

# Minimize Resource Consumption in a Upgrade aware SFC Topology design and Mapping

Kaushik Koneru

Department of Computer Science  
Georgia State University  
Atlanta, Georgia 30302  
Email: kkoneru1@student.gsu.edu

Shakeel Osmani

Department of Computer Science  
Georgia State University  
Atlanta, Georgia 30302  
Email: sosmani1@student.gsu.edu

**Abstract**—Recent Development in Network function virtualization technology has led to replacement of physical equipments in a telecom network with software based virtual network appliances. This inturn ensures easier sharing of network functions as the whole network can be administered from a centralized data center also called as Telecom Clouds(TCs). These TCs represents Virtual Network Appliances (VNAs) in a Virtual Network (VN) to honour end user service requests. The set of VNAs with Virtual Link interconnect is called a Service Function Chain (SFC). This NFV Technology enables flexible allocation of resources as per requirement which are known as SFC Request. The Topology of this SFC can be represented as simply as linear array to as complex as dense network. In this Project, we discuss of designing SFC Topology and Mapping in such a way to minimize Total Resource Consumption(TRC). Resource in a SFC include Link State information like bandwidth, cost etc., and Virtual Node information like Computing Cycles etc.

**Keywords**—SFC, NFV, SF Request, Unicast Traffic, TRC.

## I. INTRODUCTION

Now-a-days NFV being implemented in industries, Optimizing resources while mapping from virtual Network to substrate network been one of the main criteria. In this paper we discuss on how we need to optimize the resources like bandwidth across the link and CPU capacity and memory in nodes. Instead of optimizing resources while mapping we design virtual network from scratch or redesign virtual network by optimizing the bandwidth and then map by satisfying node and link constraints. Since complete optimized solution is a NP-hard problem. This project tries to establish a solution which tries to minimize the resource utilization to the maximum possible extent under the given constraints. There are few assumptions and challenges in this design and mapping problem which are discussed in detailed below.

### A. Assumption

- **Online Case:** This is a Online Scenario where request are expected to come on fly.
- **Bi-Directional Links:** Each link between the nodes are capable of sending traffic from Node  $i$  to Node  $j$  as well as from Node  $j$  to Node  $i$ . So, Every links in the topology we design are expected to be bi-direction.
- **Virtual Network Design:** Virtual Network we are designing in this project is from scratch. We will be

receiving virtual SFC request on fly and design virtual network accordingly which will be mapped later on

- **Request Life time:** Every Virtual Request life time is assumed to have very long life time. So, we ignore reusing the bandwidth allocated for some other request.
- **Cores:** We Assume every core in the every physical nodes are equally capable. In general CPU in Nodes are consumed of Intel/Arm CPU. In This project we assume that every CPU are off same capacity so every CPU may be of 2Ghz.
- **Bandwidth Sharing:** Bandwidth assigned in process are capable of sharing to other requirement as well.

### B. Virtual SFC Request

We receive the Virtual SFC Request in a particular format. These Virtual SFC requests can be requesting new service, new node, new bandwidth allocation, bandwidth upgrade. In This project we are only concern about the virtual SFC request holding bandwidth request. These virtual SFC request will be of format {node1, node2, 10MB}. This request can be decoded as 10 MB of bandwidth is requested between node1 and node2. This request can be either upgrade of the existing or new connection, We determine while designing virtual network.

### C. Challenges

*1) Bandwidth Allocation:* : Bandwidth allocation is the main theme in this project, where we considered all the bandwidth in the network can be shared among all the nodes. Basing on the virtual SFC request we design and allocate the bandwidth among the nodes. In this there are 3 cases

- **Bandwidth Request:** With Bandwidth virtual SFC request, we analyze the present existing virtual network. We optimize bandwidth allocation by considering maximum flow between the nodes in virtual SFC request. As mentioned in Fig 1, we analyze the maximum flow and assign required bandwidth between the nodes.

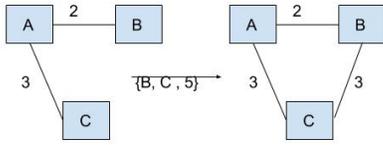


Fig 1

- **New Node Request:** In Bandwidth virtual SFC request, we receive a request where complete new nodes are involved then we create new nodes in the virtual network and assign the requested bandwidth. If two nodes are new nodes then we create a complete disjoint virtual network and design that network.

## II. RELATED WORK

Virtual Network resource allocation (and embedding) has attracted a lot of attention from the research community during the last few years. Several optimization approaches were proposed, some follow an integrated strategy considering simultaneously the resource embedding of virtual nodes and links and, most, follow a split approach that considers separately and successively the two resource embedding problems. Whatever the approach, virtual resource allocation is the core component of any virtual network resource embedding method. Network service deployments are often coupled to network topology, whether it be physical or virtualized, or a hybrid of the two. Such dependency imposes constraints on the service delivery, potentially inhibiting the network operator from optimally utilizing service resources, and reduces the flexibility. This limits scale, capacity, and redundancy across network resources.

Service Function Chaining aims to address the aforementioned problems associated with service deployment. SFC mainly investigates the solution mentioned here are 1.) Service overlay, This provides SF connectivity and is built on top of existing network topology. This as well allows operators to use overlay they prefer to create a path between SF, and to locate SF in the network as required. This is an effective means to address issues like QoS, Bandwidth provisioning. 2.) Service Classification, This is used to prioritize the traffic or Service request. Using this traffic enters a service overlay. This classification varies based on different capabilities like node, link and service offered. Initial classification determines the service function chain required to process the traffic. Subsequent classification can be used within a given service function chain to alter the sequence of service functions applied. 3.) Dataplane Metadata, this provides the ability to exchange information between logical classification points and service functions and between service functions. As such metadata is not used as forwarding information to deliver packets along the service overlay.

Virtual Link resource allocation can be classified basing upon two criteria point-to-point and point-to-multipoint network. In this paper we discuss about point-to-point allocation of resources. This resources can vary from Bandwidth to many other link state

we would like to add a note on the relation between the network service function allocation and chaining and the Virtual Network Embedding (VNE)[1]. Despite some similarities, solution is not the same, As in VNE it is mostly deals with mapping each virtual node and virtual link to a physical entities where as Service Function Chaining Mapping we map a NFV to the physical network.

## III. SFC DESIGN AND MAPPING PROBLEM

### A. Problem Overview

SFC is a set of virtual network appliances which are interconnected to each other where each and every node (VNA) are dedicated to perform single or multiple functions having their own constraints like bandwidth, node memory, CPU capacity. In this paper we will be discussing about SFC topology design with minimizing resources consumption like bandwidth, memory or CPU and enabling the mapping to substrate network. In a Service Function Chain topology design mainly consists three parts 1.Placement 2. Assignment 3.Chaining. In Placement, we considered more about service request requirements like how many service functions are required to honor the request and place them in the infrastructure. In Assignment, in placed virtual network function instances resources are assigned based on service request like cores, memory, bandwidth on virtual links. In Chaining, Paths are created between end-to-end nodes with specific bandwidth as per SFC request. SFC Mapping, is all together a new function where we need to form the connection between virtual and physical set of VNAs.

### B. Problem Definition

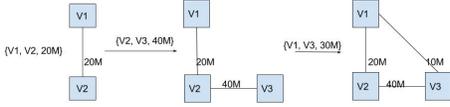
1) *Topology Design Problem:* This is an online case where virtual Service Function Chaining(SFC) request need to be honoured by creating a virtual network and then mapping to substrate network where objective is to minimizing usage of resources like bandwidth in links and CPU and Memory in Nodes. The service function chaining request can consist of any virtual network request like creating a new virtual network with specified set of service functions like CSCF, MME, Firewall, etc.,. In general these request are in structural format consists of the nodes and request. In this paper we are concerned only on the bandwidth SFC request. These request are in structural format consists of two nodes and bandwidth between them. Using these request we design virtual network from scratch. This Network Function Virtualization(NFV) infrastructure topology with each SFC can be represented as undirected graph  $G_v = (N_v, L_v)$ , where  $N_v$  is the list of the virtual nodes in topology s.t  $N_v = N_1, N_2, N_3, \dots, N_n$  and  $L_v$  is the list of all the virtual links in topology s.t  $L_v = L_1, L_2, L_3, \dots, L_n$ . Here  $|N_v|$  gives total number of virtual nodes and  $|L_v|$  gives the total number of virtual links used in the virtual topology. The performance of such virtual network depends upon VNAs(Virtual Network Appliances) and virtual link capacities. Performance of VNAs and link capacities can be increased by increasing resources. However, this is not always a feasible solution and it suffers from Moore's Law. Thus optimization of resources in such a scenario will yield cost effective as well as scalable solution. Telecom Clouds leverage the concept of shared virtual resources to maximize

performance. As the number of requests for SFC grow substantially the progressive requirement for more number of VNAs and links go up exponentially. The optimization of this problem is a NP-Hard problem and can be mathematically modelled as a minimization problem of the following components.

### Objective

Let  $L_i$  is the Virtual Link,  $B_i$  is the Bandwidth on Virtual Link  $L_i$ ,  $C_i$  is the Cost of the Virtual Link  $L_i$  and  $N_i$  is number of VNAs.

$$\min\left(\sum_{i=1}^n B_i\right) \quad // \text{minimize Bandwidth} \quad (1)$$



2) *SFC Node Mapping*: Once the resources have been assigned to the respective VNAs in the NFV topology, this needs to be mapped with substrate network, This is known as SFC mapping. In this mapping we need to follow the constraint that each and every Virtual Node can map only to a single substrate network node but whereas a substrate network node can be represented by multiple virtual network nodes.  $M(N_i^s)$  is the available Memory in memory on Substrate Node  $N_i$ ,  $M(N_i^v)$  is the available memory in bytes on Virtual Node  $N_i$ .  $C(N_i^s)$  is the Computation Power on Substrate node,  $C(N_i^v)$  is the available computational power on Virtual node.  $N_i$  is mapped to the substrate node which satisfies the memory and computational power Constraints and among all the nodes which satisfies Bandwidth and CPU substrate node with optimum Computing power and Memory is assigned. This optimum is obtained by using equation 1. In this  $\alpha_1$ ,  $\alpha_2$  are assigned basing on the cost or scarcity of available resources. If the requirement is to decrease the cost on Computational power  $\alpha_1$  will be less and  $\alpha_2$  is more relatively. If we want to decrease the cost on Memory on Nodes then  $\alpha_1$  will be more and  $\alpha_2$  is less relatively. In Node Mapping Algorithm we have used a variable NF which contains a boolean variable if substrate node has been found for a virtual node  $N_i$ . As well we have used function EQ which will validates along with the equation 1, mentioned below.

- Constraints

$$NF(N_i) = \begin{cases} 0 & \text{if SubstrateNodeisFoundfor } N_i \\ 1 & \text{if SubstrateNodeisnotFoundfor } N_i \end{cases}$$

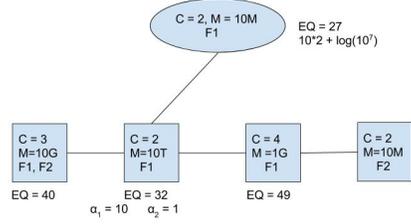
$$EQ \rightarrow (\alpha_1 * B(N_s) + \alpha_2 * M(N_s)) \quad (2)$$

- While node mapping below objective needs to be obtained

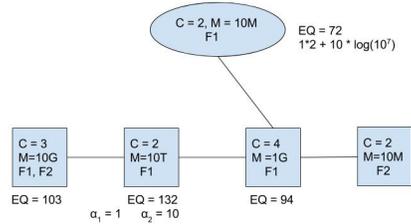
$$\min(EQ)$$

In Node Mapping, we have two scenarios where CPU is prioritized and needs to be optimized in using compared to memory then  $\alpha_1$  value should be more than  $\alpha_2$ . In Below example, we have considered  $\alpha_1$  is equal to 10 and  $\alpha_2$  equal

to 1. These values are selected in such a way that maximum value of the other variable can be compromised by these constant values.



Other scenario is memory is prioritized and needs to be optimized while mapping then  $\alpha_2$  value should be more than  $\alpha_1$ . In Below example, we have considered  $\alpha_2$  is equal to 10 and  $\alpha_1$  equal to 1.



3) *Link Mapping*: In Link Mapping we assign the link with k-shortest path which satisfies the bandwidth requirement between  $N_i$  and  $N_j$ . Link Mapping is done later Node Mapping where every virtual node  $N_i$  is mapped to Substrate node.  $E_{ij}$  is the connection between  $N_i$  and  $N_j$  in virtual when we map to substrate network we assign the link state information to the link between a selected substrate link connecting the substrate node mapped to  $N_i$  and  $N_j$ . All the link state information like bandwidth, connection type are mapped to the substrate network. In case the bandwidth is not satisfied then we deny the service request during mapping.

### C. Heuristics approach

- The first step in the problem solving is to setup a Multi-Function Telecom Cloud(TC). we set up a TC requires the following minimum input parameters, (i)name, (ii)id, (iii)dataCenterCharacteristics This contains properties like, node numbers, bandwidth available etc, (iv)VmAllocationPolicy which holds information like VM allocation algorithm, e.g FCFS, Round-Robin etc.
- In New Service Function Algorithm we analyze the service request received, From Service Request we retrieve memory required, Node details, Disk Space required and node disk space capacity.
- we compare the  $N_i$  node capacities satisfies the Service Request, If it does we assign the SR to that node. To check that we have used multiple functions. CheckCompatibility, this functions returns a boolean variable basing on upon  $N_i$  and service function feature needs to be added on to the node.
- The next method is the isNodeMultiFunction which handles allocation of features, In general TC clouds

can be either Single Function or Multi Function. If it is Single Function they only supports one service function, whereas in Multi Function they supports multiple functions. So, we check the function can be added to the existing function or is it not capable of serving any more request.

- The newServiceMapping method processes a new user service request. It checks for the  $M_{req}$  (Memory needed to service the request),  $R_{req}$  (Disk space required for the request along with the  $flag_{comp}$  (compatibility flag).

---

### Algorithm 1 Bandwidth Allocation

---

```

1: procedure BANDWIDTHALLOCATION
2:   for each USR Bandwidth Request  $R_i$  do
3:      $(N_i, N_j) \leftarrow getNodes(R_i)$ 
4:      $B_{ij} \leftarrow getBandwidth(R_i)$ 
5:     NodePresent  $\leftarrow 1$ 
6:     if  $N_i$  not in VNQueue then
7:       createVirtualNode( $N_i$ )
8:       enqueue( $N_i$ , vnQueue);
9:       enqueue( $N_i$ , unMappedQueue);
10:      NodePresent  $\leftarrow 0$ 
11:    end if
12:    if  $N_j$  not in VNQueue then
13:      createVirtualNode( $N_j$ )
14:      enqueue( $N_j$ , vnQueue);
15:      enqueue( $N_j$ , unMappedQueue);
16:      NodePresent  $\leftarrow 0$ 
17:    end if
18:    if NodePresent then
19:      if PathExistBetween ( $N_i, N_j$ ) then
20:         $b_{ij} \leftarrow MAXFLOW(N_i, N_j)$ 
21:        if  $b_{ij} < B_{ij}$  then
22:           $b_{rem} \leftarrow (B_{ij} - b_{ij})$ 
23:          CreateNewPath( $N_i, N_j, b_{rem}$ )
24:          unMappedLinkQueue( $N_i, N_j$ )
25:          enqueue(unMappedLinkQueue,
 $N_i, N_j, B_{rem}$ )
26:        else
27:          return request
28:        end if
29:      else
30:        CreateNewPath( $N_i, N_j, B_{ij}$ )
31:        enqueue(unMappedLinkQueue,  $N_i, N_j, B_{ij}$ )
32:      end if
33:    else
34:      CreateNewPath( $N_i, N_j, B_{ij}$ )
35:      enqueue(unMappedLinkQueue,  $N_i, N_j, B_{ij}$ )
36:    end if
37:  end for
38: end procedure

```

---

## IV. PSEUDO CODE

### A. Telecom Cloud Setup

Telecom cloud includes components like data center, Buffer pool, Application Queue, Base Stations etc., these components can be categorized in the following manner. 1.) Network Component: This includes two sub components  $N_{RBS}$  and

$D_{net}$  where  $N_{RBS}$  is the number of base stations in the network and  $D_{net}$  is cumulative network delay. Base stations is the physical receiving component and cumulative delay is the total delay starting from request arrival into buffer pool till it is delivered to the base station.

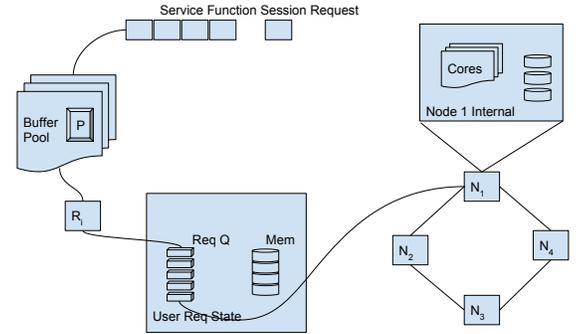
2.) Data Center: This has three components  $N_{DC}$  is the number of data centers,  $N_s^i$  which is number of servers in data center.  $N_{cpu}$  is the number of cores present in each server.

1) *Buffer Pool*: Buffer Pool is a temporary memory which is having processor "P". This processor can compute  $N_{ACR}$  which is a total number of services at a given session. Further it can calculate arrival rate which is  $R_{arr}$  and also read the request size which is  $R_{size}$ .

Buffer pool checks the liability of request which are forwarded to the VNA to hold it or deny.

2) *Application Queue*: This Component processes the forwarded request using main memory and cache. Also it stores user request state which is required for analyzing the service comparison flag, so that appropriate mapping can be done.

3) *Physical Disks*: This component stores the request till the life time of a request it is usually parallelized array of independent disks in telecom clouds.



### B. Virtual Network Bandwidth Allocation

In This scheme we analyze the input virtual SFC request and allocate the bandwidth in the virtual network as discussed. In this Algorithm, we create two queue vnQueue and unMappedQueue. Where vnQueue consists of the nodes present in virtual network whereas unMappedQueue consists of list of nodes which are not yet mapped to the substrate network. So, As the virtual SFC Request arrives we look into the request nodes, If any of these nodes are not present in vnQueue, we create a new node and enqueue it in both vnQueue and unMappedQueue. After checking the node presence we check for the link existence between these nodes, If present we check add requested bandwidth  $B_{ij}$  onto the link, If not present we check the maximum flow between these nodes and store in  $b_{ij}$  if  $b_{ij}$  is less than  $B_{ij}$  then we create a new link between these nodes with bandwidth of difference between  $b_{ij}$  and  $B_{ij}$  and enqueue this link into unMappedLinkQueue. This queue is used to map all the unmapped link from virtual link to substrate link. Once mapped these links and nodes are removed from unMappedLinkQueue and unMappedQueue.

### C. SFC Mapping

As we design virtual network using virtual SFC request we create two queues unMappedQueue and unMappedLinkQueue which consists of nodes which are unmapped from virtual to substrate and virtual links which are not yet mapped respectively. Using these queue we process node mapping and link mapping. In node Mapping we consider multiple constraints like service function, CPU capacity , Memory capacity. we generate the EQ value for every node in virtual and substrate network and try to match to the one which is having least satisfying value. After Mapping the nodes we map the links between these nodes using k-shortest mechanism, Which means the minimum path which is satisfying the required bandwidth between the nodes is mapped from virtual to substrate link. While node Mapping we have prioritized the Single function rather than multi-function substrate nodes.

---

#### Algorithm 2 SFC Mapping

---

```

1: procedure NODEMAPPING ▷ Greedy Algo
2:   NSorted  $\leftarrow$  sort $N_i^s$  basing on EQ Value and Total
   Service function
3:   for  $N_i$  in unMappedQueue do
4:      $NF(N_i) \leftarrow 0$ 
5:     for  $N_s$  in NSorted do
6:       if SFCCompare ( $N_i, N_s$ ) then
7:         if  $C(N_i) \leq C(N_s)$  &  $M(N_i) \leq M(N_s)$  then
8:           if  $EQ(N_s) \leq EQ(N_i)$  then
9:              $NF(N_i) \leftarrow 1$ 
10:             $N_s \leftarrow N_i$ 
11:            break;
12:          end if
13:        end if
14:      end if
15:    end for
16:    if  $NF(N_i)$  then
17:      dequeue( $N_i$  , unMappedQueue)
18:    else
19:      return Failure
20:    end if
21:  end for
22: end procedure
23:
24: procedure LINKMAPPING ▷ k-Shortest Path
25:  for  $E_{ij}$  in unMappedLinkQueue do
26:    ( $N_i, N_j \leftarrow$  getVirtualNodes( $E_{ij}$ ))
27:     $B_{ij} \leftarrow$  getBandwidth( $E_{ij}$ )
28:     $N_i^s \leftarrow$  getSubstrateNode( $N_i$ )
29:     $N_j^s \leftarrow$  getSubstrateNode( $N_j$ )
30:     $SE_{ij}^s \leftarrow$  sortLinks( $N_i^s, N_j^s$ )
31:    for  $E_s$  in  $SE_{ij}^s$  do
32:      if  $B_{ij} \leq$  Bandwidth( $E_s$ ) then
33:         $E_s \leftarrow E_{ij}$ 
34:        dequeue(unMappedLinkQueue,
35:         $N_i, N_j, B_{ij}$ )
36:        break;
37:      end if
38:    end for
39:  end for
40: end procedure

```

---

### REFERENCES

- [1] M. Chowdhury, M. R. Rahman, and R. Boutaba, *Vineyard: Virtual network embedding algorithms with coordinated node and link mapping*, *IEEE/ACM Transactions on Networking*.
- [2] <ftp://ftp.bme.hu/documents/internet-drafts/draft-boucadair-sfc-design-analysis-02.txt>
- [3] Paul Quinn and Jim Guichard, Cisco Systems "Service Function Chaining: Creating a Service Plane via Network Service Headers"
- [4] Marcelo Caggiani Luizelli, Leonardo Richter Bays, Luciana Salete Buriol "Piecing Together the NFV Provisioning Puzzle: Efficient Placement and Chaining of Virtual Network Functions"
- [5] Mikael Capelle, Slim Abdellatif, Marie-Jose Huguet, and Pascal Berthou "Online Virtual Links Resource Allocation in Software-Defined Networks"
- [6] Xiang Cheng, Sen Su, Zhongbao Zhang, Yan Luo, Jie Wang "Virtual Network Embedding Through Topology-Aware Node Ranking"
- [7] Marcio Melo, Susana Sargento, Ulrich Killat, Andreas Timm-Giel, and Jorge Carapinha "Optimal Virtual Network Embedding: Node-Link Formulation"