




NEW YORK INSTITUTE OF TECHNOLOGY

INCS 775
Data Center Security

Network Function Virtualization (NFV)



Dr. Zakaria Alomari
zalomari@nyit.edu

Today's Objectives

- Network Functions Virtualization
- NFV advantages
- Service Function Chaining
- VNF placement
- Real Practice

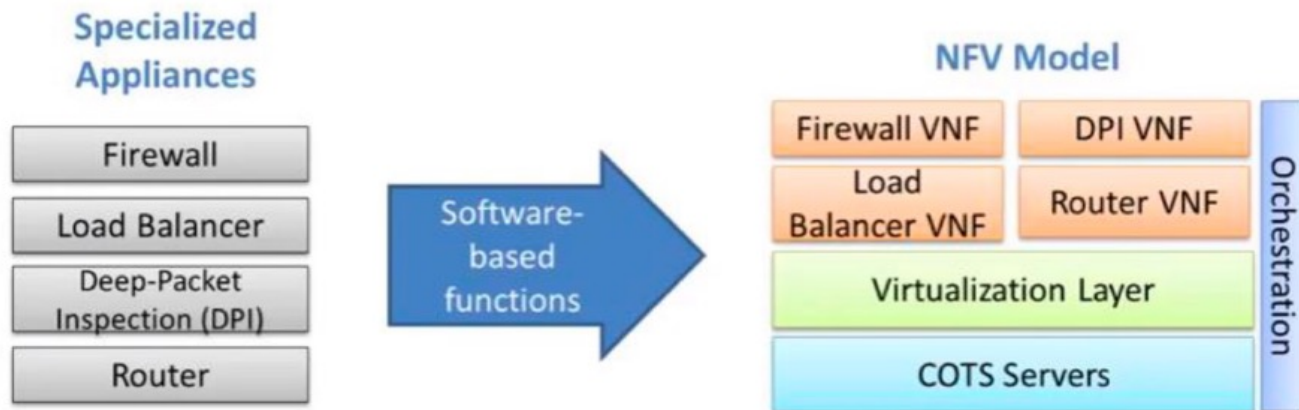
What is NFV?

- **Network Functions Virtualization (NFV)** is an **emerging technology** that provides a new approach to design, deploy, and manage networking services.
- NFV is considered as a way to **reduce cost** and **expedite service deployment** for **network operators**.
 - This is carried out by **decoupling** the **network functions (NFs)**, such as IDSs, Firewalls, NATs, DNSs from the hardware of the **traditional middle-boxes** and packaged as **virtual machines (VMs)** on commodity hardware so they can run in software.

According to the European Telecommunications Standards Institute (ETSI (2012))

What is NFV?

- Move network function from **specialized appliance** to **applications that run on Commercial off-the-shelf (COTS) equipment**.
- Individual **Virtual Network Functions (VNFs)** are an essential element of **NFV architecture**. They are handling specific **NFs** that run in one or more **VMs** on top of the **hardware networking infrastructure**.



Advantage of VNF

- Flexibility:
 - NFV gives providers the flexibility to run VNFs across different servers or move them around as needed when demand changes.
- Cost Reduction:
 - Reduce costs in purchasing network equipment via migration to software on standard servers.
 - Reduced maintenance and hardware costs
 - Reduced network maintenance costs
- Scalability/Elasticity
 - Service capacity can be flexed in real time to meet demand, ensuring hardware resources are not lying idle at times of low demand, and making it possible to handle unexpectedly high peak loads when needed.
- Time to Market
 - The total length of time it takes to bring a product from conception to market availability

Advantage of VNF

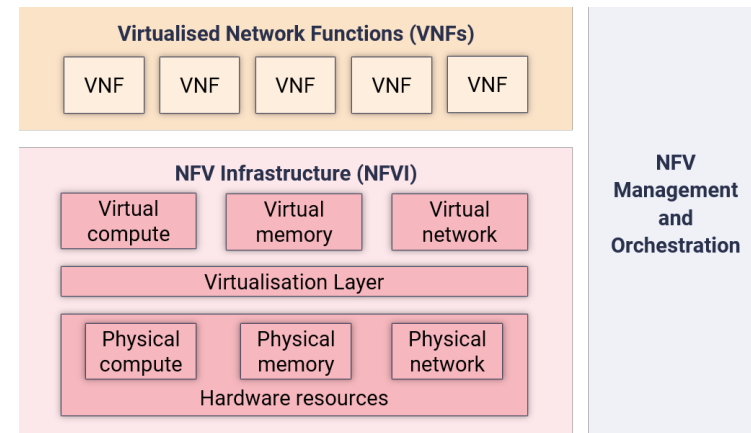
- Innovation
 - Simply put, innovation is about successfully implementing a new idea and creating value for your customers and stakeholders.
- Reduced space needed for network hardware
- Reduce network power consumption
- Easier network upgrades

NFV Framework (ETSI)

- ETSI identified **three** main working domains:

1) Virtual Network Functions (VNFs): These are **individual functions of a network** that have been virtualised. Possibilities are endless – **firewalls, Evolved Packet Core**, etc.

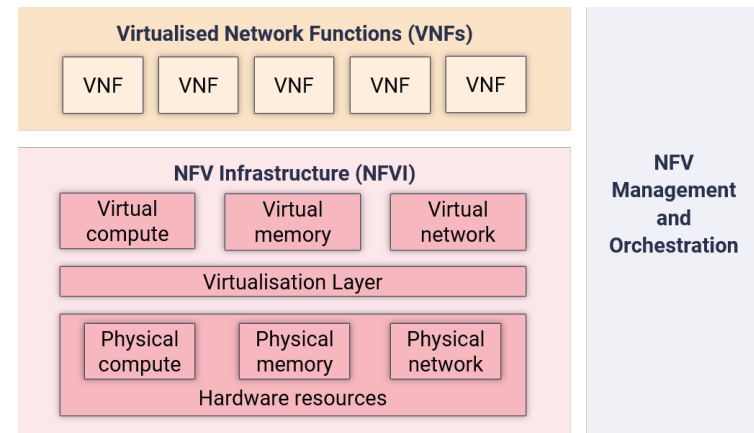
2) NFV infrastructure (NFVI): The infrastructure required to run the VNFs. This is **made up of hardware resources** (computing servers and network switches), and **virtual resources** (“abstractions” of the hardware on which the VNFs run, known as “virtual machines” (VMs)). A **virtualisation layer** (the “hypervisor”) exists to abstract between the two.



High-level ETSI NFV Framework

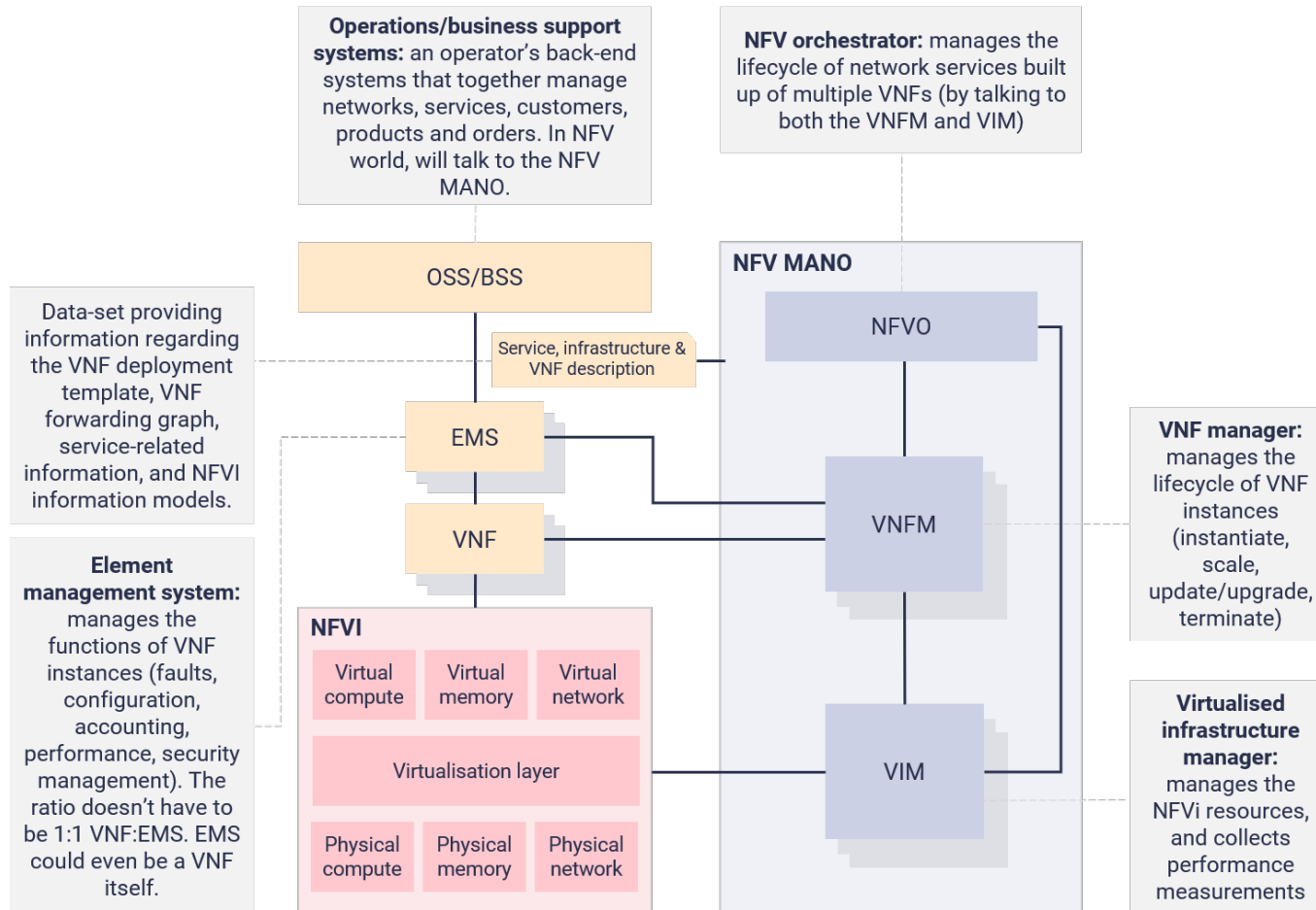
NFV Framework (ETSI)

- 3) NFV management and orchestration (MANO): MANO is the **framework** for management and orchestration of all the **resources in the NFV environment**. It is where the management of resources in the infrastructure layer takes place, and is also where resources are created and delegated and allocation of VNFs is managed.

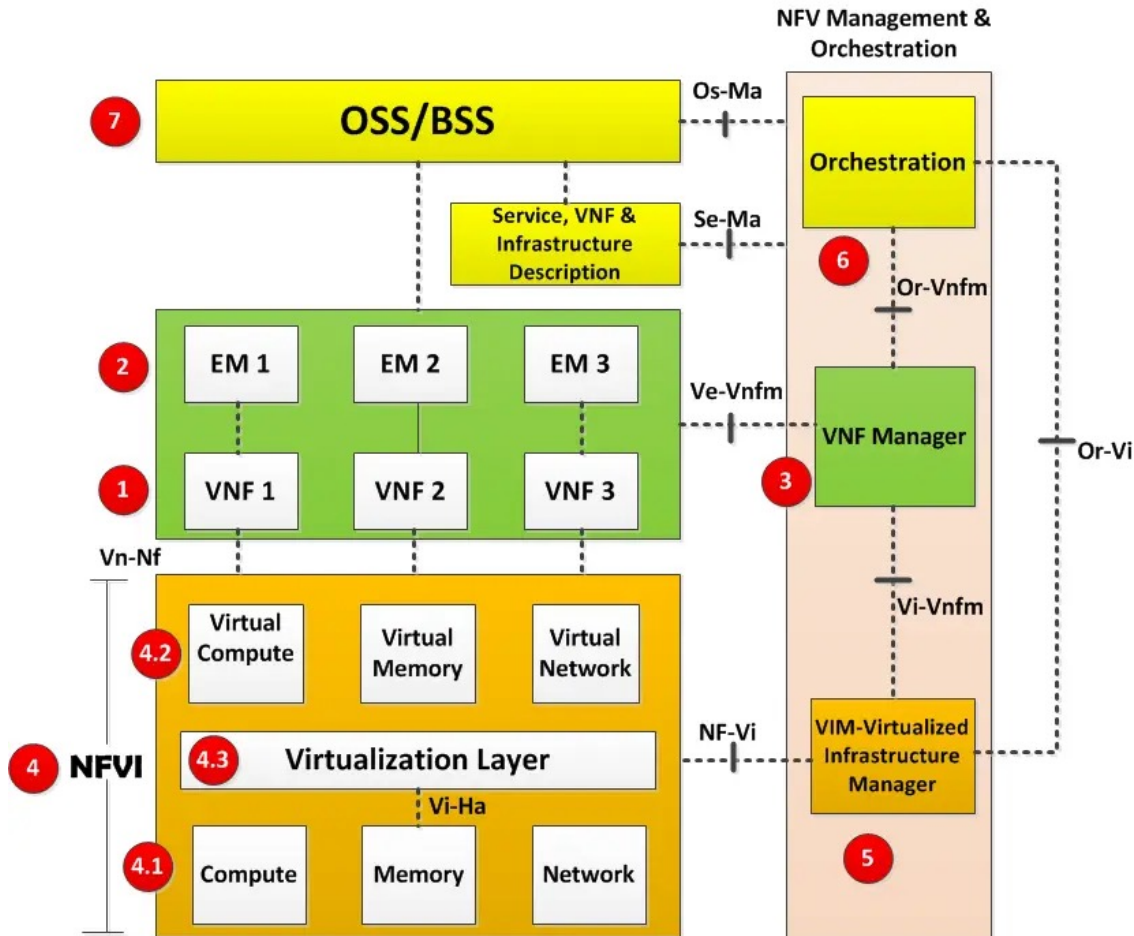


High-level ETSI NFV Framework

Full Architectural Framework



NFV Architecture



NFV Architecture

1. VNF (Virtualized Network Function):

A VNF is the basic block in NFV Architecture. It is the virtualized network element. For example when a router is virtualized, we call it Router VNF; another example is base station VNF.

2. EM (Element Management):

This is the element management system for VNF. This is responsible for the functional management of VNF i.e. FCAPS (Fault, Configuration, Accounting, Performance and Security Management). This may manage the VNFs through proprietary interfaces. There may be one EMS per VNF or an EMS can manage multiple VNFs. EMS itself can be a VNF.

NFV Architecture

3. VNF Manager:

- A VNF Manager manages a VNF or multiple VNFs i.e. it does the life cycle management of VNF instances. Life cycle management means setting up/ maintaining and tearing down VNFs.
- Additionally VNFM (VNF Manager) does the FCAPS for the virtual part of the VNF.
- The difference between EM and VNFM should be noted. EM does the management of functional components. While the VNFM does the management for the virtual components.

NFV Architecture

- **Element Management (EM):** **Manages individual components** like servers and switches within the **NFV infrastructure**. For example, EM might handle firmware updates for a physical server hosting virtual network functions.
- **Virtualized Network Function Manager (VNFM):** **Manages the lifecycle of virtualized network functions (VNFs)**. For instance, VNFM could orchestrate the deployment and scaling of a virtual firewall across multiple virtual machines in response to network traffic demands.

NFV Architecture

4. NFVI (Network Functions Virtualization Infrastructure):

NFVI (NFV Infrastructure) is the environment in which VNFs run. This includes Physical resources, virtual resources and virtualization layer, described below:

4.1 Compute, Memory and Networking Resources:

This is the physical part in NFVI. Virtual resources are instantiated on these physical resources. Any commodity switch or physical server/storage server is part of this category.

4.2 Compute, Memory and Networking Resources:

This is the virtual part in NFVI. The physical resources are abstracted into virtual resources that are ultimately utilized by VNFs.

4.3 Virtualization Layer:

This layer is responsible for abstracting physical resources into virtual resources. The common industry term for this layer is “Hypervisor”. This layer decouples software from hardware which enables the software to progress independently from hardware.

NFV Architecture

5. VIM (Virtualized Infrastructure Manager):

This is the **management system for NFVI**. It is responsible for **controlling and managing the NFVI compute, network resources and storage resources within one operator's infrastructure domain**. It is also responsible for **collection of performance measurements and events**.

6. NFV Orchestrator:

- **Generates, maintains and tears down network services of VNF themselves**. **If there are multiple VNFs, orchestrator will enable creation of end to end service over multiple VNFs.**
- **NFV Orchestrator** is also **responsible for global resource management of NFVI resources**. For example **managing the NFVI resources i.e. compute, storage and networking resources among multiple VIMs in network**.

NFV Architecture

- The **Orchestrator** performs its functions by NOT talking directly to VNFs but through VNFM and VIM.
- Example:

Let's say there are multiple VNFs which need to be chained to create an end to end service. One example of such case is a virtual IPS and a virtual FW. They can be from same or different vendors. There will be a need to create an end to end service using both VNFs. This would demand a service orchestrator to talk to both VNFs and create an end to end service.

Note that VIM, VNFM and NFVO together are also called Management and Network Orchestration (MANO)

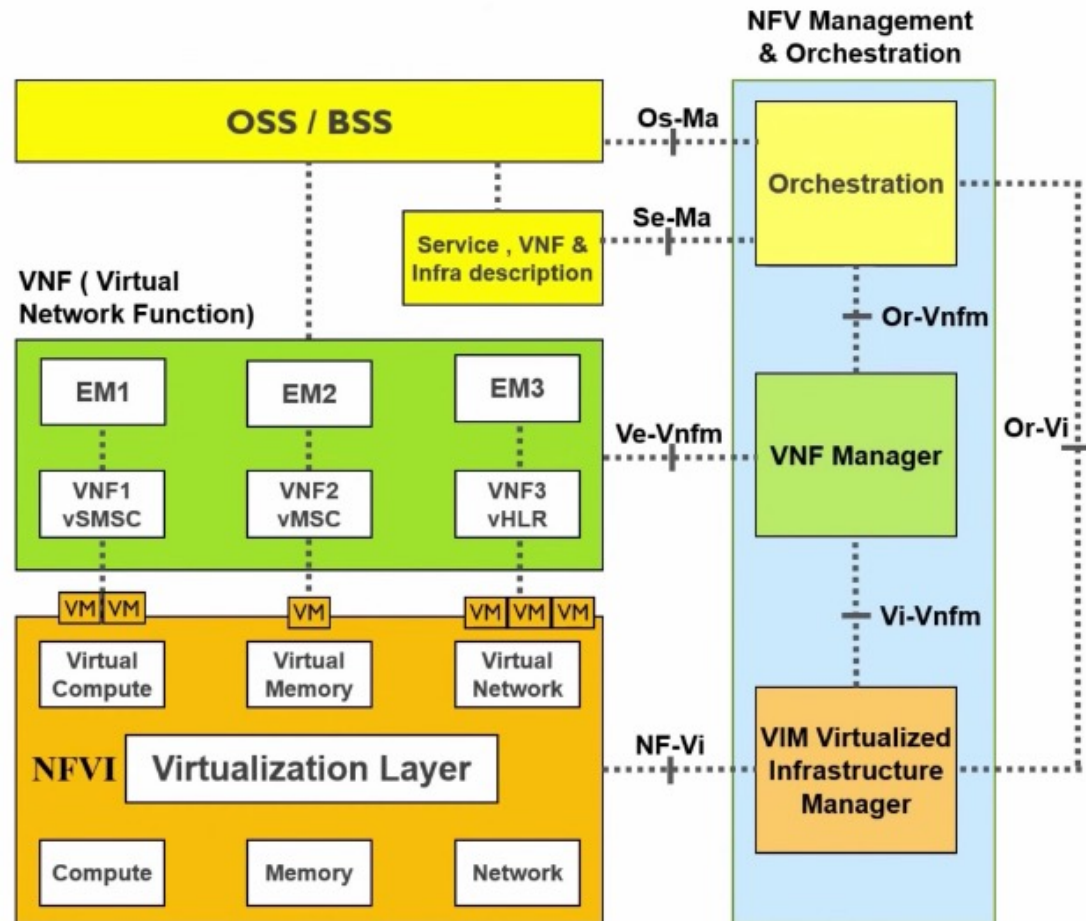
NFV Architecture

7. OSS/BSS(Operation Support System/Business Support System)

- **OSS** deals with network management, fault management, configuration management and service management. **BSS** deals with customer management, product management and order management etc.
- In the **NFV architecture**, the current BSS/OSS of an operator may be integrated with the **NFV Management and Orchestration** using standard interfaces.

NFV Framework (ETSI)

<https://www.youtube.com/watch?v=VI5UJUR1uV4>



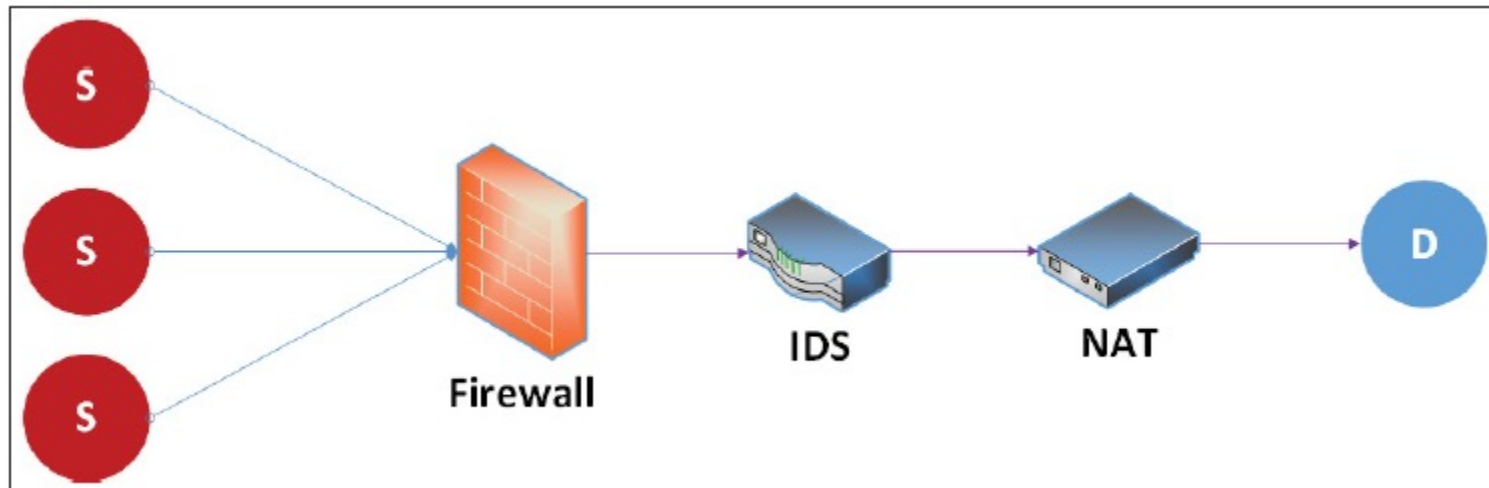
Challenges

- NFV makes the provisioning of **network functions** more **flexible and cost-effective**, but like other **emerging technology**, it brings several challenging to network operators, such as:
 - Their dynamic instantiation,
 - Migration,
 - Placement, and etc.

Service Function Chaining in Data Centers

- **Service Function Chaining (SFC)** is a technique for selecting and steering data traffic flows through various NFs “Network Functions” based on software defined networking (SDN).
- The network service chain consists of a set of VNFs that can process incoming traffic in a specific order. This sequence is called a service function chain (SFC) or VNF Forwarding Graph (VNF-FG)

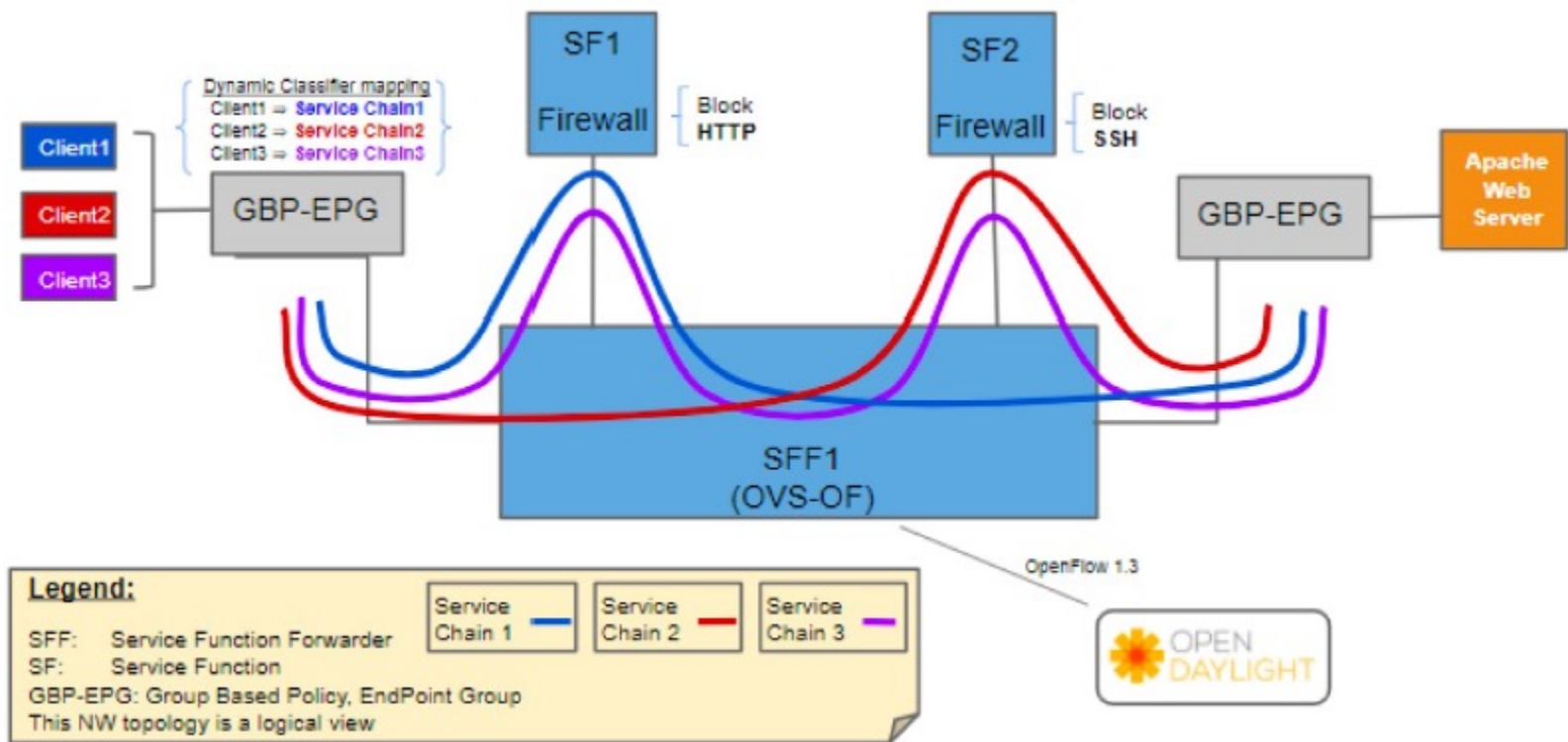
Service Function Chaining in Data Centers



Service Function Chain

Service Function Chaining in Data Centers

- Service Function Chaining (SFC):



Service Function Chaining in Data Centers

- **Service Function Forwarders**

The **Service Function Forwarder (SFF)** is the **core element** used in **Service Chaining**. It is an **OpenFlow switch** that, in the context of OPNFV (Open Platform for NFV), is hosted in an OVS bridge (Open vSwitch). In OPNFV there will be one **SFF per Compute Node** that will be hosted in the “br-int” OpenStack OVS bridge.

The responsibility of the SFF is to **steer incoming packets to the corresponding Service Function**, or to **the SFF in the next compute node**. The flows in the SFF are programmed by the **OpenDaylight SFC SDN Controller**.

- **Service Functions**

A Service Function (SF) is a **Function that provides services to flows traversing a Service Chain**. **Examples of typical SFs include: Firewall, NAT, QoS, and DPI**. In the context of OPNFV, the **SF** will be a **Virtual Network Function**. The SFs receive data packets from a Service Function Forwarder.

Service Function Chaining in Data Centers

- **Group-based policies (GBP)**

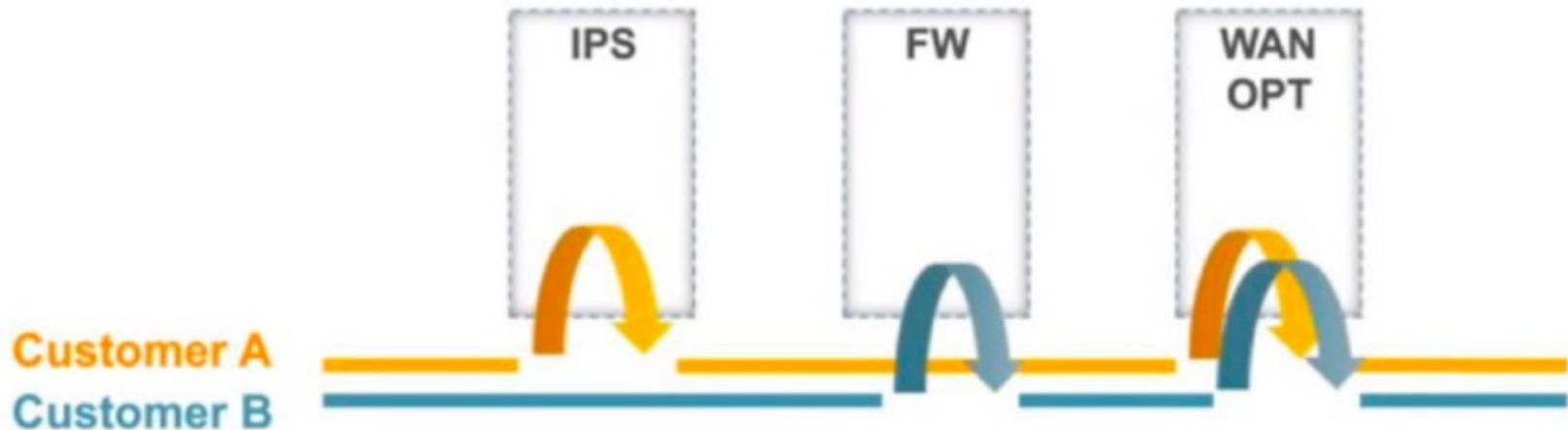
Group-based policies (GBP) provide the basis for network segmentation. Once you have established logical groupings for all your users and things, provided you have the support from the underlying network infrastructure, you can permit or deny each group access to protected resources.

- **Network endpoint group (NEG)**

A network endpoint group (NEG) is a configuration object that specifies a group of backend endpoints or services. With NEGs, Google Cloud load balancers can serve VM instance group-based workloads, serverless workloads, and containerized workloads.

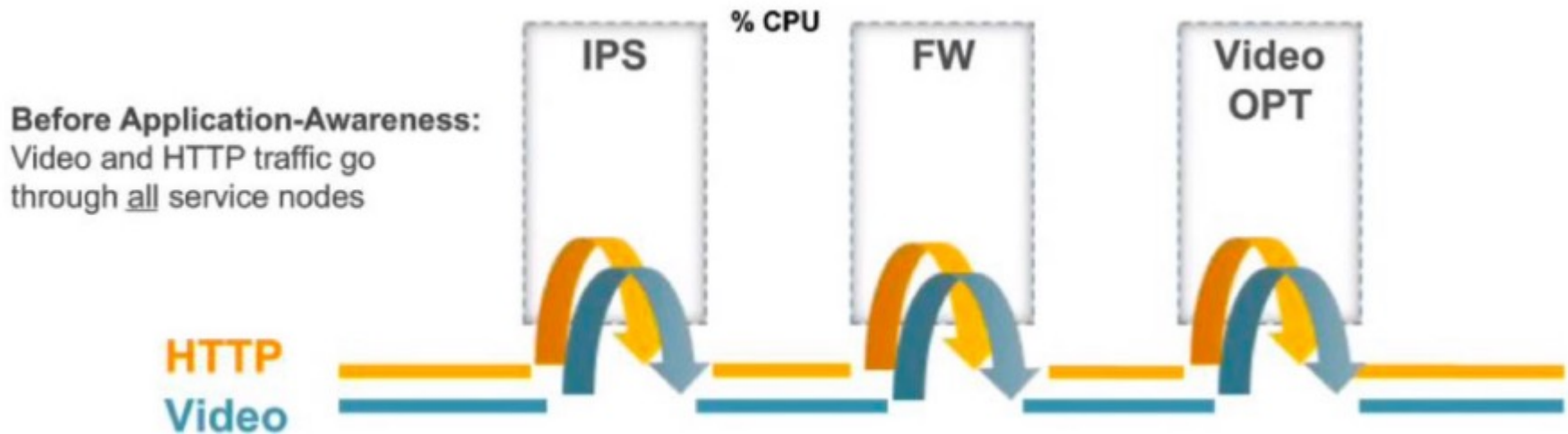
Service Function Chaining in Data Centers

- Single set of VNFs can serve multiple customers



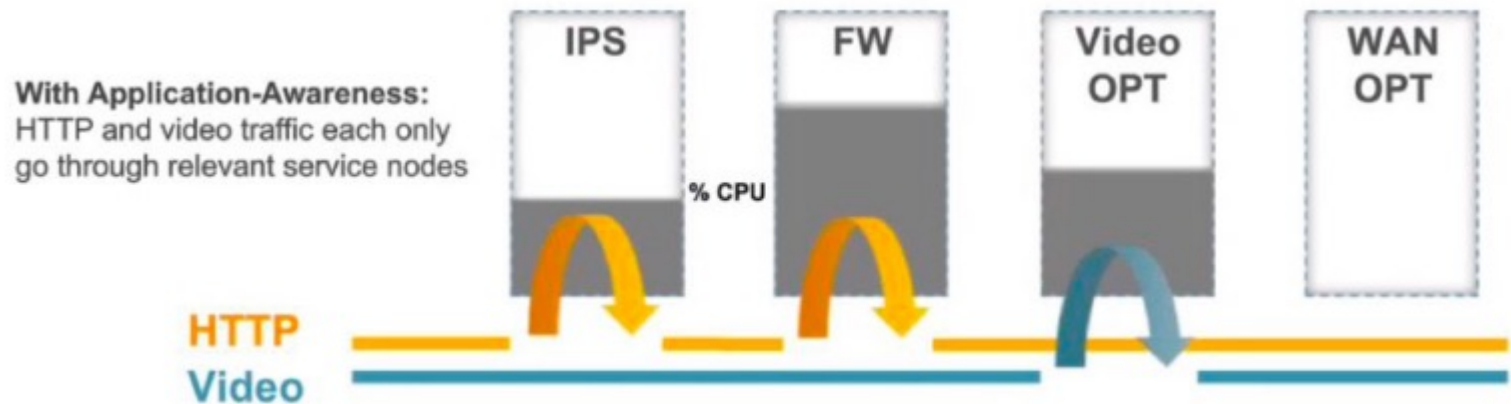
SFC Challenge Before Application-Awareness

- Inefficient service chaining
 - All traffic is **routed through all service nodes**.
 - VMs have to be provisioned for **peak traffic**.

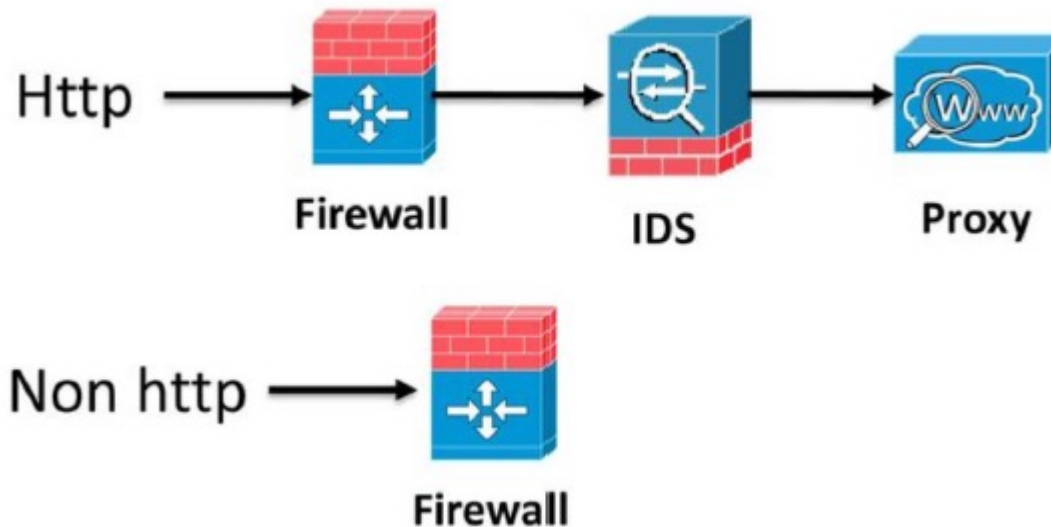


Create New Revenue by SFC Optimization

- With an **application-aware network infrastructure**, network traffic can be routed intelligently, in the right sequence and only to **relevant service nodes**.
- This intelligence:
 - Enables service automation
 - Optimizes usage of infrastructure
 - Enables new type of differentiated service combination to generate

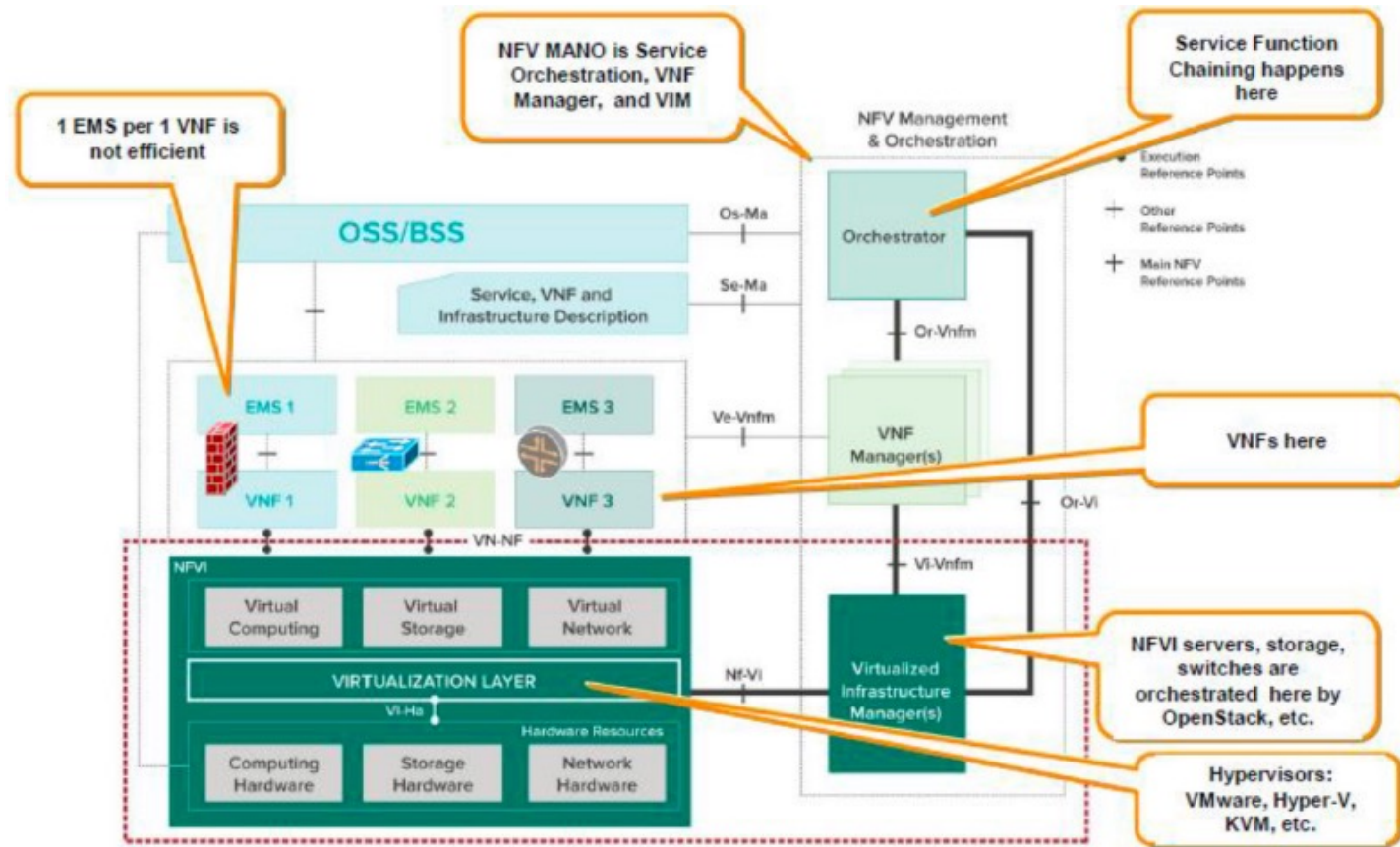


Application-aware SFC



- **Correctness:** sequential order
- **Efficiency:** not traverse unnecessary ones

SFC is Part of NFV MANO



VNF Placement

Integer Linear Programming

- ILP stands for **Integer Linear Programming**, which is a mathematical optimization technique used to solve problems where the goal is to maximize or minimize a linear objective function subject to linear equality and inequality constraints, with the additional constraint that some or all of the variables must take integer values.

Integer Linear Programming

- In this lecture we will formulate linear integer programming model involving **Binary** or **0-1 variables**.
- **Binary variables** are employed when there is a **YES** or **NO** situation.
- That is, to indicate whether a selection is made or not.

Integer Linear Programming

For example:

- Suppose we have 4 different projects to consider.
- We can either select a project, or not select it.
- So for the first project we can define the decision variable as follows:

$X_1 = 1$ if project 1 is selected, 0 otherwise

We do the same for projects 2,3, and 4 by defining X_1 , X_2 , and X_3

$X_2 = 1$ if project 2 is selected, 0 otherwise

$X_3 = 1$ if project 3 is selected, 0 otherwise

$X_4 = 1$ if project 4 is selected, 0 otherwise

Integer Linear Programming

- Or we can simply write $X_i = 1$ if project i is selected and 0 if not selected

$X_i = 1$ if project i is selected, 0 otherwise

Where $i = 1, 2, 3, \text{ and } 4$

Project	1	2	3	4
January	58	44	26	23
February	25	29	13	17
March	43	25	23	29

outlays (costs)

Now, suppose each project has the same lifespan of 3 months (January, February, and March), with corresponding outlays or costs (in thousands of dollars) as we can see in the table.

Integer Linear Programming

Project	1	2	3	4	Available Funds
January	58	44	26	23	120
February	25	29	13	17	80
March	43	25	23	29	95

Suppose these are the funds available for selected projects each month.

Project	1	2	3	4	Funds
January	58	44	26	23	120
February	25	29	13	17	80
March	43	25	23	29	95
Net Return	217	125	88	109	

and these are the net returns (in thousand dollars) from each project.

Project	1	2	3	4	Funds
January	58	44	26	23	120
February	25	29	13	17	80
March	43	25	23	29	95
Return	217	125	88	109	

In this case, our objective is to maximize returns which is $217X_1 + 125X_2 + 88X_3 + 109X_4$

Maximize $217X_1 + 125X_2 + 88X_3 + 109X_4$

Integer Linear Programming

Project	1	2	3	4	Funds
January	58	44	26	23	120
February	25	29	13	17	80
March	43	25	23	29	95
Return	217	125	88	109	

Since projects outlays are constrained by available funds, we write for January, February, and March:

$$\begin{aligned}
 &\textbf{Maximize} \quad 217X_1 + 125X_2 + 88X_3 + 109X_4 \\
 &\textbf{s.t.} \quad \begin{aligned}
 &58X_1 + 44X_2 + 26X_3 + 23X_4 \leq 120 \quad (\text{January}) \\
 &25X_1 + 29X_2 + 13X_3 + 17X_4 \leq 80 \quad (\text{February}) \\
 &43X_1 + 25X_2 + 23X_3 + 29X_4 \leq 95 \quad (\text{March})
 \end{aligned}
 \end{aligned}$$

Integer Linear Programming

Solving using LINDO or Excel's Solver

$$X_1=1 \quad X_2=0 \quad X_3=1 \quad X_4=1$$

Optimal Return = 414

Maximize $217X_1 + 125X_2 + 88X_3 + 109X_4$

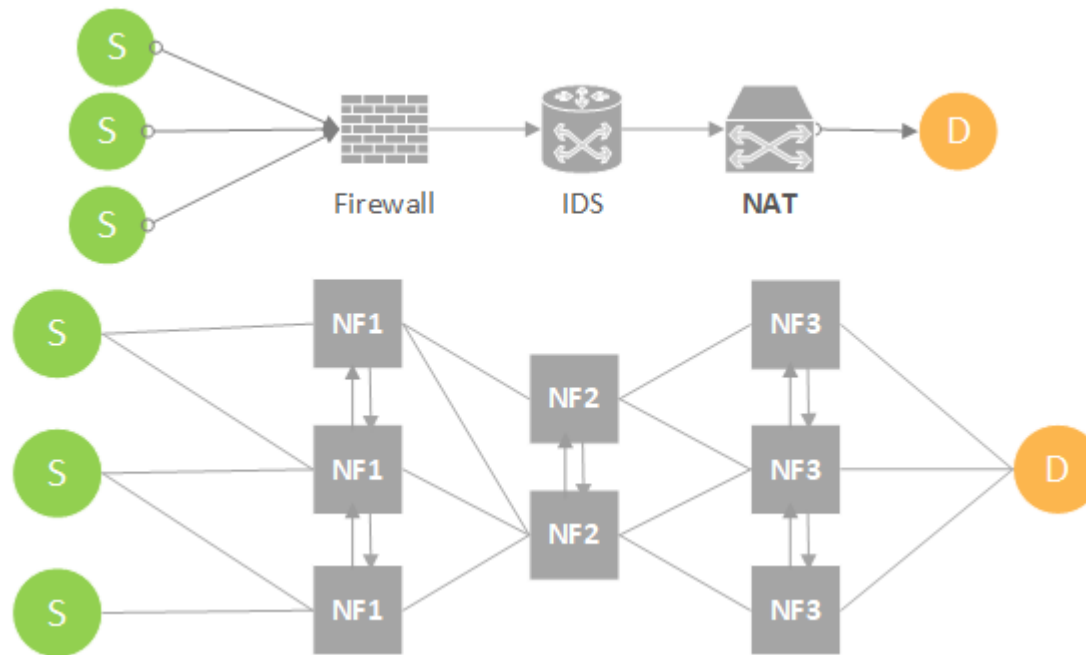
s.t. $58X_1 + 44X_2 + 26X_3 + 23X_4 \leq 120$ (January)
 $25X_1 + 29X_2 + 13X_3 + 17X_4 \leq 80$ (February)
 $43X_1 + 25X_2 + 23X_3 + 29X_4 \leq 95$ (March)

And then complete the model by stating that the decision variables must be binary.

Upon solving this model software like LINDO or Excel Solver, we find that the optimal solution is $X_1 = 1$, $X_2 = 0$, $X_3 = 1$, and $X_4 = 1$ with a corresponding net return on 414.

This is, to maximize net return, undertake project 1, 3 and 4 only.

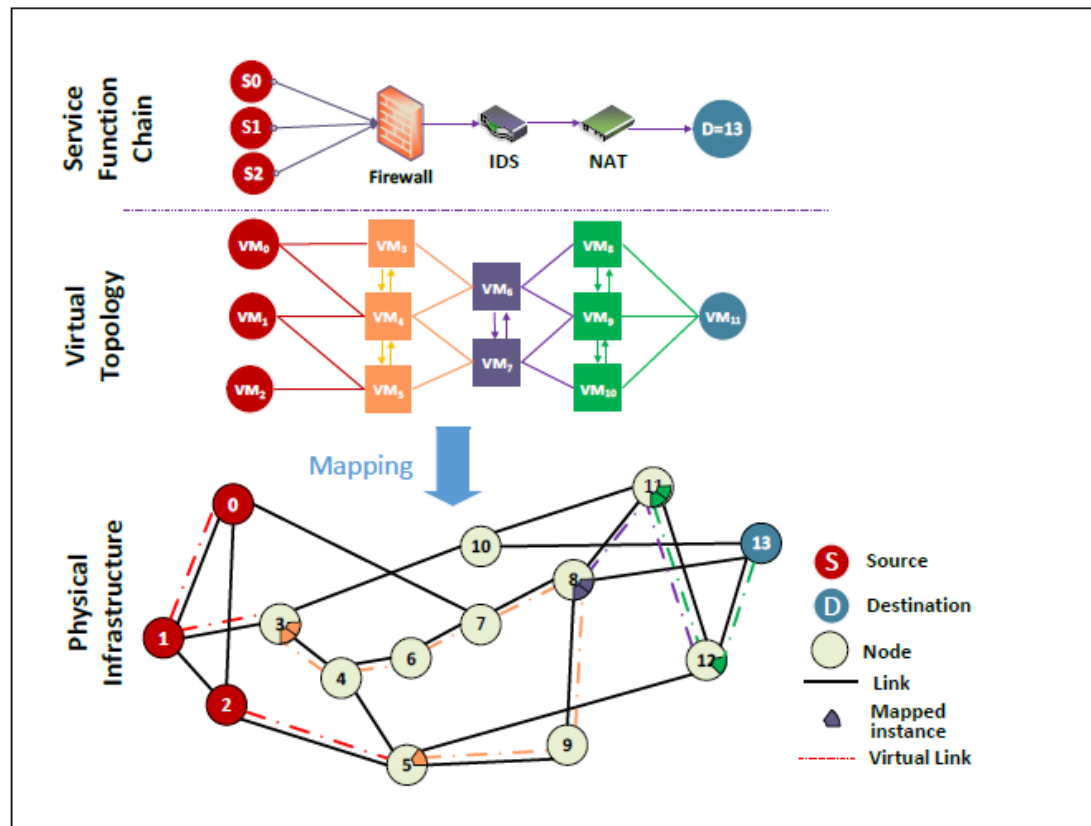
Example of VNF placement



Service Function Chain



Example of VNF placement



SFC embedding problem

Mapping Phase: Problem Formulation

- In this lecture, we **formulate** the **SFC mapping problem** as an **Integer Linear Program (ILP)** with the **objective** of **minimizing the Service function chaining (SFC) provider's operational cost** in terms of instance deployment costs, bandwidth and synchronization costs.

Mapping Phase: Problem Formulation

- The **physical infrastructure** owned by the **SFC provider** is made out from several **POPs** that are geographically distributed.
- The **infrastructure is modeled** by a **graph** $G = (N, P)$ where $N = \{0, 1, \dots, |N|\}$ represents the **set of POPs** and $P = \{(m, n) \in (N \times N) | m \text{ and } n \text{ are directly connected}\}$ denotes the **set of physical links** that connect the **POPs**.
- Each POP $n \in N$ contains an amount of **physical resources** C_n expressed as the **maximal number** of **t2.tiny** instances (VM) that the POP can host.
- Note that a **t2.tiny** (VM) instance contains **1 vCPU**, **1 GiB** and **1 GB** of **memory** and **disk**, respectively. A physical link $(m, n) \in P$ that connects the POP m with POP n has a bandwidth capacity B_{mn} .

Mapping Phase: Problem Formulation

- Furthermore, a **service function chain** is represented as a graph $V = (I, L)$ where $I = \{0, 1, \dots, |I|\}$ is the set of **virtual instances** in the chain and L is the **set of virtual links connecting them**.
- Each **VNF instance** $i \in I$ has a **resource requirement** of 1 vCPU, 1 GiB of memory, and 1GB of storage. Each **virtual link** $(i, j) \in L$ has bandwidth requirement b_{ij} .
- It is worth noting that, for simplicity, the endpoints of the chain (i.e., sources and destinations) are considered also instances with requested resources equal to zero. They are constrained to be mapped onto particular physical POPs that are provided in the VNF request.

Mapping Phase: Problem Formulation

- Furthermore, we define two decision variables:

1. The first one is denoted as $x_{im} \in \{0, 1\}$ and indicates whether or not VNF instance i is embedded into POP m .
2. The second **decision variable** is denoted as $y_{ij,mn} \in \{0, 1\}$. If $y_{ij,mn} = 1$, the **virtual link** (i, j) uses the **physical link** mn . It is worth noting that a **virtual link** is embedded through a **physical path** (i.e, multiple connected physical links). Hence, several physical links could be used to embed a virtual link. In other words, If $y_{ij,mn} = 1$, the physical link (m, n) is part of the physical path used to embed the virtual link (i, j) .

Mapping Phase: Problem Formulation

TABLE OF NOTATIONS

Symbol	Definition
$G = (N, P)$	Graph G where N is the set of nodes and P is set of physical links
$V = (I, L)$ instances	The virtual network Graph V with I is the set of VNF and L is the set of virtual links
C_n	Available capacity at POP $n \in N$ expressed in number of instances
B_{mn}	Bandwidth capacity of the physical link connecting nodes m and n
$b_{i,j}$	Bandwidth requirement of the virtual link connecting instances i and j
δ_{im}	Deployment costs per unit of time for VNF instance i into POP m
$\Delta_{m,n}$	Bandwidth cost per bandwidth unit in physical link (m, n)
f_{im}	Boolean constant set to 1 if VNF instance i has to be embedded into node m
s_{ij}	Boolean constant set to 1 if there is a synchronization between instances i and j
x_{im}	Boolean decision variable indicating whether or not instance i is embedded into node m
$y_{ij,mn}$	Boolean decision variable indicating whether virtual link (i, j) is mapped into physical link (m, n)
\mathbb{C}	Operational cost
\mathbb{S}	Synchronization cost

Mapping Phase: Problem Formulation

- **Objective Function:** The **objective function** when embedding an **SFC request** aims at minimizing the operational costs \mathbb{C} and synchronization cost of the embedded VNF instances \mathbb{S} . It can be expressed as:

$$J = \min_{\substack{(x_{im})_{i \in I, m \in N} \\ (y_{ij, mn})_{(i,j) \in L, (m,n) \in P}}} (\mathbb{C} + \mathbb{S}) \quad (1)$$

- In the following, we provide more details on how to compute the operational and synchronization costs:

Network Function Virtualization

MIDDLE-BOXES

Expensive hardware

Hard to deploy

Hard to modify

Hard to scale

Provision for peak-load

VIRTUAL NETWORK FUNCTIONS

Low-cost software

Easy to deploy

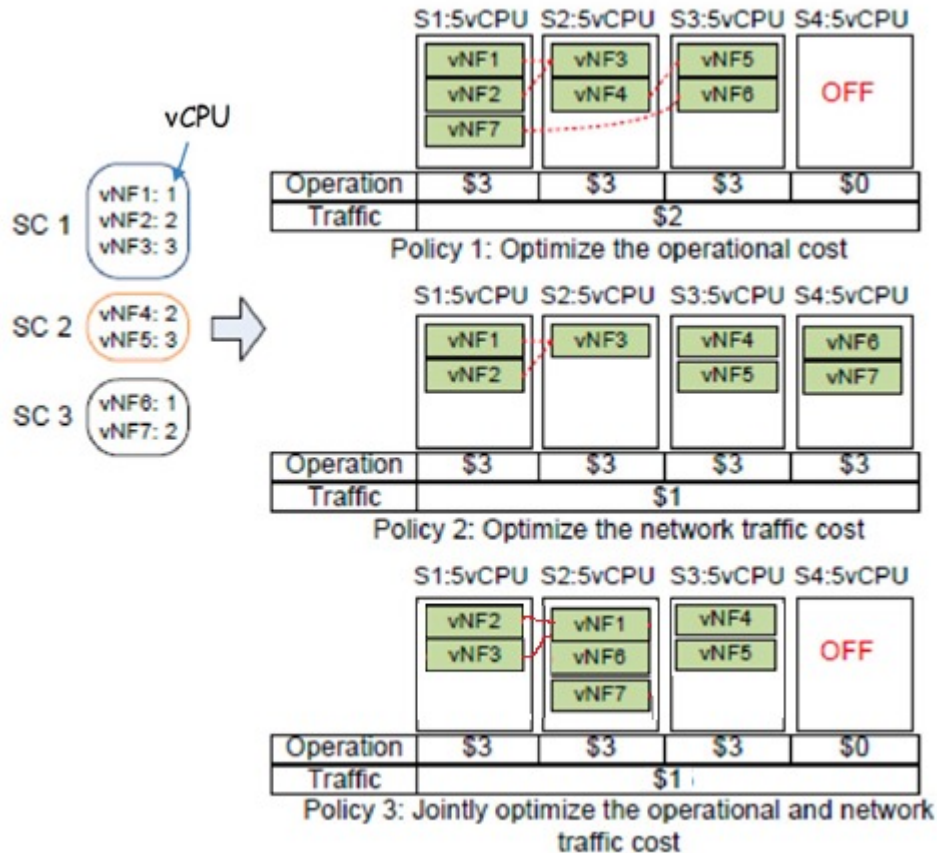
Easy to modify

Easy to scale

Scale resources on demand

Example of VNF placement

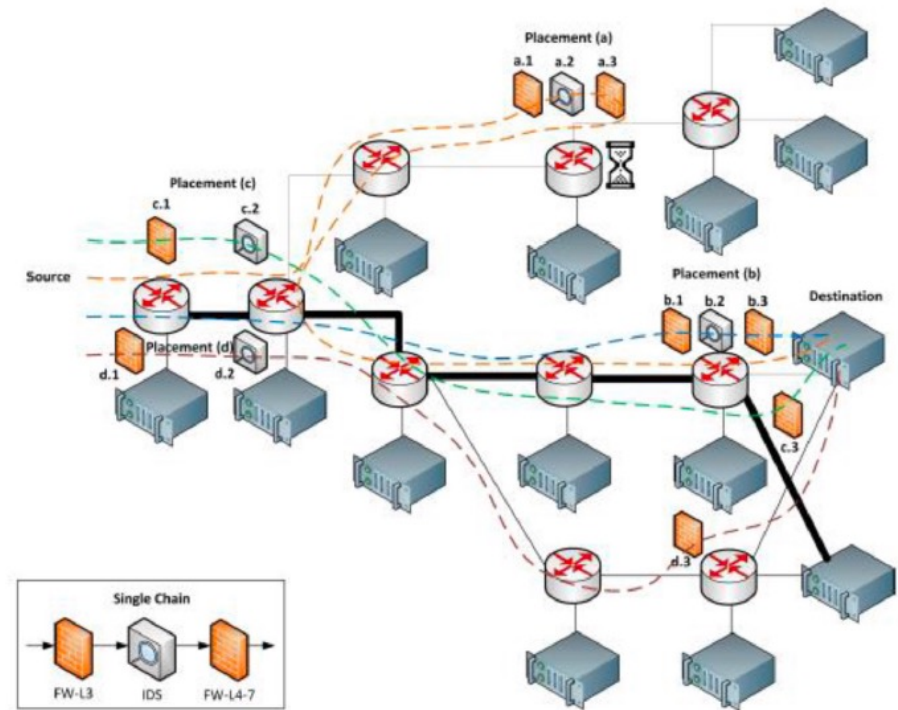
- For ease of exposition, we give the example with **only one resource type** (vCPU) of VNFs and physical nodes.
- Assume that the cost to operate one physical node (in terms of power) is \$3/hour and that network delay cost is \$0.5/link
- Electricity prices vary from a region to another, the running and management costs for each VNF were randomly varied [R1].



[R1] [Electric Power Monthly - U.S. Energy Information Administration \(EIA\)](#)

Example of VNF placement with different policies

- Virtual security appliances need to be efficiently deployed within the data center infrastructure to **minimize and balance** the **overall processing time** and the **processing power consumption** needed for performing different security functions.
- **Security appliances** are chained in specific order to perform different security functions on the traffic.



VNF placement with different policies

- In the context of **networking and virtualization**, **VNF placement** refers to the process of determining where to deploy or locate VNF within a network infrastructure.

Different policies can be used to guide the placement decisions. Let's explore some common VNF placement policies:

1. **Latency-aware placement:** This policy focuses on **minimizing network latency by placing vNFs closer to the end-users or the network components** they interact with most frequently.

VNF placement with different policies

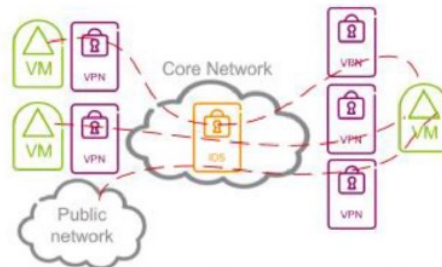
2. **Load-aware placement:** With this policy, VNFs are placed based on the current workload or resource utilization of different parts of the network.
3. **Cost-aware placement:** This policy takes into account the cost factors associated with deploying and operating VNFs. It considers factors like power consumption, network bandwidth, hardware requirements, and licensing fees.
4. **Fault-tolerant placement:** In this policy, VNFs are distributed across the network to enhance fault tolerance and resilience. By placing redundant vNF instances in geographically diverse locations, the network can continue to operate even if one or more instances fail.

VNF placement with different policies

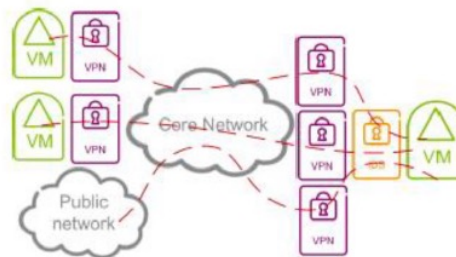
5. **Security-aware placement:** With the increasing focus on network security, this *policy considers the security requirements of vNFs and their sensitivity to potential threats*. By placing security-critical vNFs closer to the network perimeter or sensitive data sources, it becomes easier to implement security measures and monitor traffic effectively.

Placement Should Meet the Security Requirements

- Security requirements:
 - The traffic should be encrypted before reaching the untrusted core network and.
 - All traffic should be inspected by the IDS.



(a) Potentially optimal solution that does not meet all security requirements

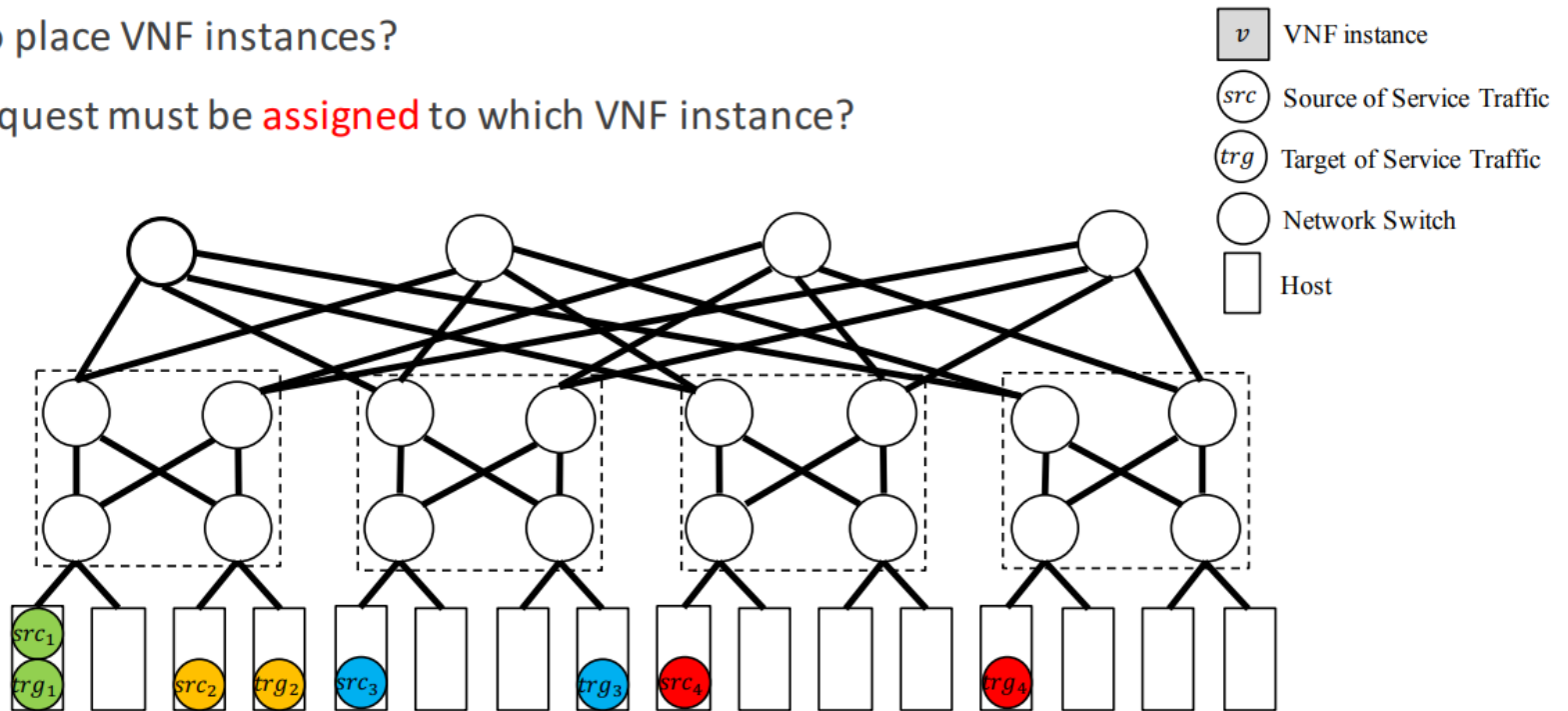


(b) Possible solution that satisfies all security requirements

VNF Services in Cloud

Where to place VNF instances?

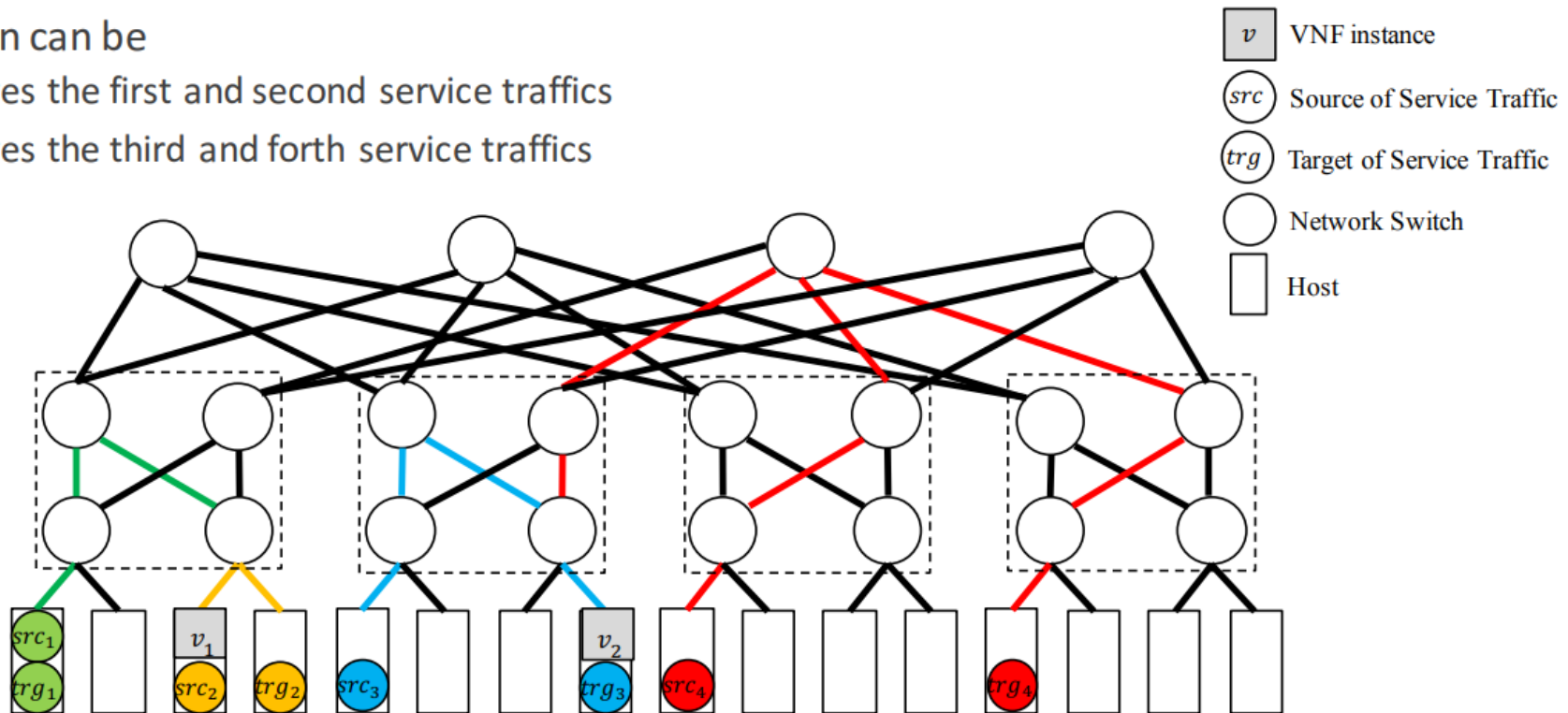
Which request must be assigned to which VNF instance?



VNF Services in Cloud

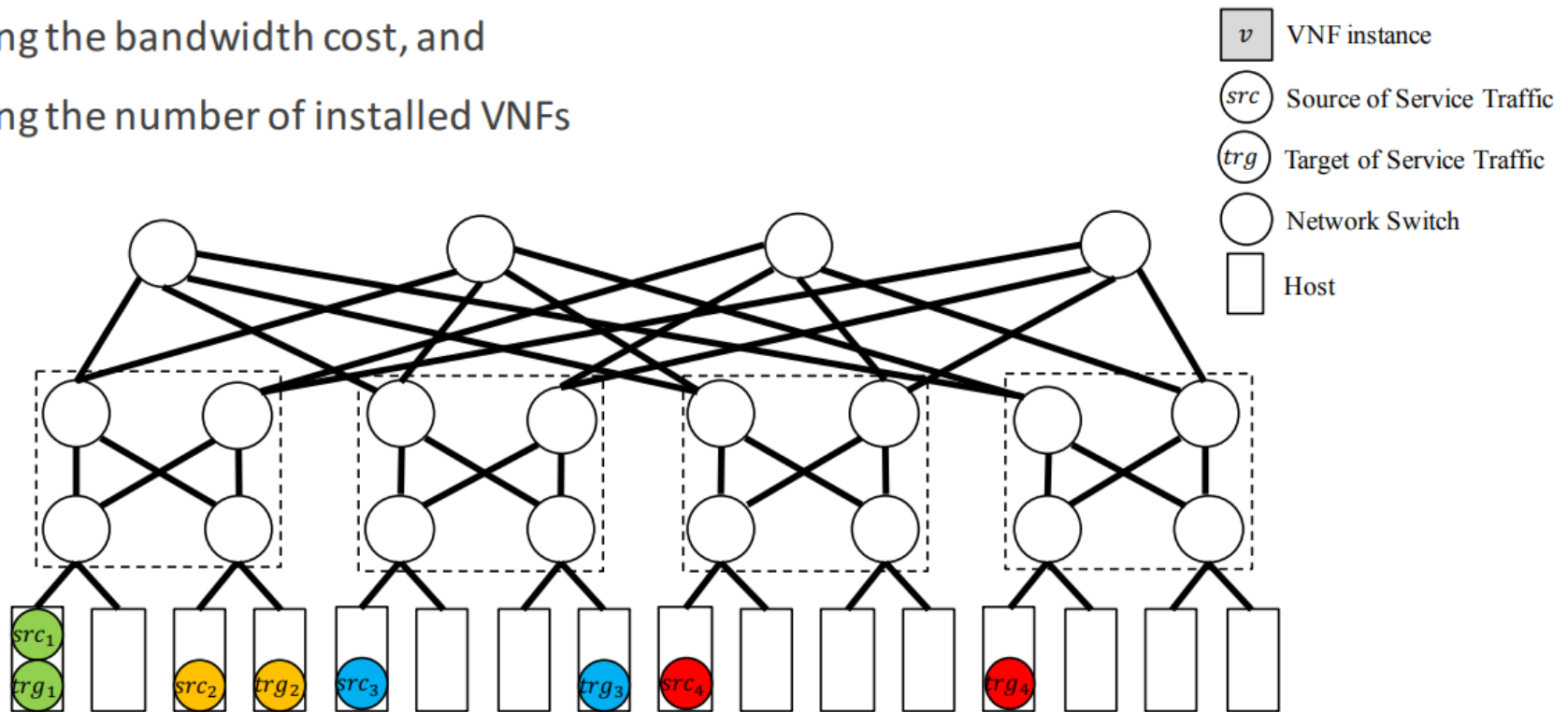
A solution can be

- v_1 serves the first and second service traffics
- v_2 serves the third and fourth service traffics



VNF Services in Cloud (Conflicting Objectives)

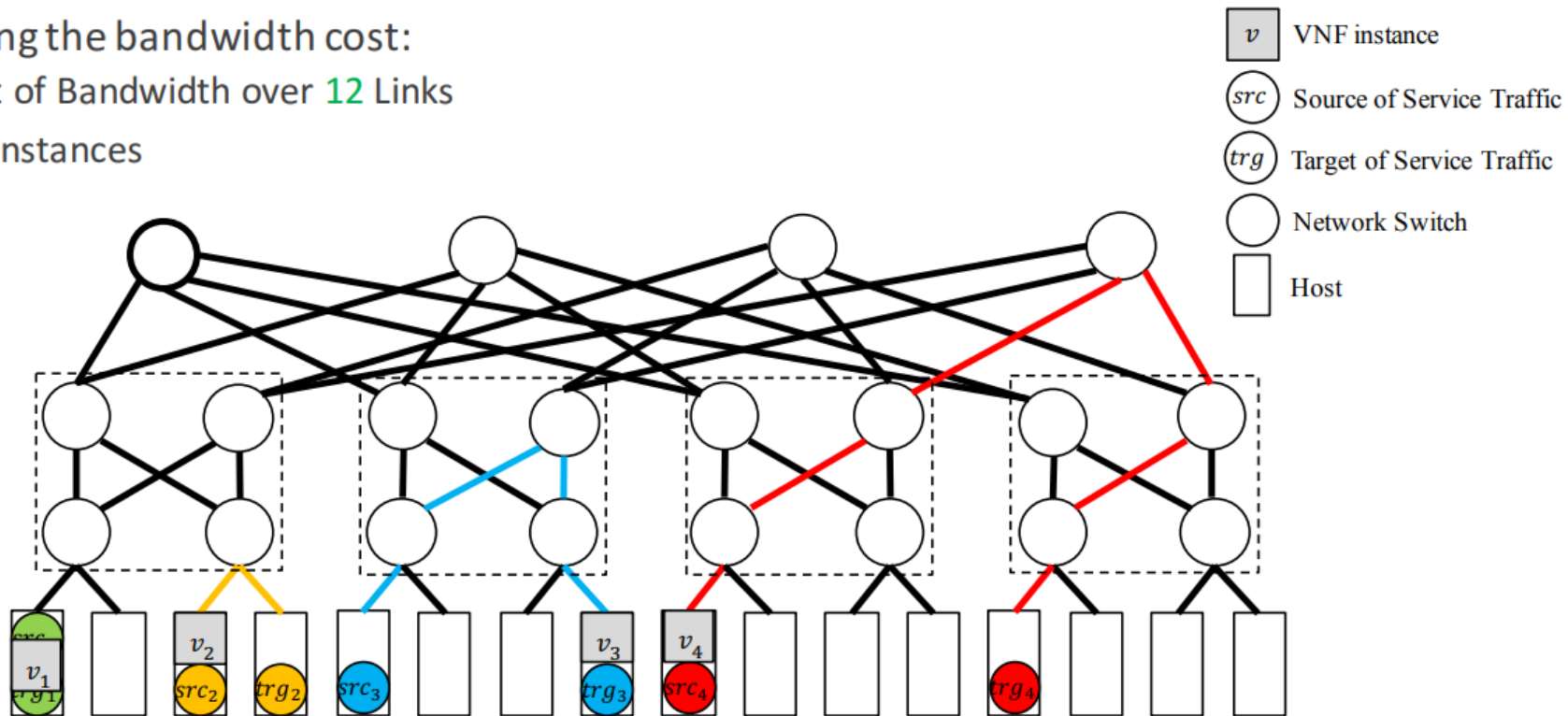
Minimizing the bandwidth cost, and
Minimizing the number of installed VNFs



VNF Services in Cloud (Conflicting Objectives)

Minimizing the bandwidth cost:

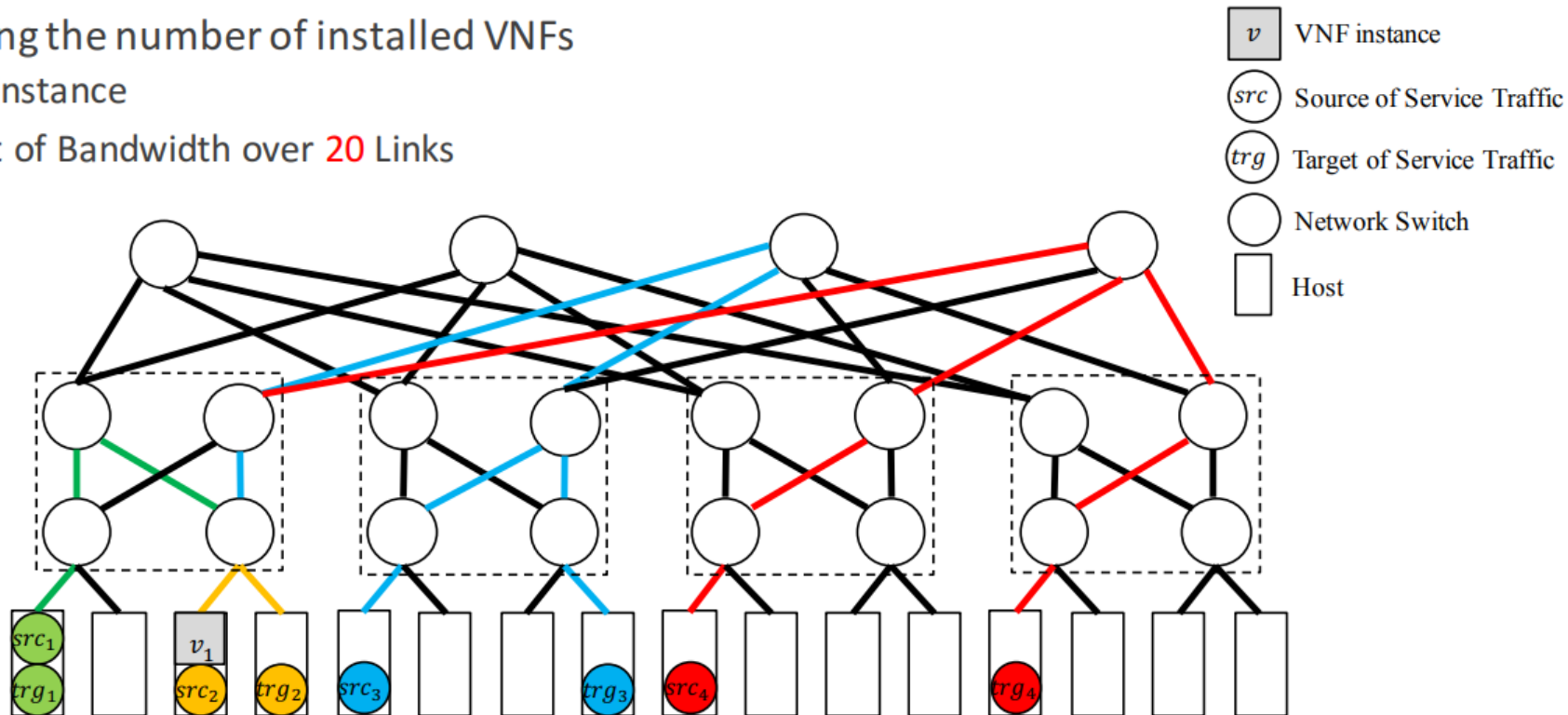
- 12 Unit of Bandwidth over 12 Links
- 4 VNF instances



VNF Services in Cloud (Conflicting Objectives)

Minimizing the number of installed VNFs

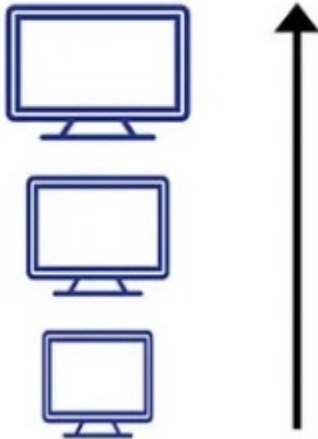
- 1 VNF instance
- 34 Unit of Bandwidth over 20 Links



Scaling

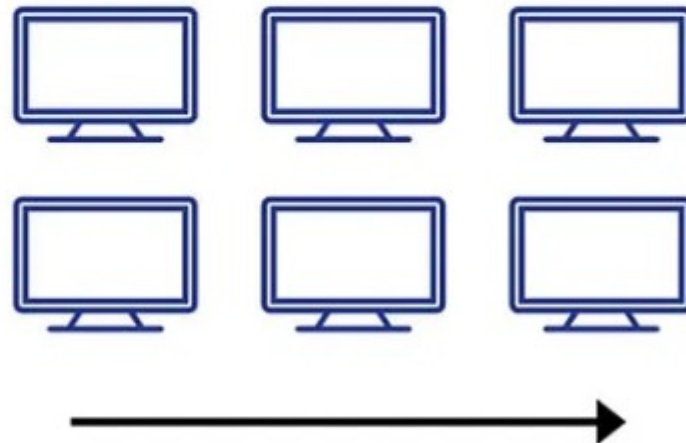
VERTICAL SCALING

Increase size of instance
(RAM, CPU etc.)



HORIZONTAL SCALING

(Add more instances)



Elasticity Mechanisms and Overhead

MECHANISMS

Horizontal Scaling of VNF instance

- Installing a new VNF instance
- Removing an existing VNF instance

Migration of a VNF instance

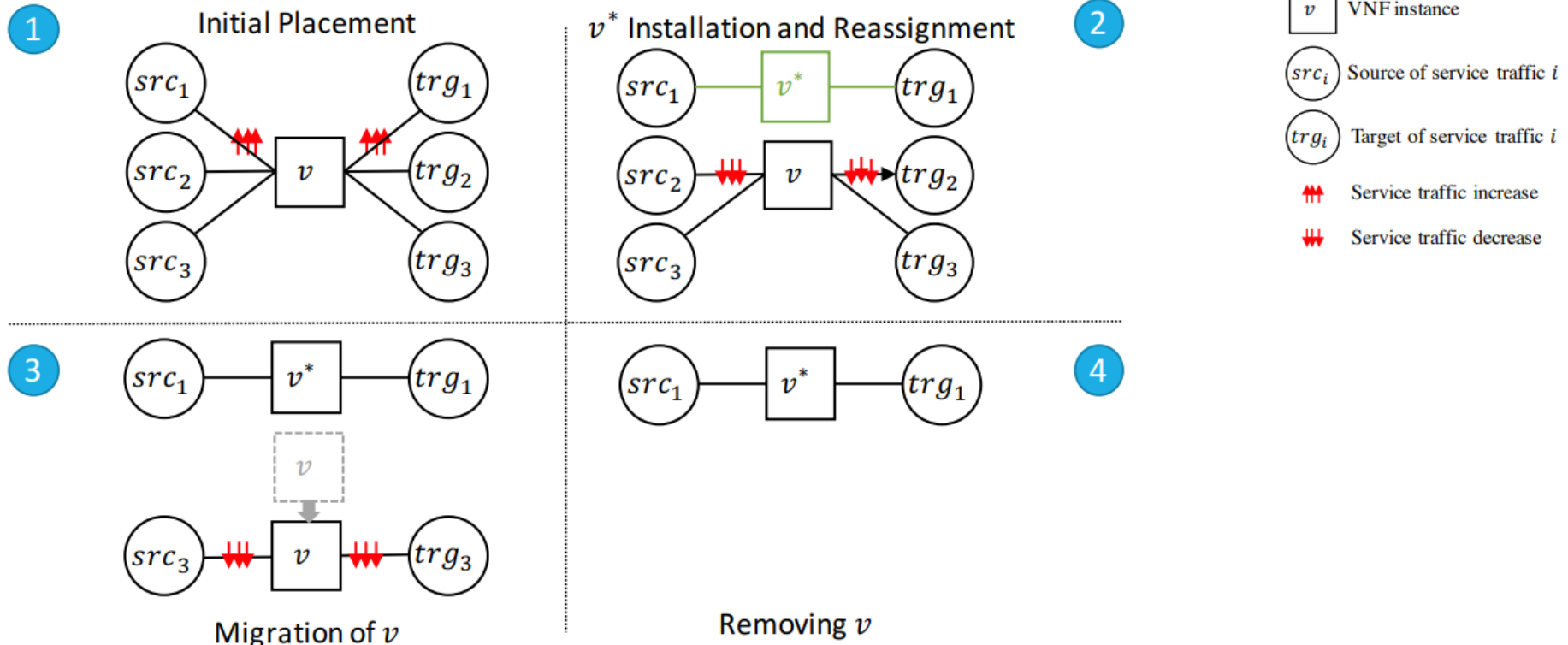
Reassignment of workload to another VNF instance

OVERHEAD

Migration overhead

Reassignment overhead

Elasticity Mechanisms and Overhead



Optimal placement is NP-Hard

- The optimal placement of VNFs is known to be an **NP-hard problem**.
- Optimizing both network and computing resources at the same time may make the problem **difficult to solve in a reasonable period of time**.
- The service chain embedding problem is an NP-hard problem; therefore, **finding an optimal solution is not viable due to the large number of requests processed in the production environment**. Hence, we propose heuristics to solve this problem and explore potential solutions.

Optimal placement is NP-Hard

- Related works can be classified into four main categories:
 - A **fixed number of middleboxes** are assumed to be already deployed in the network, and the optimal solution attempts to find for each traffic flow the optimal routes through them.
 - A set of **pre-defined routes is assumed** and the optimal solution is found for placing the VNFs within static routes.
 - The third category tackles these two objectives separately. The placement objectives are prioritized and then run sequentially
 - In the last category, both objectives are considered simultaneously to tackle the optimal placement.