Service-aware Dynamic NFV Resource Allocation

Speaker: Tao Gao

2018-03-23

Group Meeting Presentation

What is VNF?

- High CAPEX and OPEX when deploying and updating network infrastructure, due to the need of specified network hardware (middle-boxes).

- Middle-box suffers several shortcomings:
  i. they are expensive
  ii. require specialized managing personnel
  iii. have high energy costs
  iv. do not allow to add new functionality
  v. have short lifecycles

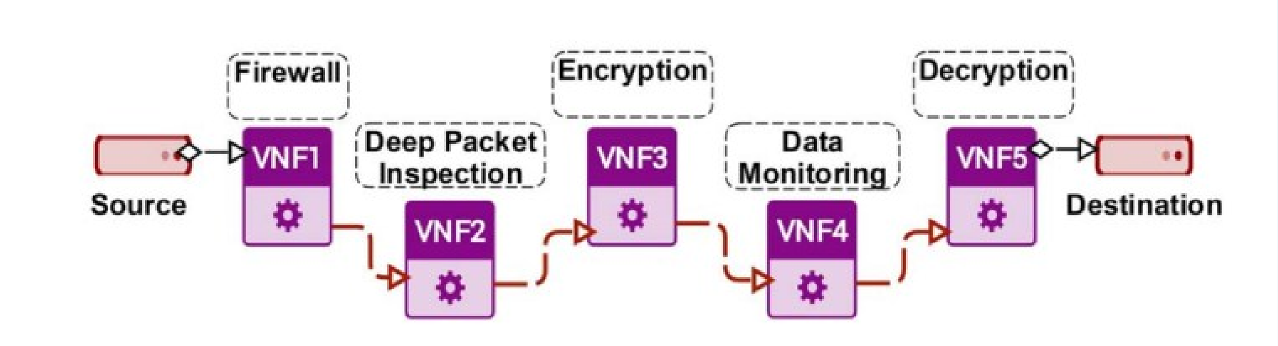
- The number of middleboxes is comparable with the number of routers and switches needed to maintain the operation of the network.

What is VNF?

- Traditional middleboxes are managed as single modules of software, programmed to play the role of a particular Virtual Network Function (VNF), this allows modularity and isolation of each function, so they can be managed independently.

- NFV facilitates installation and deployment of VNFs on general purpose servers thus allowing dynamic migration of VNFs from one server to another, that is, to any place of the network.

- In the NFV ecosystem, an Network Service (NS) is a set of chained VNFs.



Juliver Gil Herrera and Juan Felipe Botero, "Resource Allocation in NFV: A Comprehensive Survey," IEEE Trans. On Network And Service Management, vol. 13, no. 3, pp. 518-532, 2016

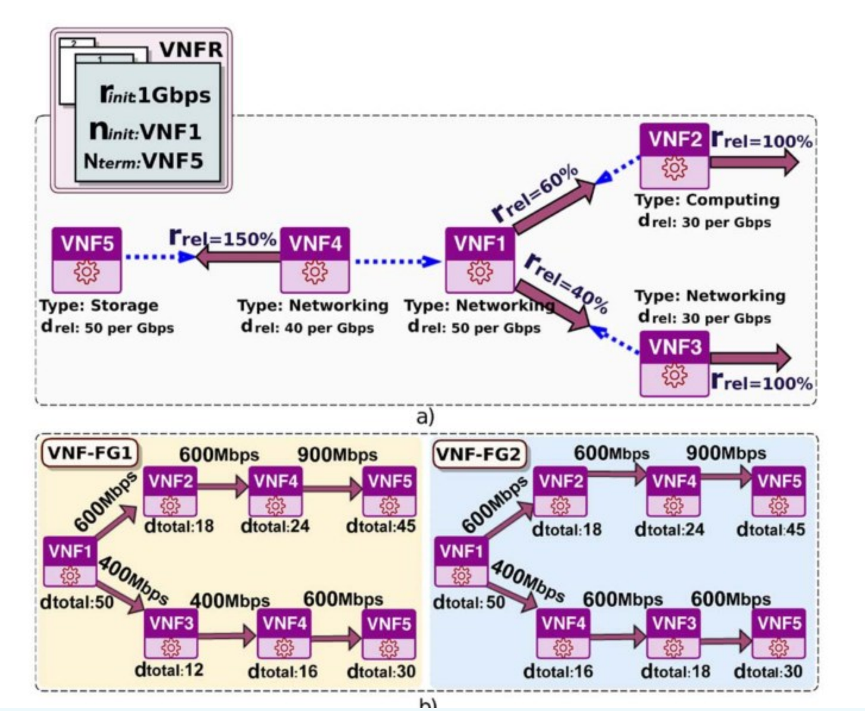
Problems and Challenges

- An NS is built and deployed in NFV by defining its:
  i) number of VNFs
  ii) their respective order in the chain
  iii) the allocation of the chain in the Network Functions Virtualization Infrastructure (NFVI), also called Substrate Network (SN)

- NFV-RA includes three stages:
  i) VNFs - Chain Composition (VNFs-CC)
  ii) VNF - Forwarding Graph Embedding (VNF-FGE):
  iii) VNFs - Scheduling (VNFs-SCH):

Problems and Challenges

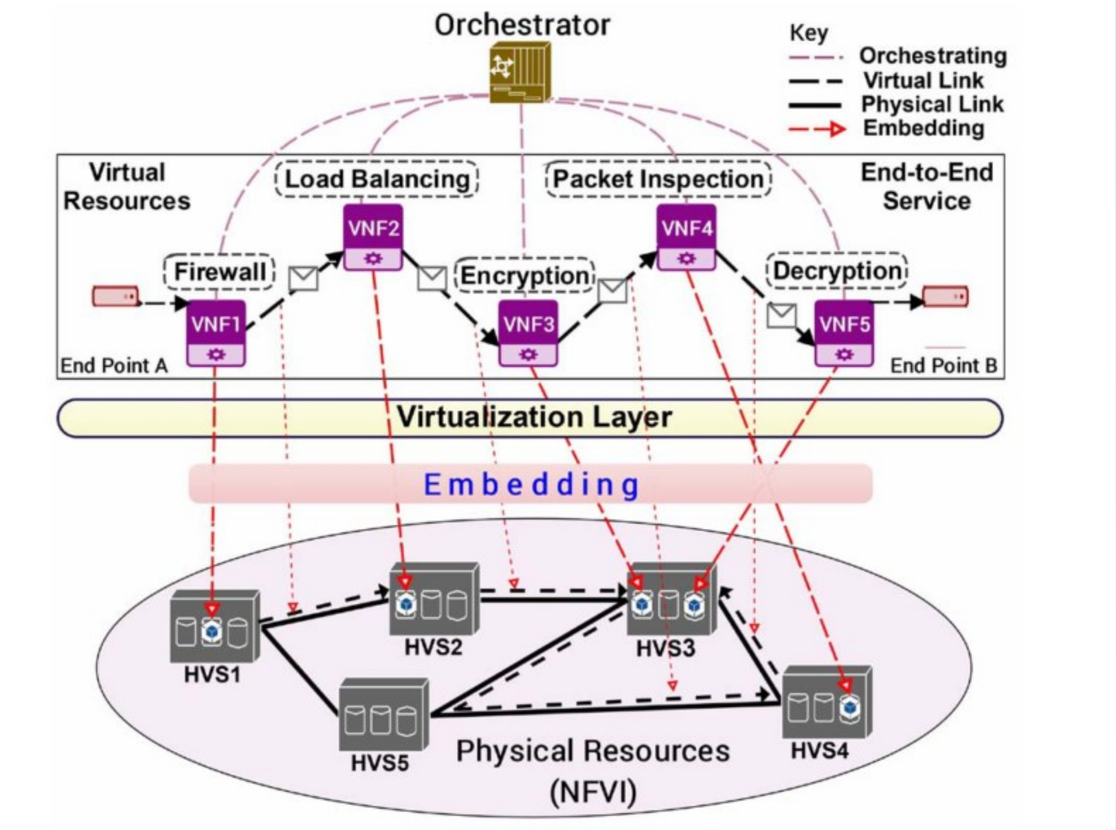
VNFs - Chain Composition (VNFs-CC): How to concatenate the different VNFs efficiently in order to compose an NS in the most adequate way, with respect to the TSP goals?



Juliver Gil Herrera and Juan Felipe Botero, "Resource Allocation in NFV: A Comprehensive Survey," IEEE Trans. On Network And Service Management, vol. 13, no. 3, pp. 518-532, 2016

Problems and Challenges

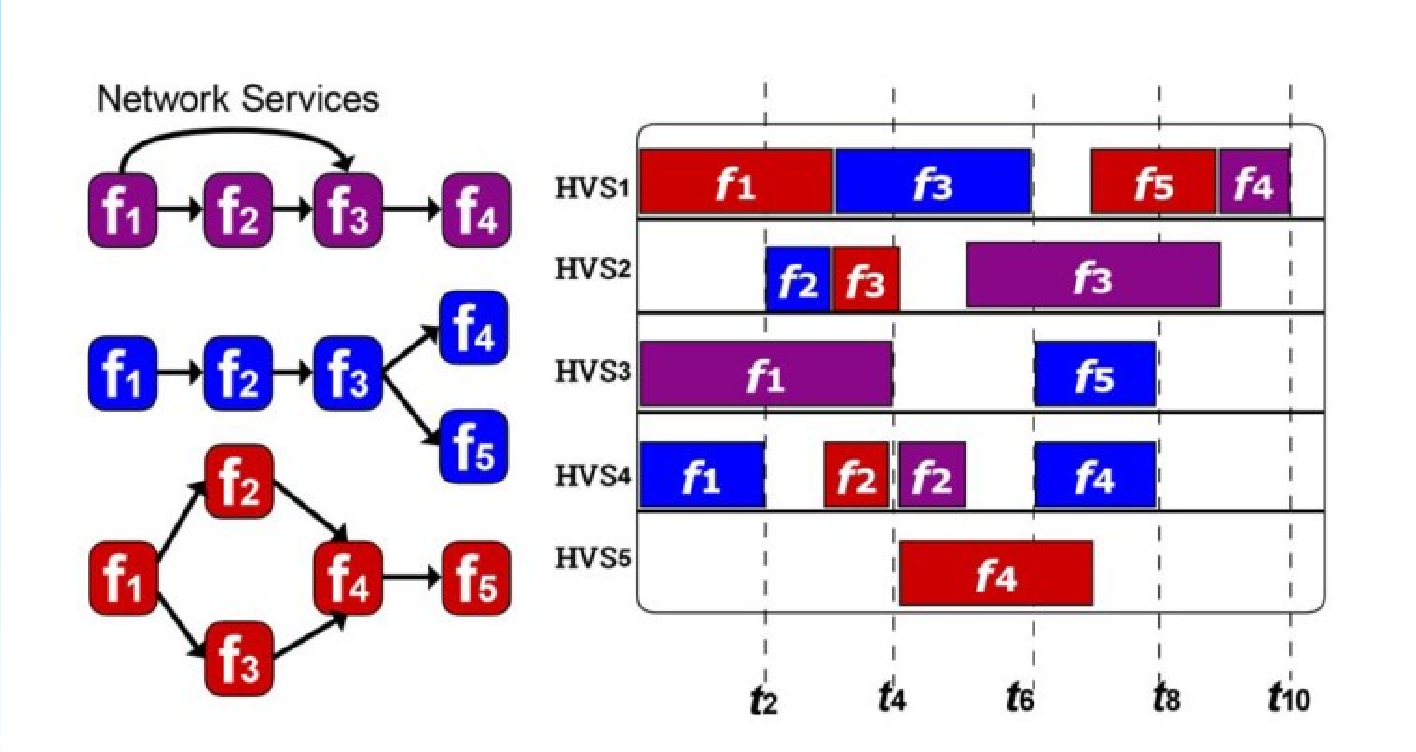
VNF - Forwarding Graph Embedding (VNF-FGE): Seeks to find where to allocate the VNFs in the network infrastructure in a suitable way, considering a set of requested network services.



UCDAVIS NETWORKS RESEARCH LAB

Problems and Challenges

VNFs - Scheduling (VNFs-SCH): A proper scheduling of VNFs' execution should be performed in order to minimize the total execution time of the network services, and thus obtain improved performance.



Research Directions

- In the VNF-FGE stage, the virtual network function can be allocated into a VM, or a container in a VM. For a small-size function, a VM is a waste.
- Current research are mainly focused on the static traffic scenario.
- **VNF can be used by many network services. A instance should be reused due to the cost of installation and uninstallation of a VNF. The operation and management of virtual network function should be service-aware.**
- There is few research about survivable VNF-RA

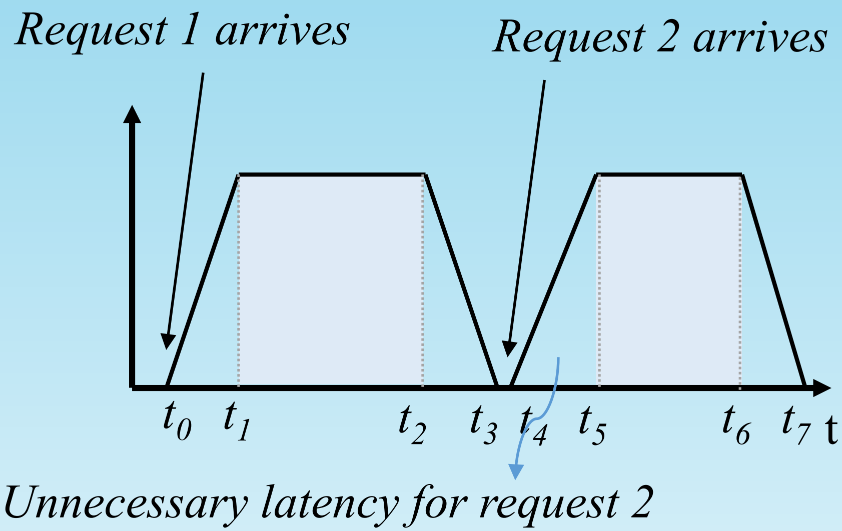
Shortcomings of Current Research

- There are few research considering the VNFs-FGE step and VNFs-Scheduling step together.

- Current research usually focus on the static traffic, however, the traffic in networks are dynamic. Especially for the scheduling problem, considering the dynamic traffic scenario is necessary.

- No research consider the VNF itself. E.g. installing or removing an instance of a VNF will consume some time.

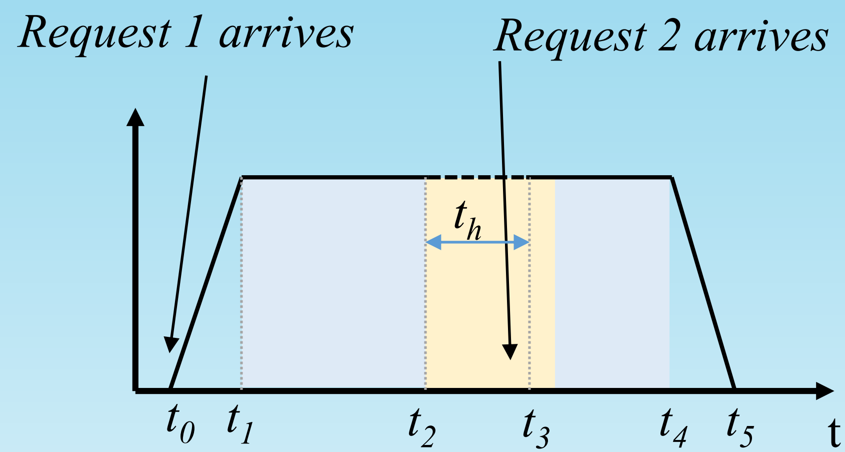
Service-aware VNF Scheduling



$t_0 \sim t_1$ & $t_4 \sim t_5$: VNF setting up

$t_2 \sim t_3$ & $t_6 \sim t_7$: VNF removing

$t_1 \sim t_2$ & $t_5 \sim t_6$: VNF working



$t_0 \sim t_1$: VNF setting up
$t_2 \sim t_3$: VNF holding
$t_4 \sim t_5$: VNF removing

$t_1 \sim t_2$ & $t_3 \sim t_4$: VNF working

t_h: The VNF holding time

Problem Statement

For a service request:

- ✓ **Data rate**: should be assigned to virtual links.
- ✓ **VNF orde**r: some VNFs have specified relationship.
- ✓ **Deadline of execution time**: all VNFs should be finished in a certain period.

For a VNF:

- ✓ **Resource usage**: the VM hosting the instance of the VNF consumes CPU, memory, storage, and other kinds of resource.
- ✓ **Establishment time**: t_{init}, the time consumed when an instance of the VNF is set up in a VM of a node.
- ✓ **Removing time**: t_{rem}, the time consumed when an instance of the VNF is removed from a VM of a node.
- ✓ **Idling cost**: c_{idle}, the cost per time unit caused by an instance when it is idle.
- ✓ **Time parameters**: minimum holding time t_{min}, maximum holding time t_{max}.
- ✓ **Processing capacity**: the capacity processing the traffic data.

UCDAVIS NETWORKS RESEARCH LAB

Problem Statement

For a network:

- ✓ **VNF-capable nodes**: a node can be deployed a VM, such a node has a limited resource of CPU, memory, and storage.
- ✓ **Physical link capacity**: physical fiber links have some bandwidth resource, which can be allocated to several virtual links.
- ✓ **Buffer**: a VNF-capable node has a buffer, which maintain a queue that want to use the instance of the VNF. The buffer has a limited size.

Overall network cost:

$$C_{net} = \sum_i C_{ins}^i$$

i^{th} instance cost:

$$C_i^{ins} = \alpha \cdot (t_{init}^i + t_{rem}^i) + t_{hold}^i \cdot c_{idle}$$

Serving time:

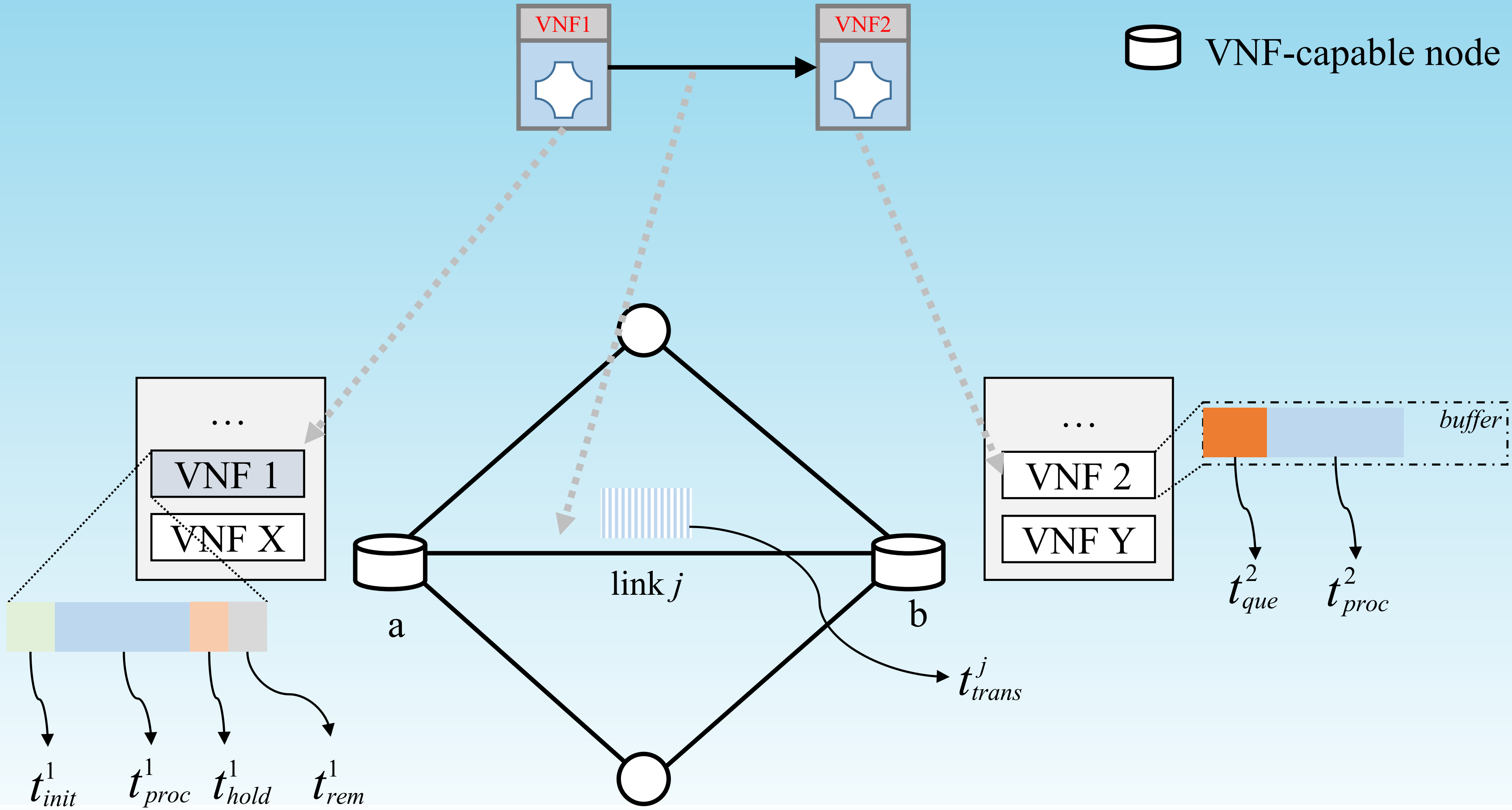
$$t_{serv} = \sum_n (t_{que}^n + t_{init}^n + t_{proc}^n) + \sum_j t_{trans}^j$$

$$t_{proc}^n = \frac{w_s}{p_n} \quad [1]$$

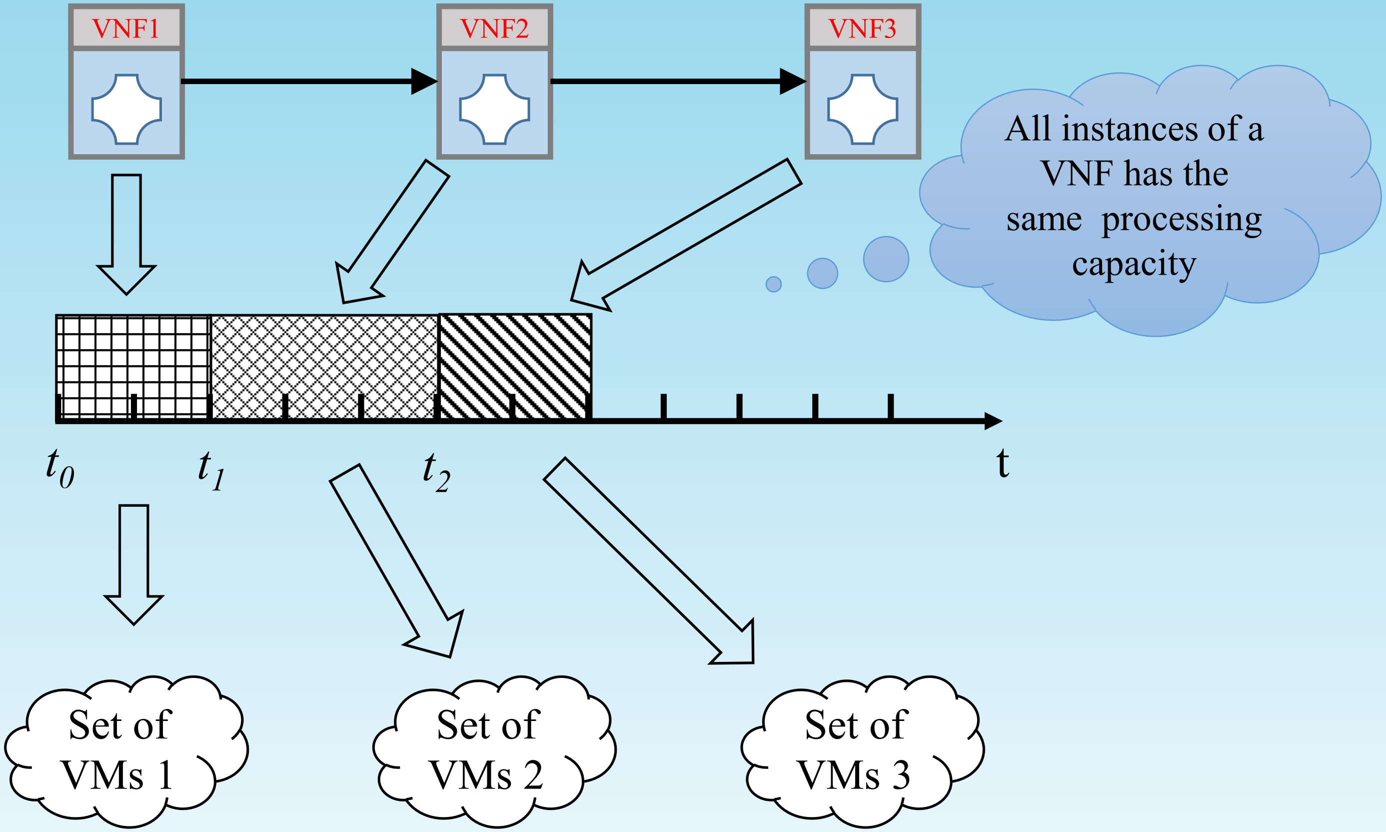
$$t_{trans}^j = \frac{w_s}{b_j} \quad [1]$$

[1] H. Alameddine, "Scheduling Service Function Chains for Ultra-Low Latency Network Services," in Proc. Network and Service Management (CNSM), 2017

Problem Statement



VNF-FGE & Scheduling



Assume there is no transmission latency

Check the available nodes at certain time slot whose VMs host the required VNF

UCDAVIS NETWORKS RESEARCH LAB

VNF-FGE & Scheduling

Check whether there exist a node hosting all required VNFs

Set of VMs 1 ∩ Set of VMs 2 ∩ Set of VMs 3 = ?

Yes → Schedule Finish

No ↓

For any two adjacent sets from the first:

Set i ∩ Set i+1 = ?

Not empty → Repeat this procedure with next Set

Empty ↓

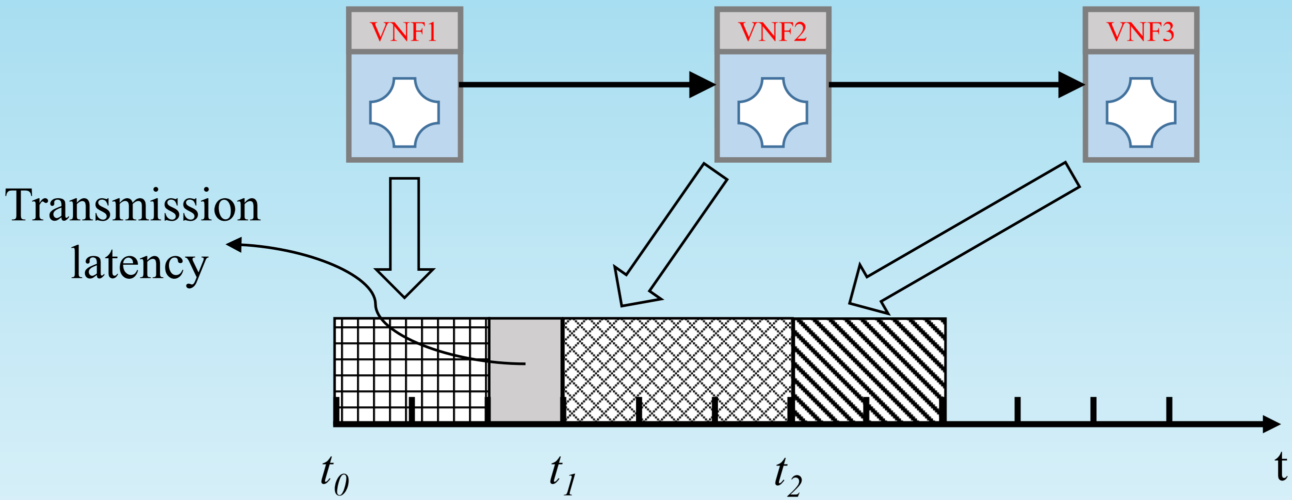
1. If both sets are not empty but have no common nodes
2. If Set i+1 is empty but Set i is not
3. If Set i is empty but Set i+1 is not
4. If both sets are empty

UCDAVIS NETWORKS RESEARCH LAB

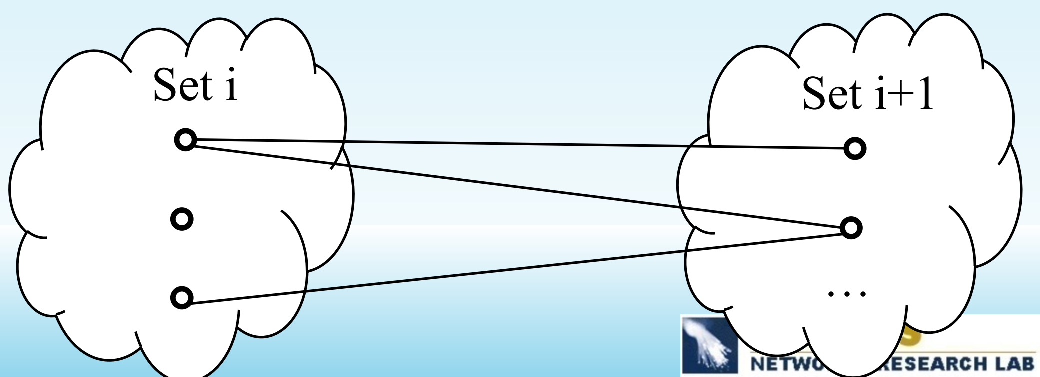
VNF-FGE & Scheduling

1. If both sets are not empty but have no common node

Sub-step 1. Re-check available nodes considering transmission latency



Sub-step 2. Construct auxiliary graph and select nodes



If there is established connections, selects the one with shortest path in subtract network

If not, select the nodes whose load is lightest and establish a connection

VNF-FGE & Scheduling

2. If Set i+1 is empty but Set i is not or

Select the node (***N***) with lightest load in the set, check whether there is enough capacity to install the instance of VNF i+1

⇩ If there is

Set up an instance of VNF i+1 in node ***N***

⇩ If not

Check the available nodes hosting VNF i+1 at time slot $t_1 + \Delta t$

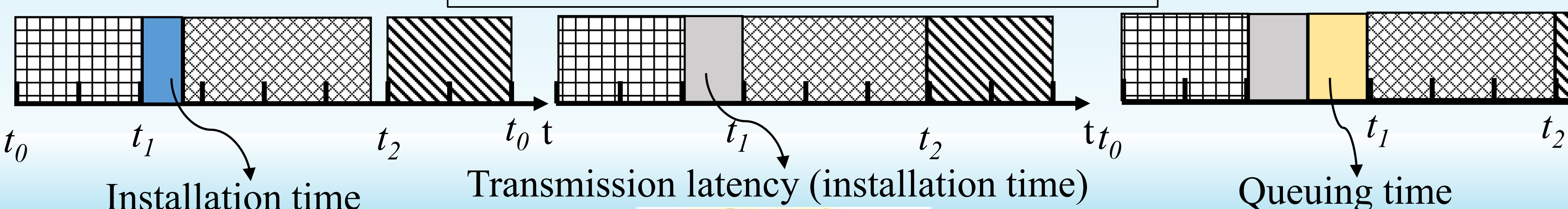
If doesn't exist ⇩

Select the nearest available node ***M***, install an instance of VNF i+1, establish a connection

⇩ If exists

Add to the buffer of that node, establish a connection

Update the sets of available nodes



UC DAVIS NETWORKS RESEARCH LAB

VNF-FGE & Scheduling

3. If Set i is empty but Set i+1 is not

Select the node (***N***) with lightest load in the set, check whether there is enough capacity to install the instance of VNF i

↓ If there is

Set up an instance of VNF i in node ***N***

↓ If not

Check the available nodes hosting VNF i at time slot $t_1 + \Delta t$

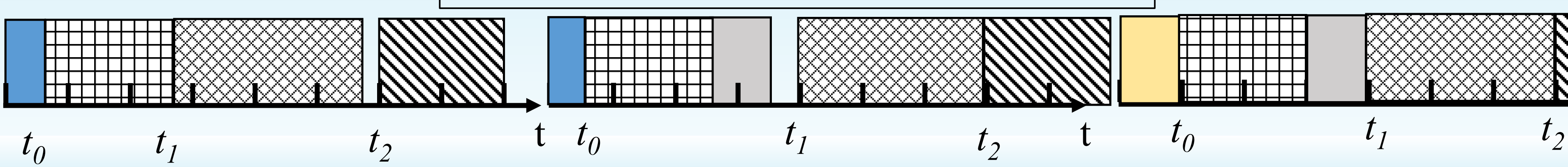
If doesn't exist ↓

Select the nearest available node ***N'***, install an instance of VNF i, establish a connection

↓ If exists

Add to the buffer of that node, establish a connection

Update the sets of available nodes



VNF-FGE & Scheduling

4. If both sets are empty

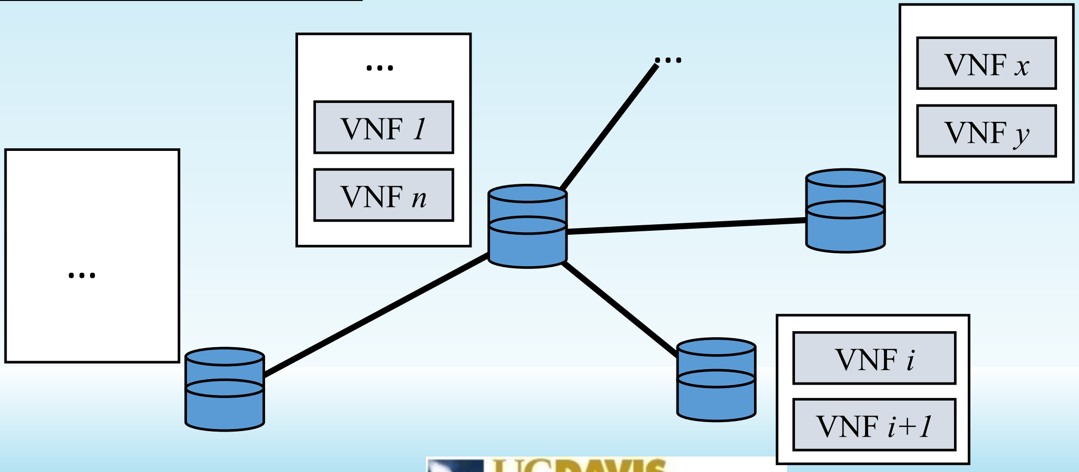
Keep finding until the last set or not a empty set

All sets are empty

Select the node having the largest capacity and install VNFs in the specified order; if the node doesn't have enough capacity, find the nearest datacenter

Find a not empty set (Set *n*)

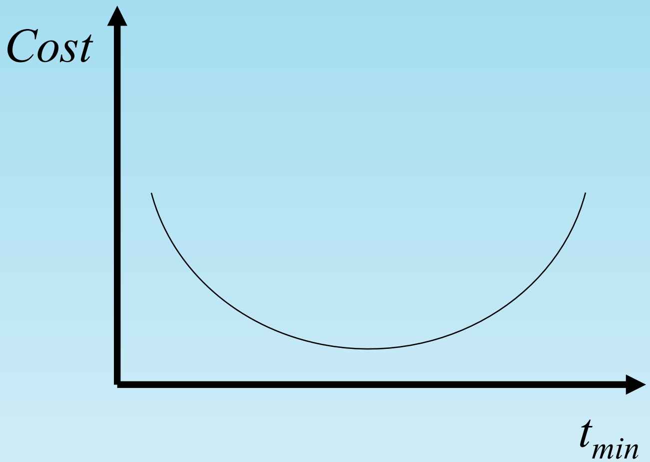
Install VNFs in the specified order; if the node doesn't have enough capacity, find the nearest datacenter



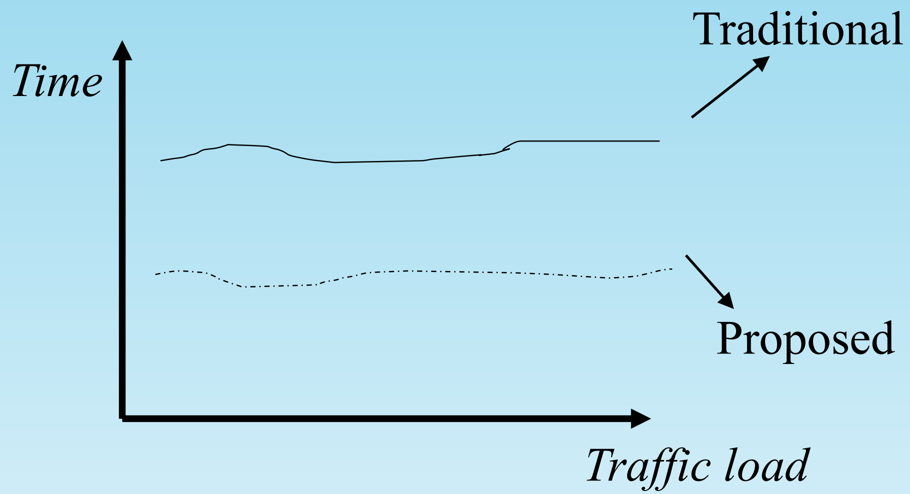
UCDAVIS NETWORKS RESEARCH LAB

Expected Results

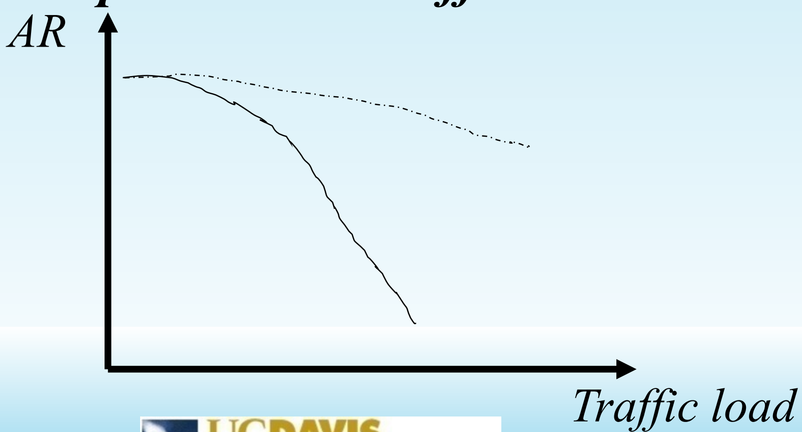
Network cost vs t_{min}:



Average serving time vs traffic load:



Accepted ratio vs traffic load:



UC DAVIS NETWORKS RESEARCH LAB

Suggestions

- ✓ **Employ Flow Deviation Algorithm**
- ✓ **Refer to energy efficient ethernet**

…

Thanks!