Resource Management in Service Chaining

draft-irtf-nfvr-g-resource-management-service-chain-02

Seungik Lee (ETRI)
Sangheon Pack (Korea Univ.)
Myung-Ki Shin (ETRI)
EunKyoung Paik (KT)
Rory Browne (Intel)
Recall

• Problems
  – VNF placement/scheduling in building/maintaining service chains to satisfy given policies

• Use cases
  – path optimization, load balancing, redundancy, traffic optimization, energy efficiency

• Goals
  – build a framework, algorithms, contributions to SFC
Changes since IETF-93

• -02
  – added an evaluation model
  – as a new section #5
Evaluation Model

• Objective
  – determine optimal service chains for the use cases

• Key considerations
  – traffic processing capacity of a VNF instance
  – amount of traffic passed on a VL instance

• System models
  – VNF placement
  – flow distribution ratio

• Objective functions
  – throughput optimization
  – load balancing
VNF Placement

- Indicator function $v_{(i,k,n)}^s$ for VNF placement (VPIF)
  - If $v_{(i,k,n)}^s = 1$, function $s$ is installed on $n$th VM with class $k$ in SN $i$
  - $N_{(i,k)}^s = \sum_n v_{(i,k,n)}^s$: the number of VNF instances where function $s$ is installed with class $k$ in SN $i$

![Diagram](image)

- $N_{(1,1)}^1 = 1$
- $N_{(1,1)}^3 = 1$
- $N_{(1,2)}^2 = 2$
- $N_{(1,1)}^3 = 1$
Flow Distribution Ratio

- Traffic flow distribution ratio among NFPs (TFR)
  - $f^s_{(i,j)}$: flow ratio that passes link $(i, j)$ and is already processed by function $s$
  - The amount of flows assigned to link $(i, j)$ for function $s$: $F^c_df^s_{(i,j)}$

$F^c_d$: CPU demand for processing the flow (or Flow rate)
**Objective Functions**

- **Throughput optimization**

\[
\max_{(f,v)} \sum_{(i,j) \in E} \sum_{s \in F} f_{(i,j)}^s C(i,j) + \sum_{s \in F} \sum_{k \in H} \sum_{i \in V_{SN}} \sum_{n} f_{(i,k,n)}^s c_k
\]

Throughput for VL

Throughput for VM (VNF)

- **Load balancing for VNF**

\[
\max_{(f,v)} \left( \min_{(i,k,n)} c_k - F_d^c f_{(i,k,n)}^s \right), s \in F, i \in V_{SN}, k \in H
\]

Remaining CPU capacity for each VNF instance (i.e., VM)

- **Load balancing for Virtual Link**

\[
\max_{(f,v)} \left( \min_{(i,j) \in E} C(i,j) - F_d^c f_{(i,j)}^s \right), s \in F
\]

Remaining capacity for each link
Next Steps

• Build a framework and heuristic algorithms for prototyping
• Merging I-Ds for “Policy-based Resource Management”
  – sub-topics:
    • policy, service chains, use cases (reliability), orchestrations
  – relevant I-Ds
    • draft-irtf-nfvg-nfv-policy-arch
    • draft-krishnan-nfvg-policy-based-rm-nfviaas
    • draft-irtf-nfvg-resource-management-service-chain
    • draft-bernini-nfvg-vnf-orchestration
    • draft-felix-nfvg-recursive-orchestration
    • and any others?
  – → needs further discussion