Mathematical formulations

Mandar Deshpande - 210010014

November 28, 2024

1 Parameters and Variables

1.1 Parameters

- $G(V, E)$: Physical network consisting of compute nodes connected together
- $C1(v)$: Compute resource at $v \in V$
- $M1(v)$: Memory resource at $v \in V$
- $B1(e)$: Bandwidth for $e \in E$
- W: Set of VNFs $\{VNF_1, VNF_2, VNF_3, \dots\}$.
- $C2(VNF_i)$: Compute resource requirement for VNF_i
- $M2(VNF_i)$: Memory resource requirement for VNF_i
- Y: Set of SFCs $\{SFC_1, SFC_2, SFC_3, \dots\}$.
- SFC_i: Ordered set of VNFs
- $B2(SFC_i)$: Bandwidth required by SFC_i
- $D(SFC_i)$: delay threshold for SFC_i
- $O(SFC_i)$: originating node for SFC_i

1.2 Decision Variables

$$x^v_{VNF_i} = \begin{cases} 1, & \text{if VNF}_i \text{ is hosted at } v \in V, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{SFC_i} = \begin{cases} 1, & \text{if SFC}_i \text{ is fully provisioned ,} \\ 0, & \text{otherwise.} \end{cases}$$

2 Constraints

2.1 VNF to Node mapping

One VNF can be provisioned on atmost one compute node

$$\sum_{v \in V} x^v_{VNF_i} \leq 1 \quad \forall VNF_i \in W$$

2.2 Compute Capacity

Sum of compute requirements of all VNFs on a given node should be less than or equal to compute capacity of that node

$$\sum_{VNF_i \in W} x^v_{VNF_i} \cdot C_2(\text{VNF}_i) \leq C_1(v), \quad \forall v \in V$$

2.3 Memory Capacity

Sum of memory requirements of all VNFs on a given node should be less than or equal to memory capacity of that node

$$\sum_{VNF_i \in W} x^v_{VNF_i} \cdot M_2(\mathrm{VNF}_i) \leq M_1(v), \quad \forall v \in V$$

2.4 SFC Provisioning

The SFC is provisioned if and only if all its VNFs are provisioned

$$Y_{SFC_i} = 1 \iff \left(\sum_{VNF_j \in SFC_i, v \in V} x^v_{VNF_j} \right) = |SFC_i|, \quad \forall SFC_i \in Y$$

2.5 Originating node

If the SFC is provisioned then first VNF and last VNF of that SFC should be placed at O_{SFC_i}

$$\left(Y_{SFC_i} = 1 \implies x^{O(SFC_i)}_{SFC[j]} = 1, \quad \forall j \in \{1, |SFC_i|\} \right), \quad \forall SFC_i \in Y$$

Notations in this equation are slightly confusing. Note that $SFC[j]$ is j^{th} VNF of that chain and SFC_i is i^{th} SFC

2.6 Bandwidth limit

For every link, bandwidth of that link should be more than sum of bandwidths of SFC using that link. According to the formulation, packet should got to VNF_{j+1} just after VNF_j. i.e. VNF_{j+1} comes just after VNF_j in service function chain

$$B_1(e) \geq \sum_{SFC_i \in Y} \left(Y_i \cdot \sum_{\substack{(VNF_j, VNF_{j+1}) \in SFC_i \\ v_1 \in V, v_2 \in V}} x^{v_1}_{VNF_j} \cdot x^{v_2}_{VNF_{j+1}} \cdot f(e, v_1, v_2) \cdot B_2(SFC_i) \right), \quad \forall e \in E$$

$$f(e, v_1, v_2) = \begin{cases} 1, & \text{if } e \text{ is an edge in the shortest path from } v_1 \text{ to } v_2, \\ 0, & \text{otherwise.} \end{cases}$$

2.7 SFC Delay threshold

For every provisioned SFC, sum of link latencies along shortest path should be less than delay threshold of that SFC.
According to the formulation, packet should got to VNF_{j+1} just after VNF_j. i.e. VNF_{j+1} comes just after VNF_j in service function chain

$$D(SFC_i) \leq \sum_{\substack{(VNF_j, VNF_{j+1}) \in SFC_i \\ v_1 \in V, v_2 \in V}} Y_{SFC_i} \cdot x^{v_1}_{VNF_j} \cdot x^{v_2}_{VNF_{j+1}} \cdot g(v1, v2)$$

$$g(v_1, v_2) = \text{length of shortest path from v1 to v2 when we use latency as link weight}$$

3 Objective

Maximize the number of network services that can be provisioned over the given network.

$$max \quad \sum_{SFC_i \in Y} y_{SFC_i}$$