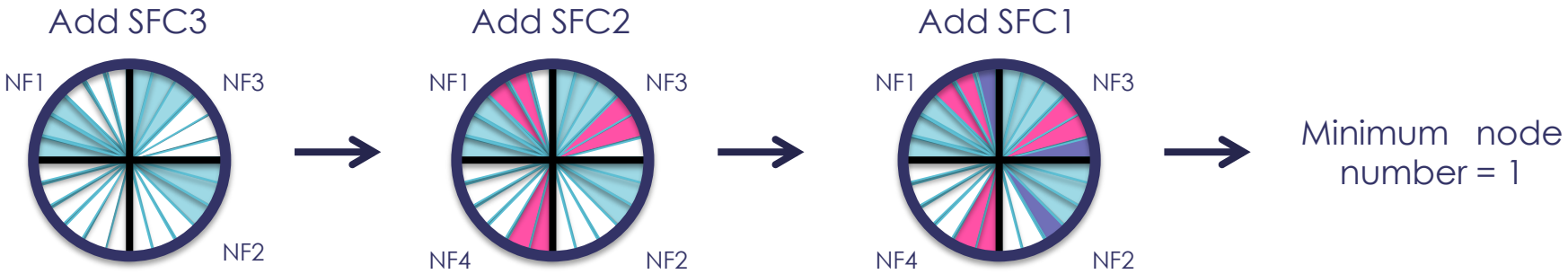
THALES

Step 1: Find the minimum number of nodes



Rank SFC requests by decreasing CPU requirement

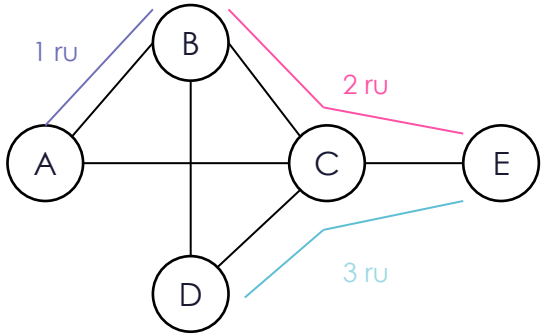
SFC3 > SFC2 > SFC1



THALES

Step 2: Select the nodes based on their modified betweenness centrality

ru = resource unit



NFV Infrastructure

- Compute the betweenness centrality but only with paths from ingress to egress node of requests
- Weigh by CPU requirement

Results:

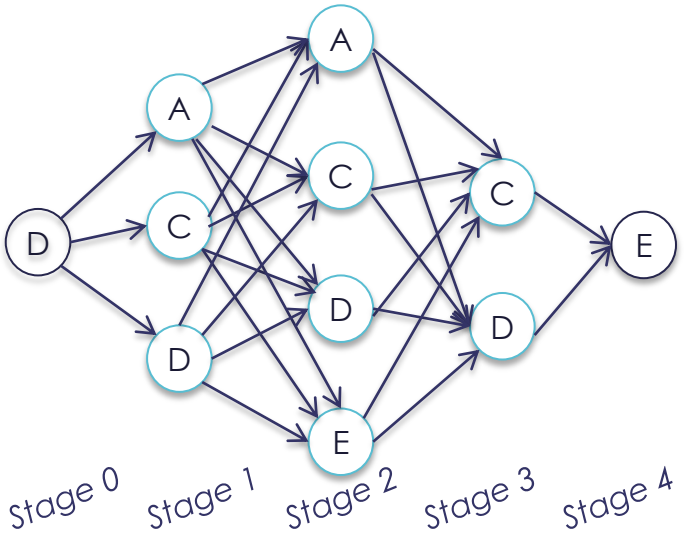
A = 1
B = 3
C = 5
D = 3
E = 5

→

C and E have the highest centrality
As we need only one node, C is chosen for example

THALES

Step 3: Allocate the SFC requests with an adapted Viterbi algorithm



e.g.: minimum node number = 4 and centrality gives A,C,D,E

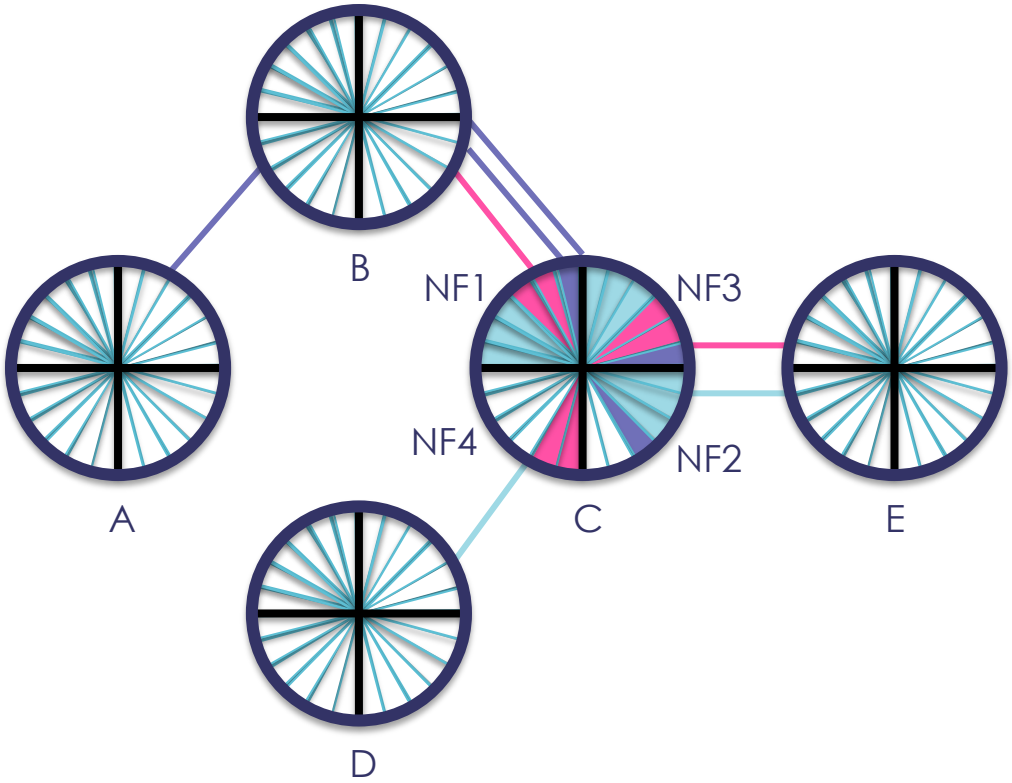
- Directed graph respecting the NF sequence order (SFC request)
- A graph representing for each NF the set of **nodes which have available resources** to host this NF

- A node at "step n-1" is linked to all "step n" nodes
- Each link has a **weight** representing the shortest path length
- The Viterbi Algorithm returns the sequence of nodes (one node by step) minimizing the path between the ingress and the egress nodes.

Our adaptation lies in the set of nodes formation. Our Viterbi graph is dynamic. It could change between two steps. The choice of a node at step n-1 influences the set of possible nodes at step n.

THALES

Example of a final deployment with our heuristic



- Only one node instantiated
- One request redirected: SFC1 has two more hops than the shortest path between A and B

THALES

Simulation

ILP:

- CPLEX Optimization Studio

Heuristic:

- Python 3

Random graph model:

- Erdos Renyi model with degree 3

SFC requests:

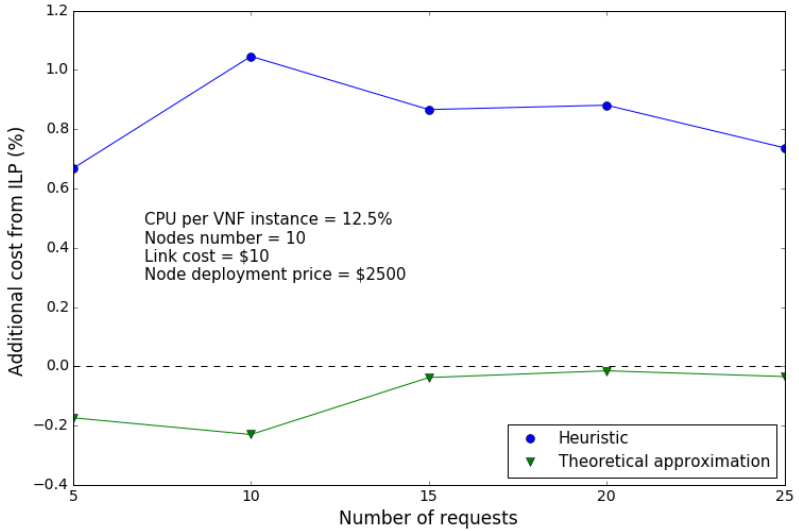
- CPU requirement at random, length of SFC request = 3, Number of NF types = 4

30 iterations

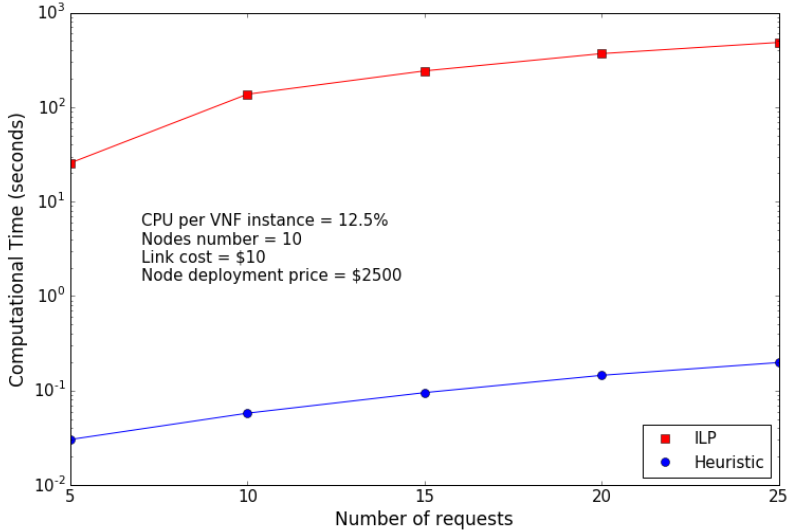
« Theoretical approximation »:

- Lowest bound: *N_min* nodes, path length = shortest path
- Replace ILP for big instances

Heuristic vs. ILP (vs. Theoretical approximation)



Heuristic cost close to the ILP solutions

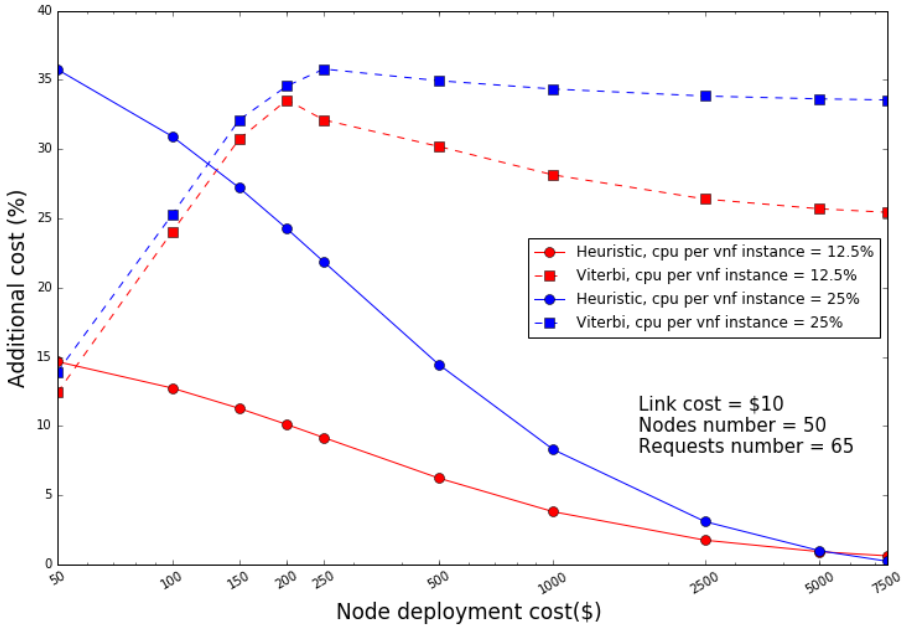


Factor 1,000 between the heuristic and the ILP computation time

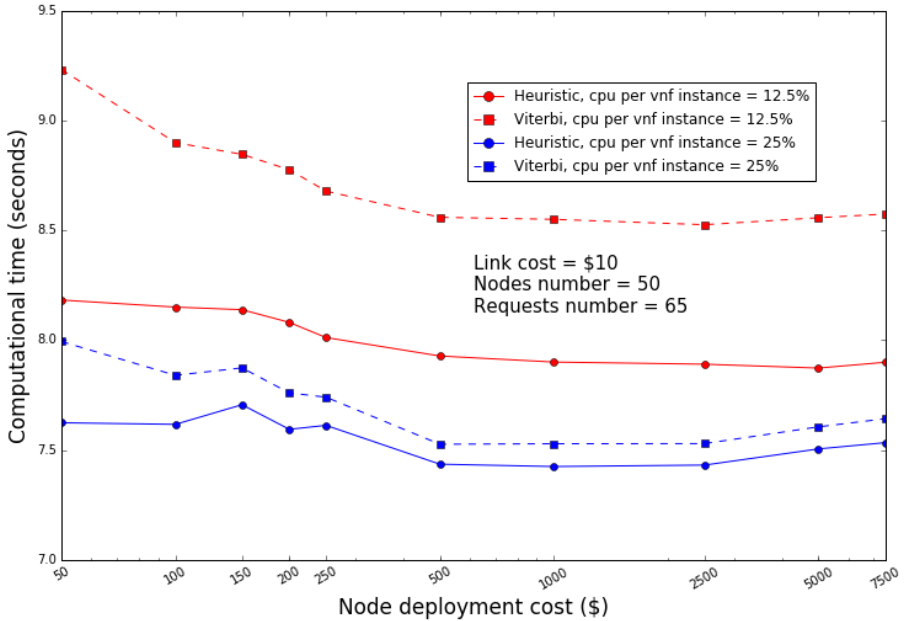
For large instances: the ILP will be replaced by the Theoretical approximation for biggest instances

THALES

Heuristic vs. state of art “Viterbi-based” heuristic [1]



Heuristic very efficient for node cost >> link cost



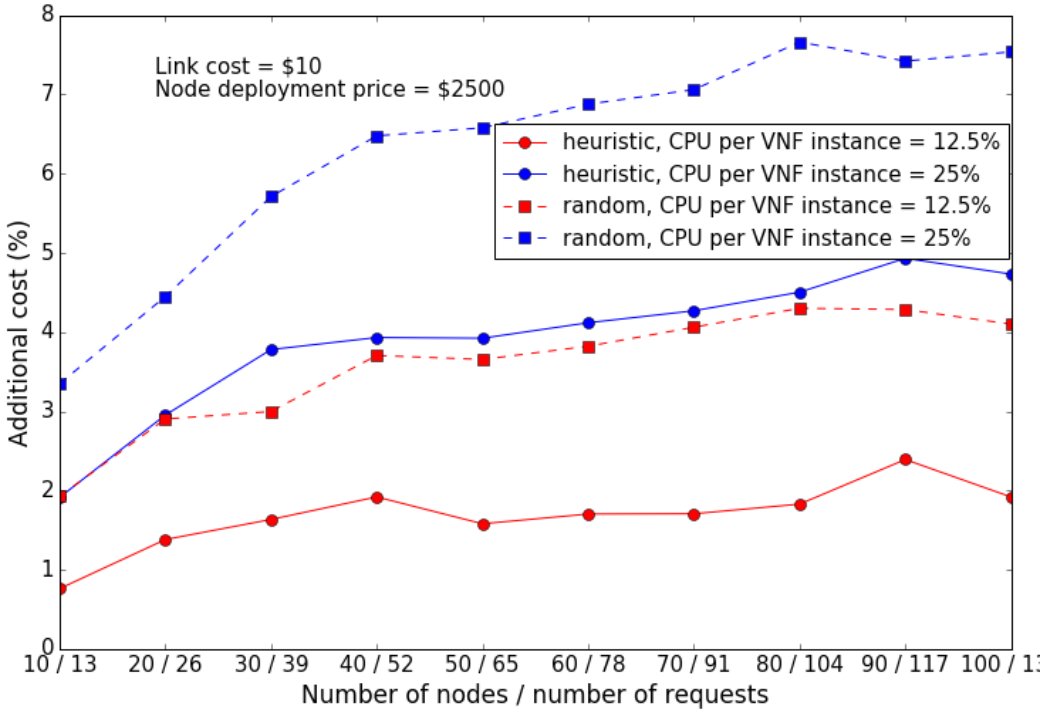
Low computational time

Blue = 4 cores / server
Red = 6 cores / server

[1] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, “On orchestrating virtual network functions,” in CNSM 2015.

THALES

Comparison Heuristic with centrality vs. Heuristic with random



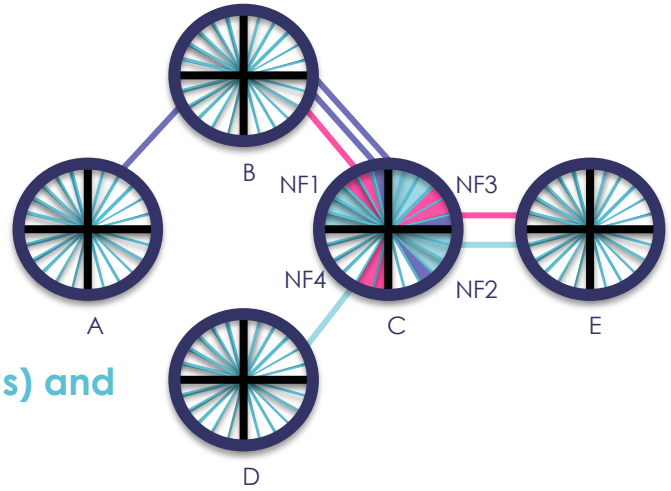
Additional cost for different instance size for the heuristic - 2 strategies for node selection: Centrality vs. Random

- Centrality is worth to use (solid lines under dashed lines)
- "Random" corresponds to a random choice for the set of the min number nodes to instantiate

THALES

Conclusion

Goal: Find a cost-effective placement for the virtual network function chaining problem



Heuristic in 3 steps:

- Find min number of nodes to instantiate
- Centrality-based election of nodes
- Multi-stage graph allocation

Our heuristic minimizes the number of used nodes (PoPs) and reroutings

Our heuristic is efficient, particularly for a node price larger than the flow unit price

Future work:

- Add SLO penalties
- Online problem
- Tackle the reliability problem

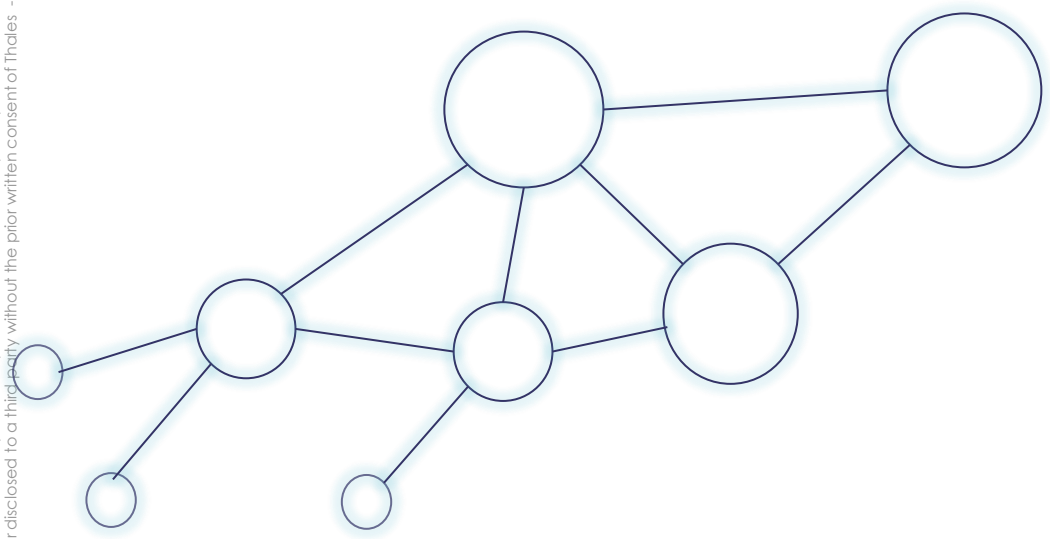
THALES

On-line placement and chaining

THALES

Online NFV Orchestration – Problem Statement

Online Orchestration



VNF located in PoPs

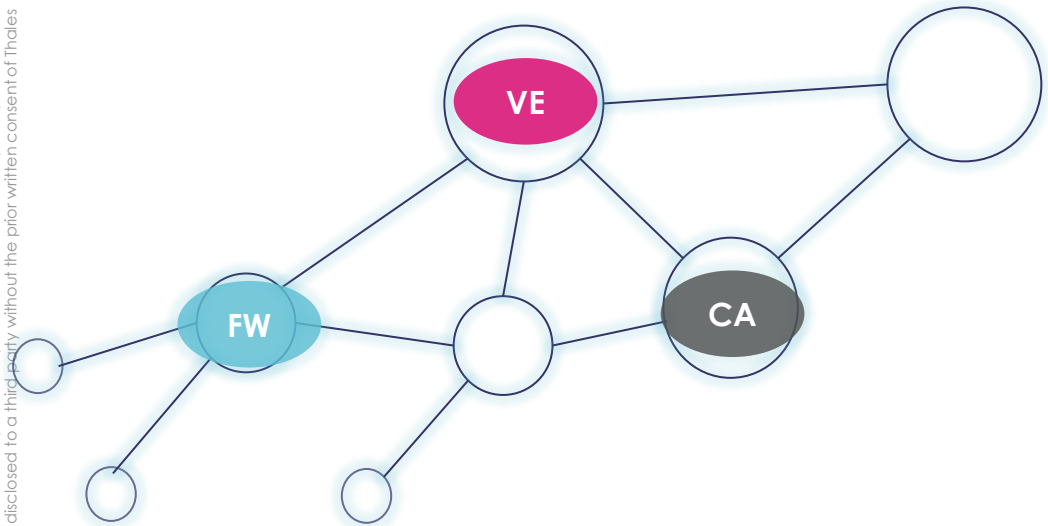
- FW — Firewall
- DPI — Deep Packet Inspection
- CA — Cache
- VE — Video Encoder
- IDS — Intrusion Detection System

THALES

Online NFV Orchestration - Problem Statement

Online Orchestration

- VNF location



VNF located in PoPs

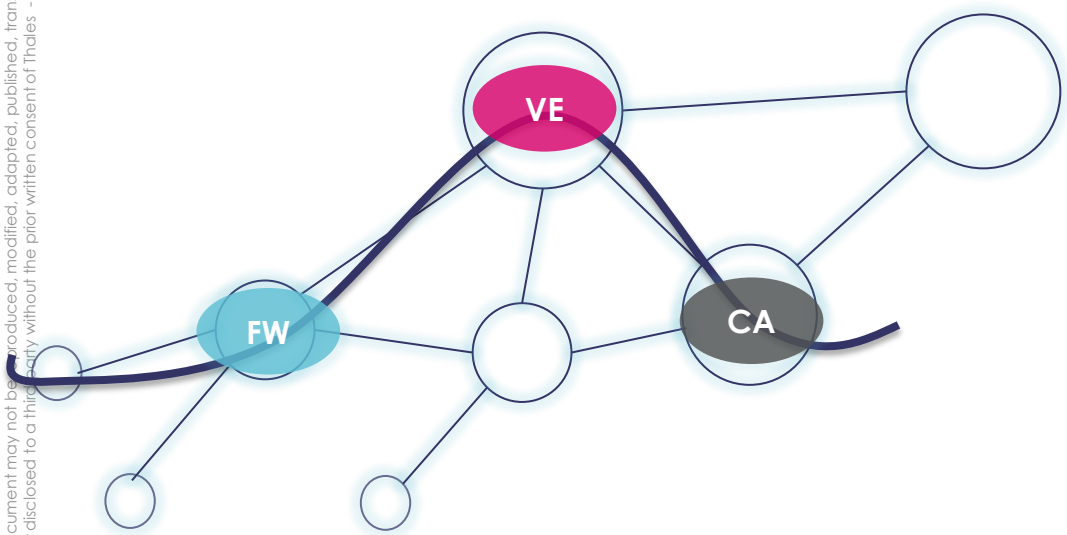
- FW — Firewall
- DPI — Deep Packet Inspection
- CA — Cache
- VE — Video Encoder
- IDS — Intrusion Detection System

THALES

Online NFV Orchestration - Problem Statement

Online Orchestration

- VNF location
- Routing

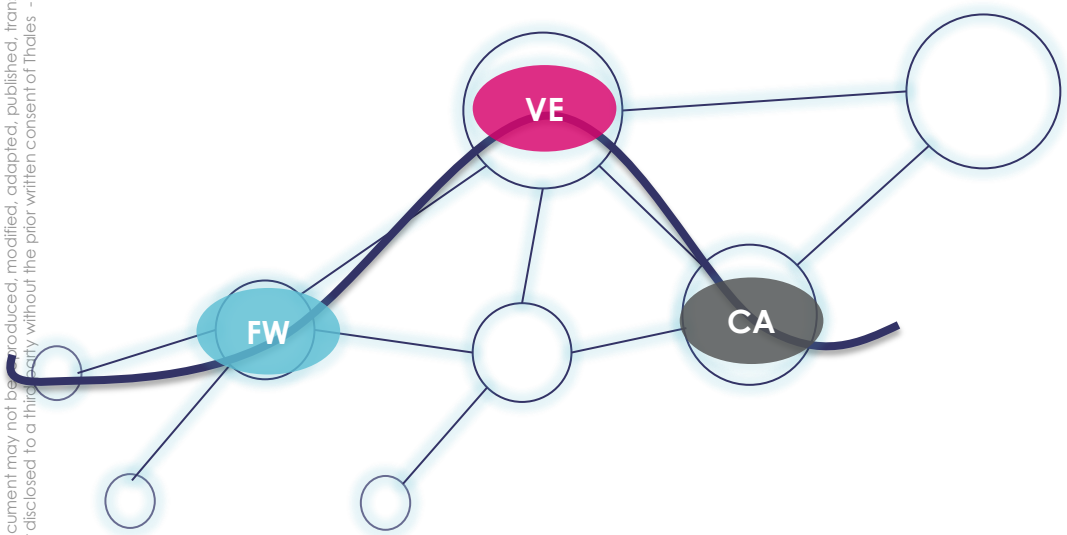


THALES

Online NFV Orchestration - Problem Statement

Online Orchestration

- VNF location
- Routing



Joint problem

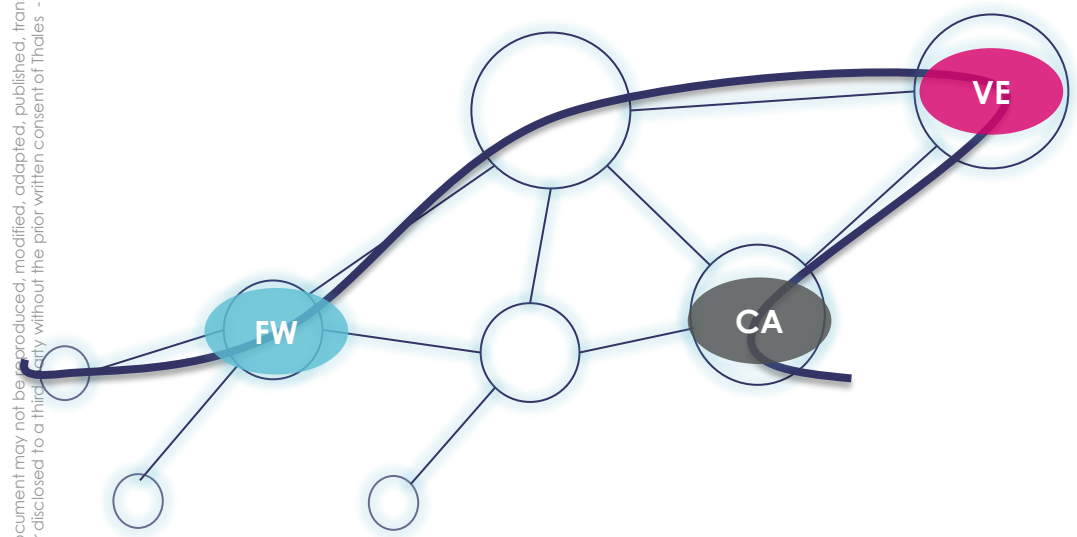
- VNF placement constrains possible routing
- Optimal placement depends on load
- Load depends on routing

THALES

Online NFV Orchestration - Problem Statement

Online Orchestration

- VNF location
- Routing



Joint problem

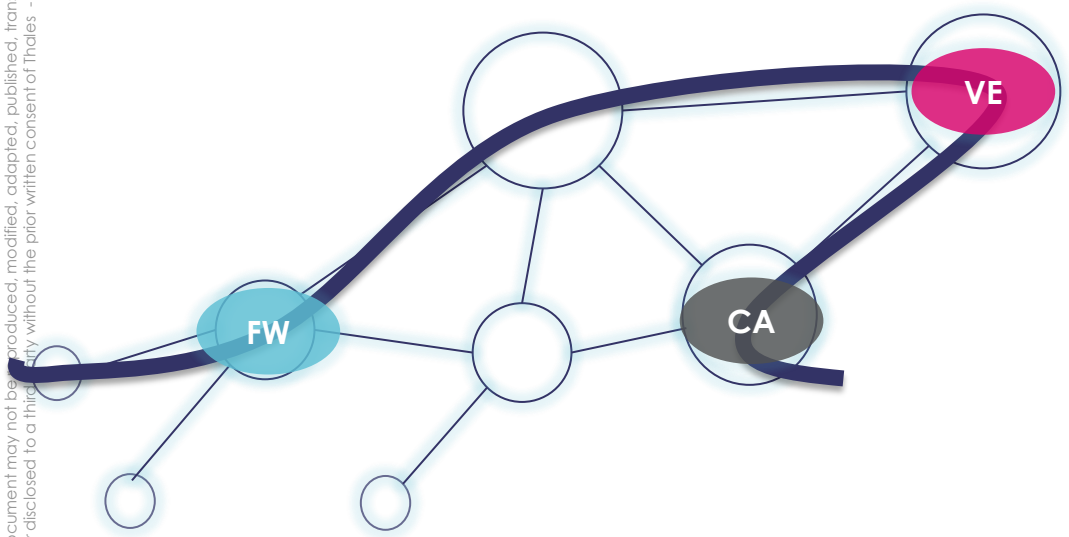
- VNF placement constrains possible routing
- Optimal placement depends on load
- Load depends on routing

THALES

Online NFV Orchestration - Problem Statement

Online Orchestration

- VNF location
- Routing



Joint problem

- VNF placement constrains possible routing
- Optimal placement depends on load
- Load depends on routing

Online problem:

- New flows
- Changing loads
- Optimal placement changes

THALES

Online NFV Orchestration - Problem Statement

- Competing goals:
  - Minimize network cost
  - Keep number of rerouted flows low

- Solved with:
  - ILP: gives optimal results on small topologies in terms of cost and number of rerouted flows
  - Game theory algorithm:
    - Gives a theoretical bound of how far the cost is from the optimal
    - Is quicker than the ILP
    - Can be tuned to favor either a low cost or a low number of rerouted flows
    - Can be distributed

COST MODEL

C=Cl+Cp+Co

Cl: Links

- Linear Cost on each link w.r.t to bandwidth

Cp: Computing

- Piece-wise linear w.r.t load on the PoP
  - Cost grows faster as load grows
  - Energy
  - Limited PoP capacity

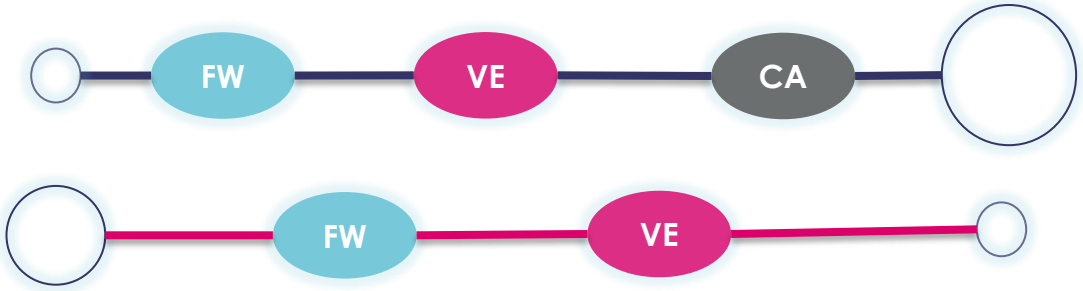
Co: Opening Cost

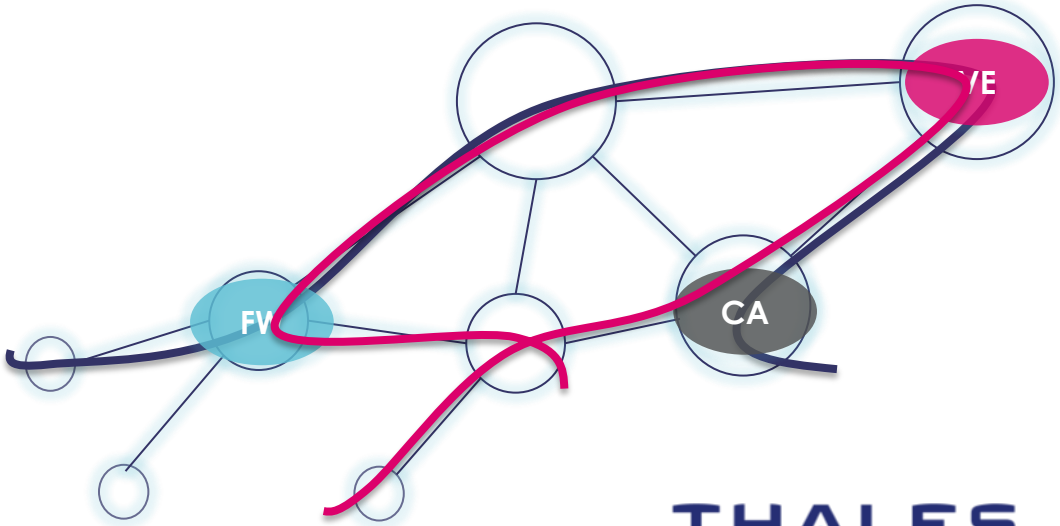
- Constant on each VNF

Online NFV Orchestration – Game theory algorithm

Game theory heuristic

- Each Service chain is a player
- Each player tries to minimize their cost:
  - link cost
  - computing cost
  - opening cost
- Cost paid by each request is the proportion of each global cost



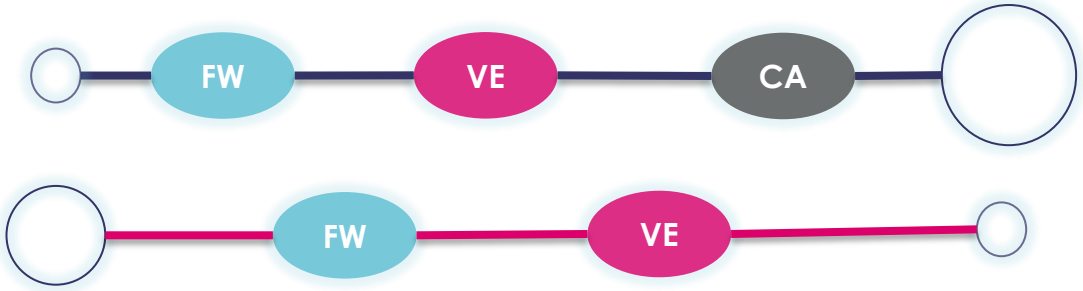


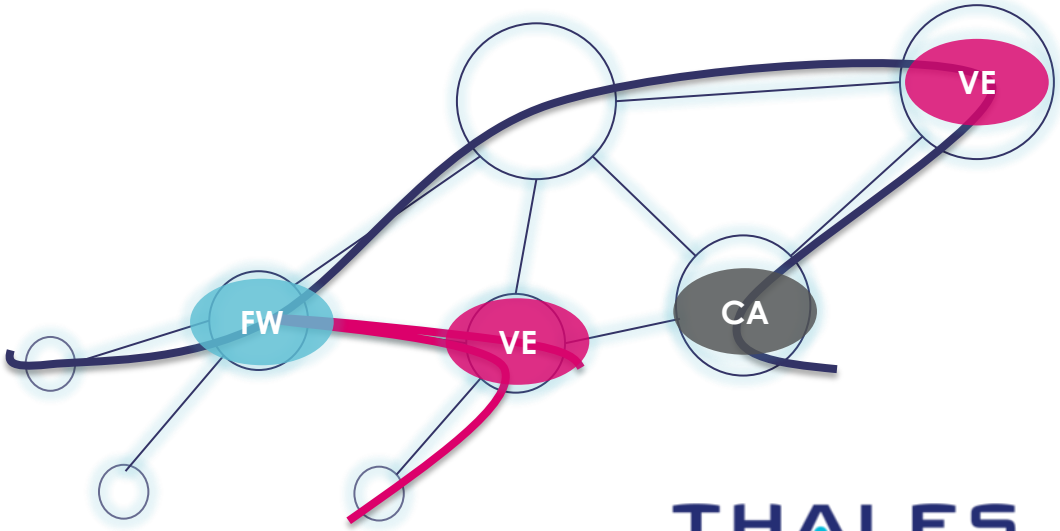
THALES

Online NFV Orchestration – Game theory algorithm

Game theory heuristic

- Each Service chain is a player
- Each player tries to minimize her cost:
  - link cost
  - computing cost
  - opening cost
- Cost paid by each request is the proportion of each global cost



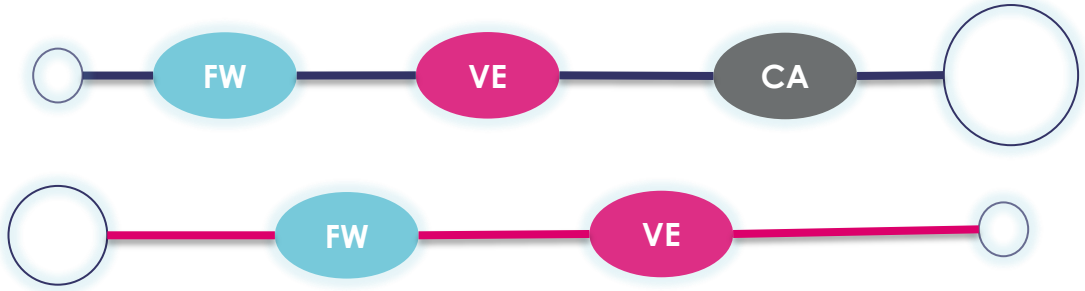


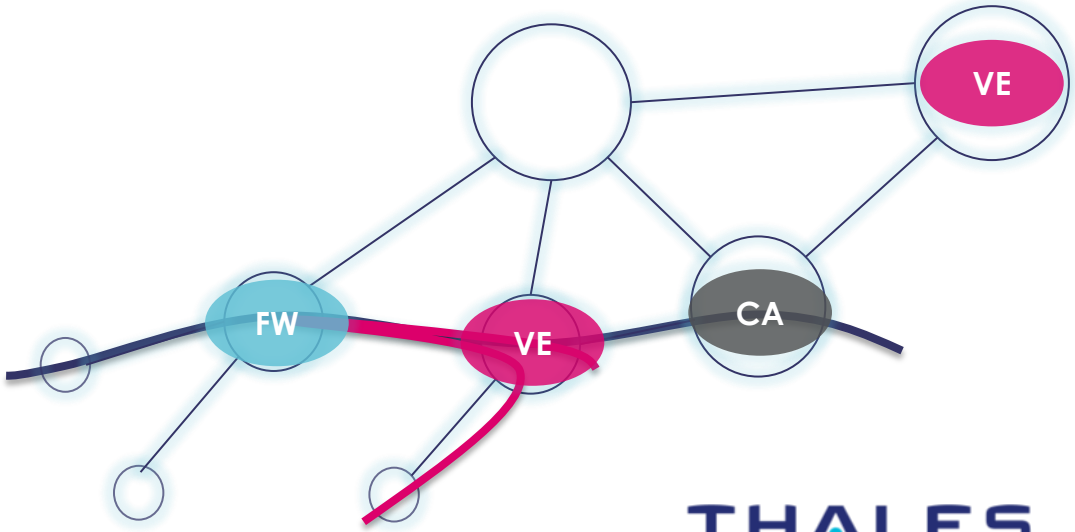
THALES

Online NFV Orchestration – Game theory algorithm

Game theory heuristic

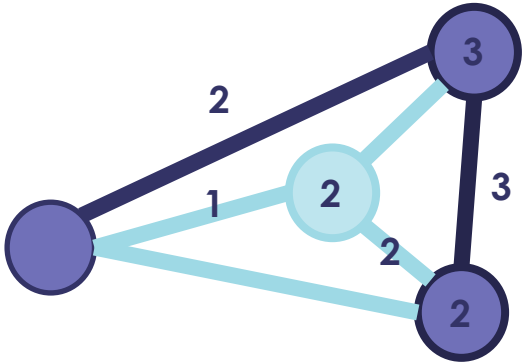
- Each Service chain is a player
- Each player tries to minimize her cost:
  - link cost
  - computing cost
  - opening cost
- Cost paid by each request is the proportion of each global cost
- Each player's decision can be made independently
- Cost depends on other player placement



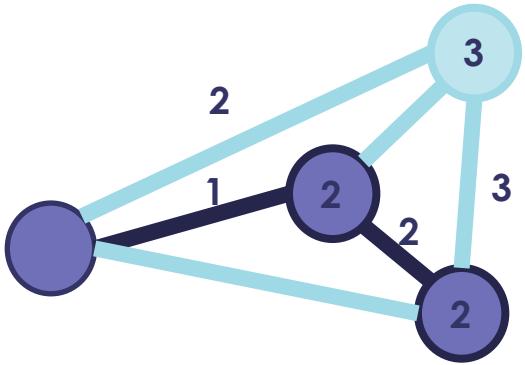


THALES

Online NFV Orchestration – Game theory algorithm



Cost: 10



Cost: 7

Tuning reroutings with parameter α

- α = 0 no rerouting, we keep the first offline solution
- α =1 every « interesting rerouting »

α close to 0		α closer to 1
Higher cost		Lower cost
Less reroutings	Trade-off	more reroutings
Shorter convergence time		longer convergence time

THALES

Online NFV Orchestration – Evaluation

Topology

- The measured intensity of traffic is equal to request size
- Traffic Matrix: All to All
- Real traffic matrix

Uhlig et al. Providing public intradomain traffic matrices to the research community. ACM SIGCOMM Computer Communication Review, 36(1):83–86, 2006.

- Request are chains of either 3 or 5 VNFs

THALES

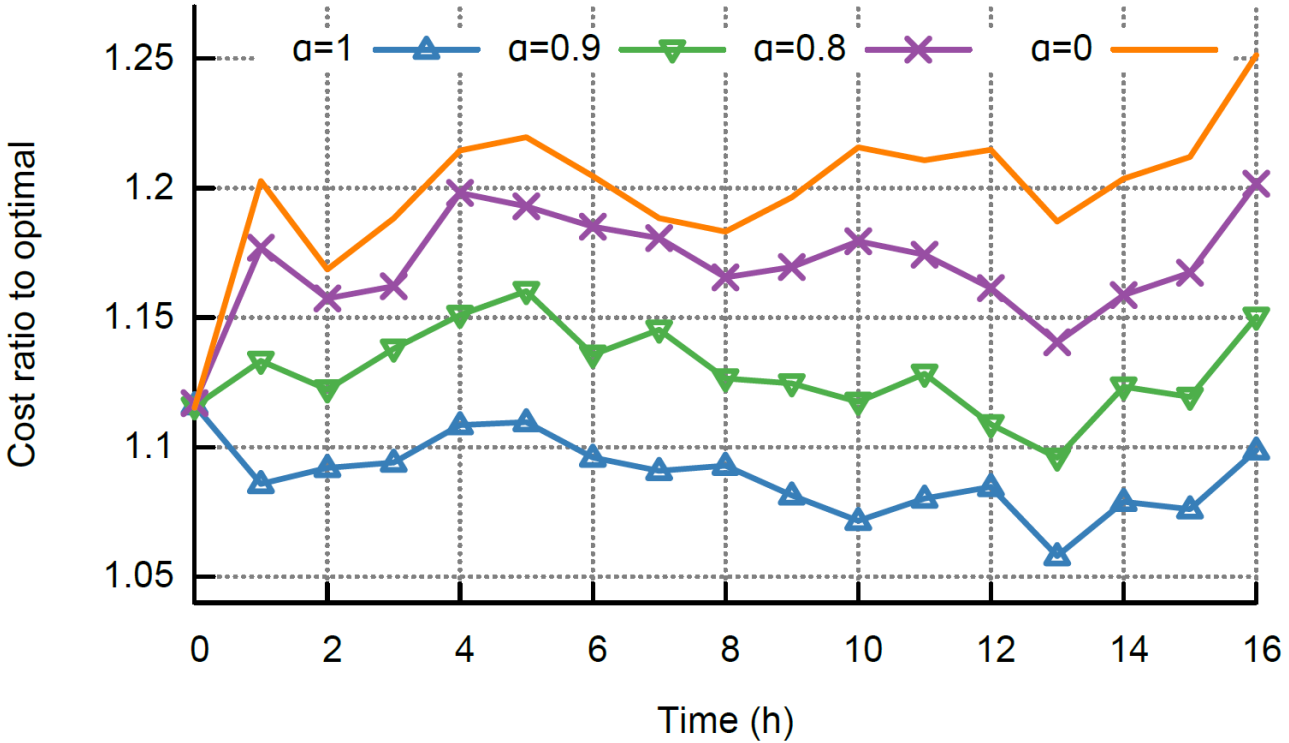
Online NFV Orchestration – Evaluation, comparison with optimal cost

Cost comparison

PoP capacity:
total traffic/Nb of PoPs

Computing cost:
Under cap=$1
Over cap= $5
Link cost=$1
Opening cost=$20

- **Alpha can be tuned to give close to optimal cost if it is equal to 1**
- **Lower values of alpha give results farther from the optimal**



THALES

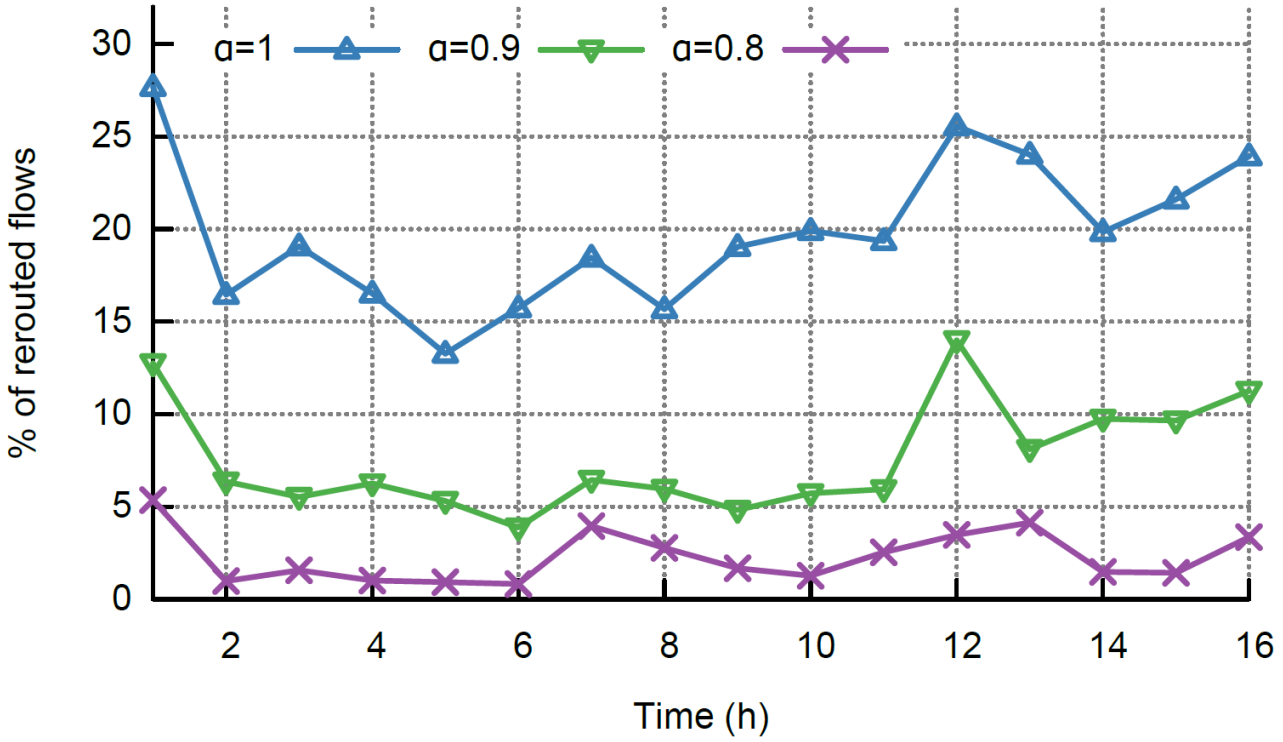
Online NFV Orchestration – Evaluation

Rerouted flows

PoP capacity:
total traffic/Nb of PoPs

Computing cost:
Under cap=$1
Over cap= $5
Link cost=$1
Opening cost=$20

- **When alpha is equal to 1 the number of rerouted flows can be high**
- **It is much lower for lower values of alpha**



THALES

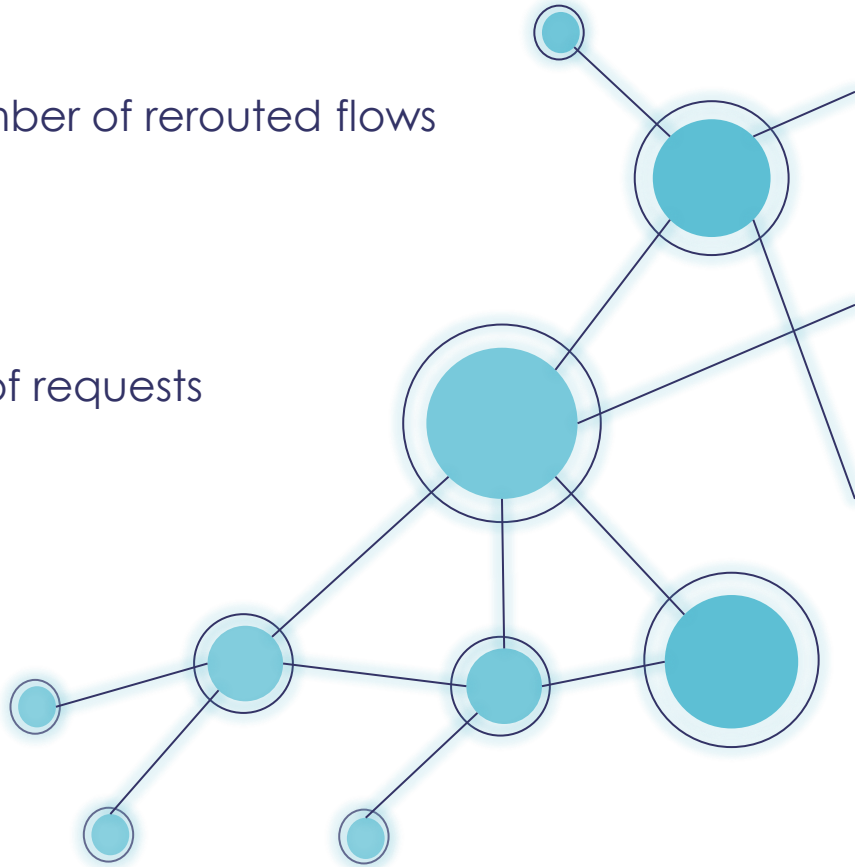
Online NFV Orchestration – Take away

ILP

- Gives optimal placement and optimal number of rerouted flows
- Works on small topologies

Heuristic

- Works on big topologies and high number of requests
  - Tested on data center and WAN topology
- Can be tuned to be either
  - Closer to optimal but with more rerouting
  - Quicker with a low number of rerouting
- Can be distributed



THALES

Conclusions

NFV is part of a major shift in networking

TODAY	TOMORROW
IP networks	Software Defined Networks
Static provisionning of topology and SLAs	Dynamic control of network resources
Tight mapping of services onto resources	Unbundling of resource & service layers
Low interoperability (basic IP support)	Rich set of API based interoperability
Dedicated infrastructure	On-demand resource & services
Silo systems	Network data analytics

THALES

NFV is about IT and network convergence

A network is made of software

- Software methodology applies: build, run, manage

A network is more than software

- Requires orchestration of VNFs (placement and chaining)

THALES

Thanks

Paper references

- Online NFV orchestration with game theory, RESCOM 2017,
  - Mathis OBADIA, Mathieu Bouet, Vania Conan, Luigi Iannone, Jean-Louis Rougier
- A graph approach to placement of service functions chains
  - Nicolas Tastevin, Mathis Obadia, Mathieu Bouet
  - IFIP/IEEE IM 2017

THALES