Optimizing Flow Bandwidth Consumption with Traffic-diminishing Middlebox Placement

Yang Chen, Jie Wu, and Bo Ji

Center for Networked Computing

Temple University, USA

VNF: Evolution of Network Service

- Network Function Virtualization (NFV)
  - Virtualizing network functions into software building blocks
- Virtualized Network Function (VNF) or Middlebox
  - Software implementation of network functions
  - Improve performance & enhance security
- Examples

Proxy

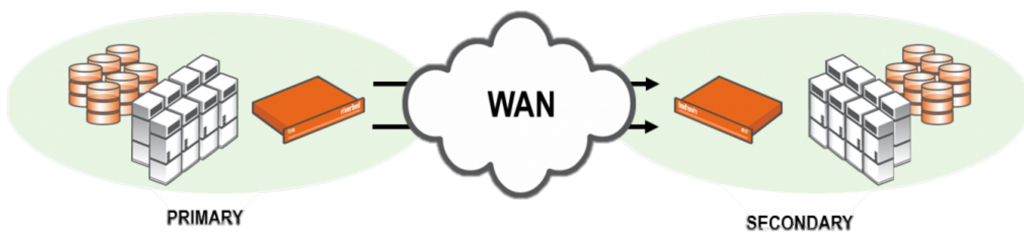
Firewall

NAT

- Middlebox Deployment
  - Deployment location selection on multiple servers

VNF Traffic Changing Effects [1]

- VNFs may change flow rates in different ways
  - Citrix CloudBridge WAN accelerator: 20% (diminishing)



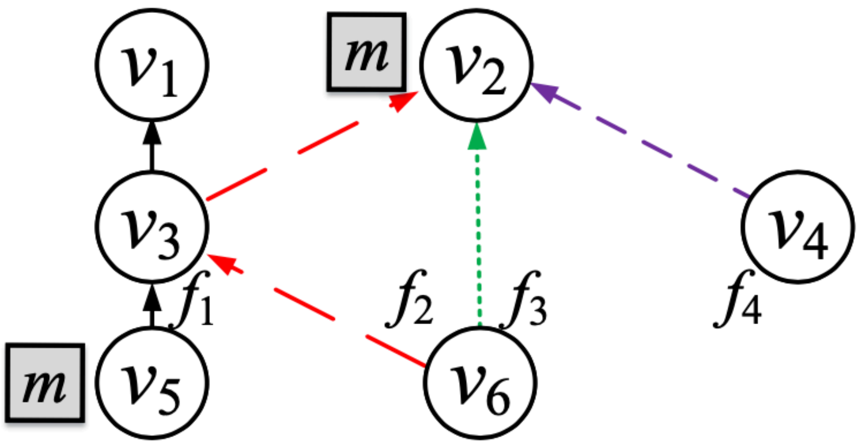
  - BCH(63,48) encoder: 130% (expanding)

[1] Traffic Aware Placement of Interdependent NFV Middleboxes (INFOCOM '17)

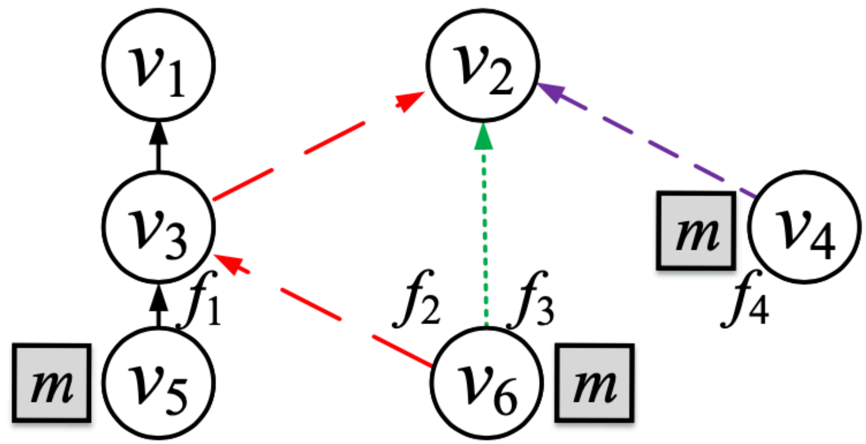
A motivating example

Traffic-diminishing ratio of VNF m: 0.5

Initial flow rate: f_1 (4), f_2 (2), f_3 (2), f_4 (2)



(a) Two middleboxes.



(b) Three middleboxes.

0.5*4*2+2*2+2+2=12

0.5*4*2+ 0.5* 2*2+ 0.5* 2+ 0.5* 2=8

Total bandwidth consumption

2. Our model

- Problem
  - Deploy a single type of VNFs with traffic-diminishing effect into the network
- Objective
  - Minimize total bandwidth consumption of all flows on all links along their paths
- Constraint
  - Each flow gets processed
  - Deploy a limited number of the single type of VNFs

3. Problem Formulation

A mathematical optimization problem on minimizing total flow bandwidth consumption

$$\min_{\{m_v, f_v | v \in V\}} b(\mathcal{P}, \mathcal{F}) = \sum_{f \in F} b(f) = \sum_{f_v = 1, f \in F} r_f(|p_f| - (1 - \lambda) l_v(f)) \quad (1)$$

Single flow

$$\text{s.t. } \mathcal{P} = \{v \mid m_v = 1, \forall v \in V\} \quad (2)$$

$$|\mathcal{P}| = \sum_{v \in V} m_v \leq k \qquad \forall m \in M \quad (3)$$

$$\sum_{v \in p_f} f_v = 1 \qquad \forall f \in F \quad (4)$$

$$f_v \leq m_v, \ \mathcal{F} = \{f_v | \forall f \in F\} \qquad \forall v \in V \quad (5)$$

$$m_v = \{0, 1\}, \ f_v = \{0, 1\} \qquad \forall v \in V \quad (6)$$

4. Solution for general topologies

- NP-hard
- Decrement function $d(\mathcal{P}) = \sum_{f \in F} r_f \cdot |p_f| - b(\mathcal{P})$
  - Decrement of total bandwidth consumption compared to no VNFs
- Marginal decrement $d_{\mathcal{P}}(\mathcal{S}) = d(\mathcal{P} \cup \mathcal{S}) - d(\mathcal{P})$
  - Additional bandwidth decrement by deploying on $\mathcal{S}$ beyond $\mathcal{P}$
- Decrement function is submodular
  - More VNFs, less bandwidth consumption
  - Flow gets processed no later than $\mathcal{P}$

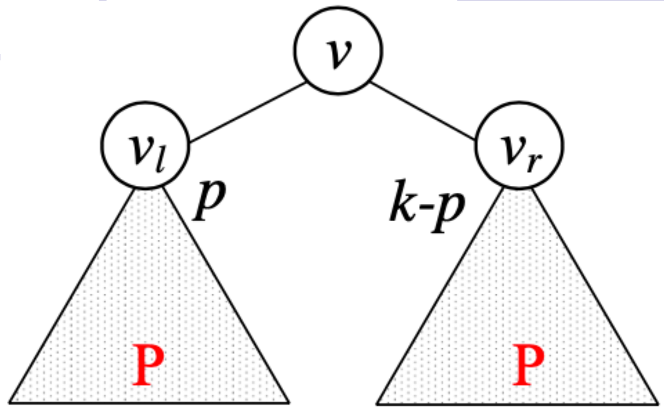
4. Solution for general topologies (cont'd)

- Solution
  - General Topology Placement (GTP)
- Steps
  - Iteratively select $v \in V$ with the maximum marginal decrement until all flows are fully served
- Approximation ratio $1 - \frac{1}{e}$
- Time complexity (|V|: #vertices)
  - $O(|V|^2 \log |V|)$

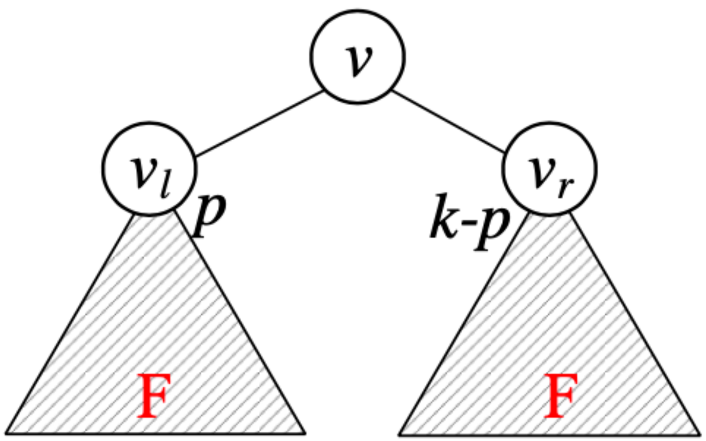
5. Two solutions for trees
Solution 1: Dynamic Programming (DP)

- $F(v, k)$
  - Minimum total occupied bandwidth of all flows with k deployed middleboxes in subtree T_v rooted at v
  - All flows get fully processed in T_v
- $P(v, k, b)$
  - Same as F(v,k)
  - When flows with only a total bandwidth b processed
- Optimal solution
- Time complexity (|V|: #node, r_{max}: largest flow rate)
  - $O(|V|\,(\log|V|)^3 r_{max})$

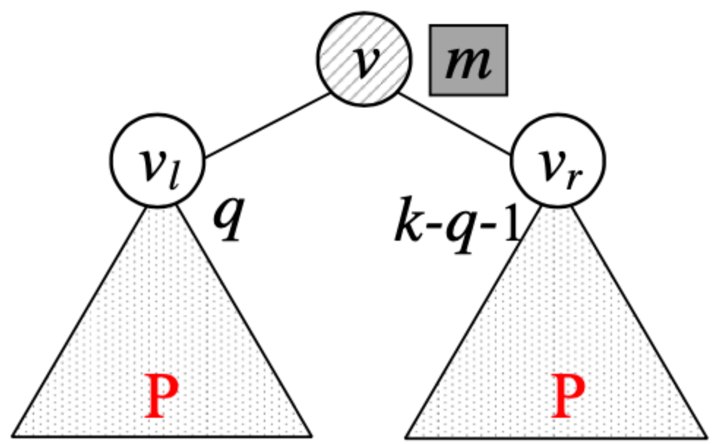
Solution 1: Dynamic Programming (DP)



Partially processed



(a) Subtree fully processed



(b) Processed on v

Fully processed

Solution 2: Heuristic Algorithm for Trees (HAT)

- Lowest Common Ancestor (LCA)
  - LCA(v,w): lowest vertex have both v and w as descendants
- Steps
  - Deploy one VNF on each leaf vertex
  - Delete two VNFs on v and w with minimum difference of the total bandwidth value
  - Place one VNF on LCA(v,w)
  - Until total number of deployed VNFs no more than k

4. Solution for trees (cont'd)

- Maintenance of all difference values
  - Min-heap
  - Improve time efficiency
- Time complexity
  - $O(|V|^2 \log |V|)$
  - |V|: #vertices

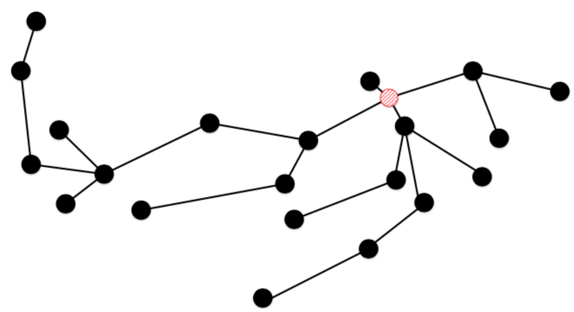
7. Simulation

- Comparison algorithms
  - Random
    - Randomly deploy k VNFs
  - Best-effort
    - Deploy on the vertex, which can reduce the total bandwidth of flows most, until k VNFs are deployed
- Our proposed algorithms
  - General topo
    - Alg. GTP
  - Tree topo
    - Algs. GTP, DP, HAT

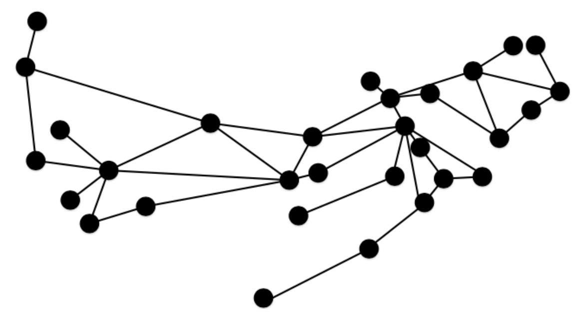
Settings

- Topology

(a) The Archipelago (Ark) Infrastructure.



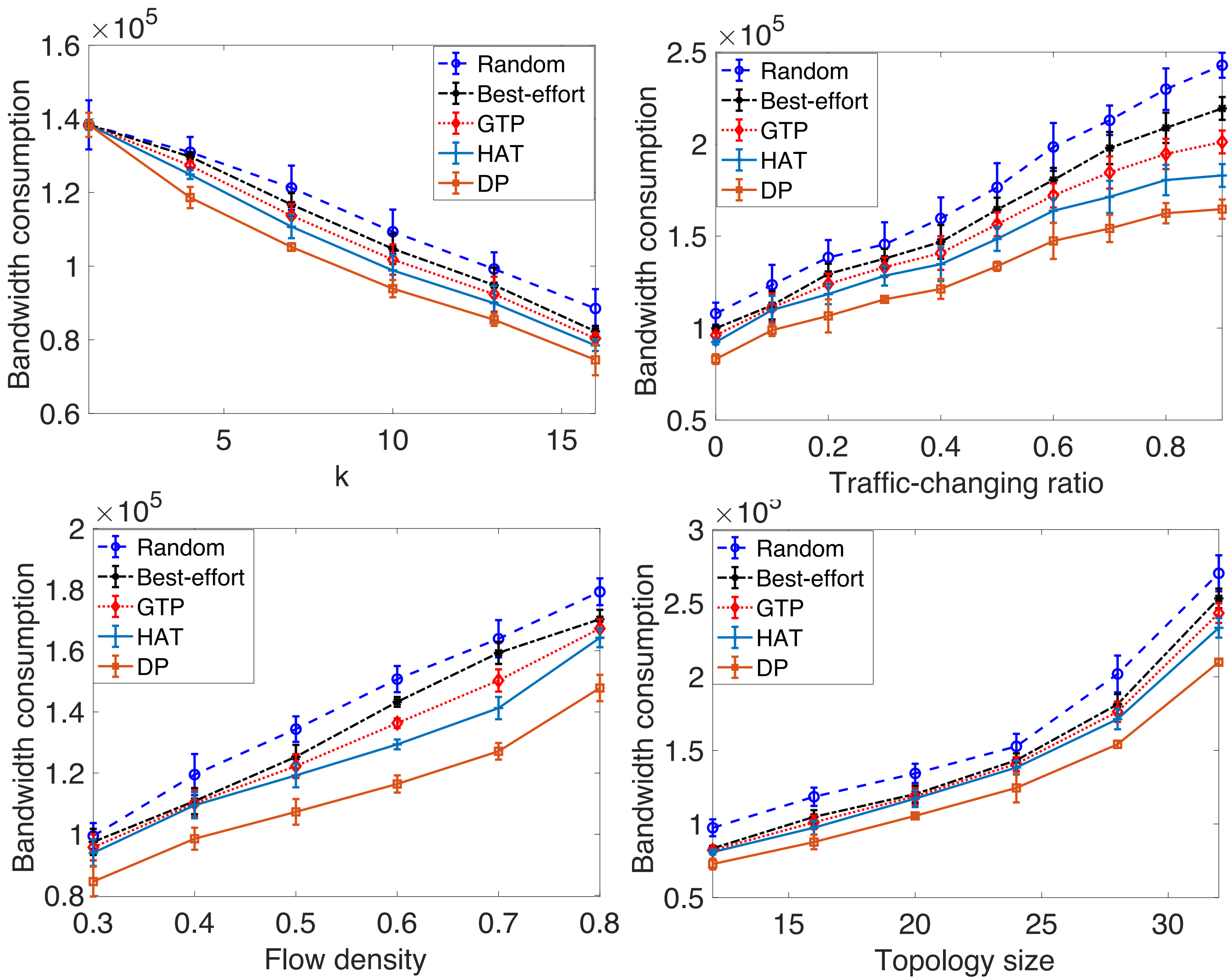
(b) Tree topo (subgraph of (a)).



(c) General topo (subgraph of (a)).

- Middlebox traffic-diminishing ratio
  - From 0 (e.g., spam filters) to 0.9 (e.g., traffic optimizer) with a stride of 0.1
  - Additional simulation on spam filter
- Flow rate distribution
  - CAIDA data center 1-hour packet trace

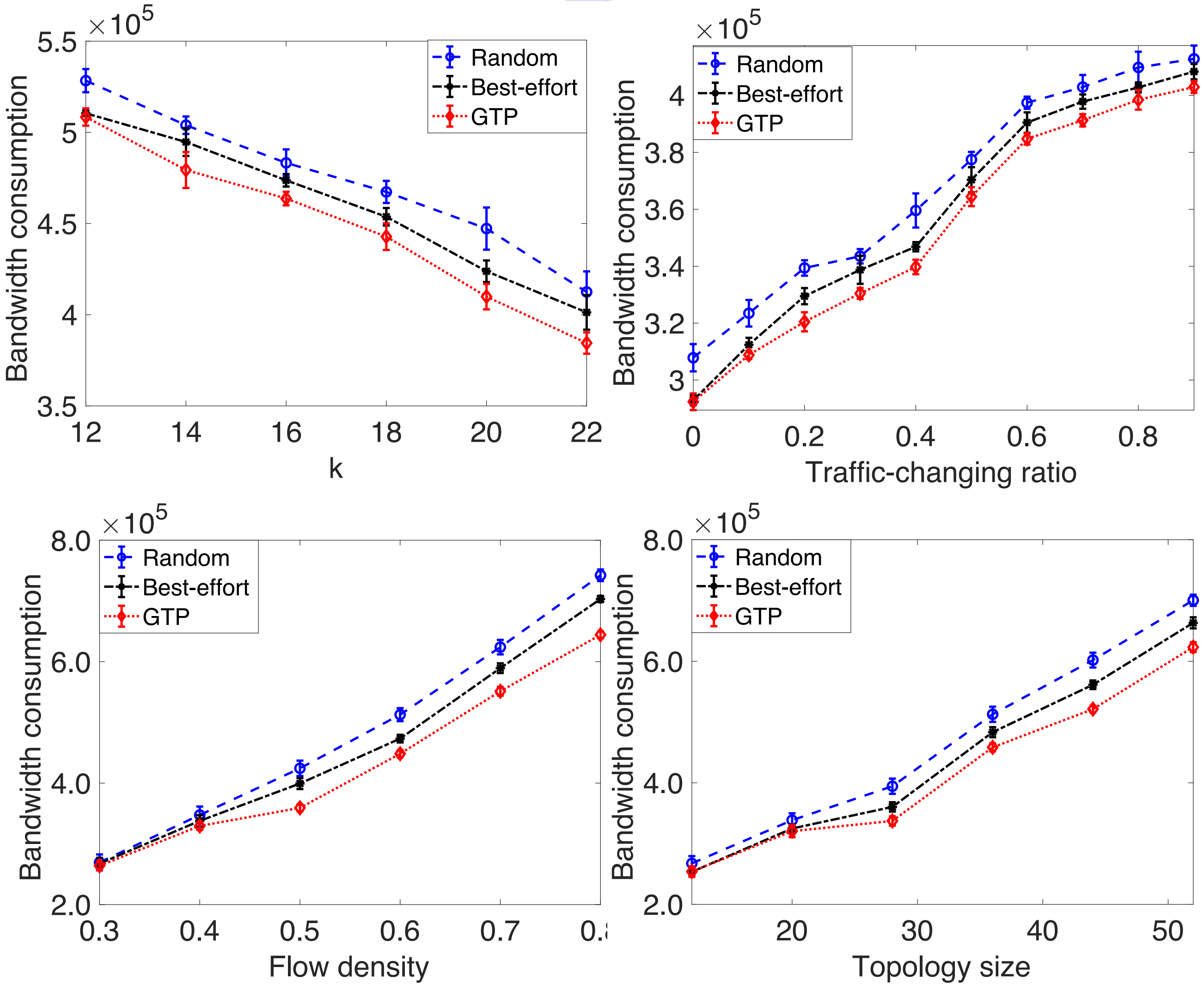
Simulation results of tree



Tree Topology

- Alg. DP performs best for all four variables
- k = 1, only one feasible placement plan for all methods
- Traffic-changing ratio has the largest impact on the bandwidth consumption
- Random has the biggest fluctuation

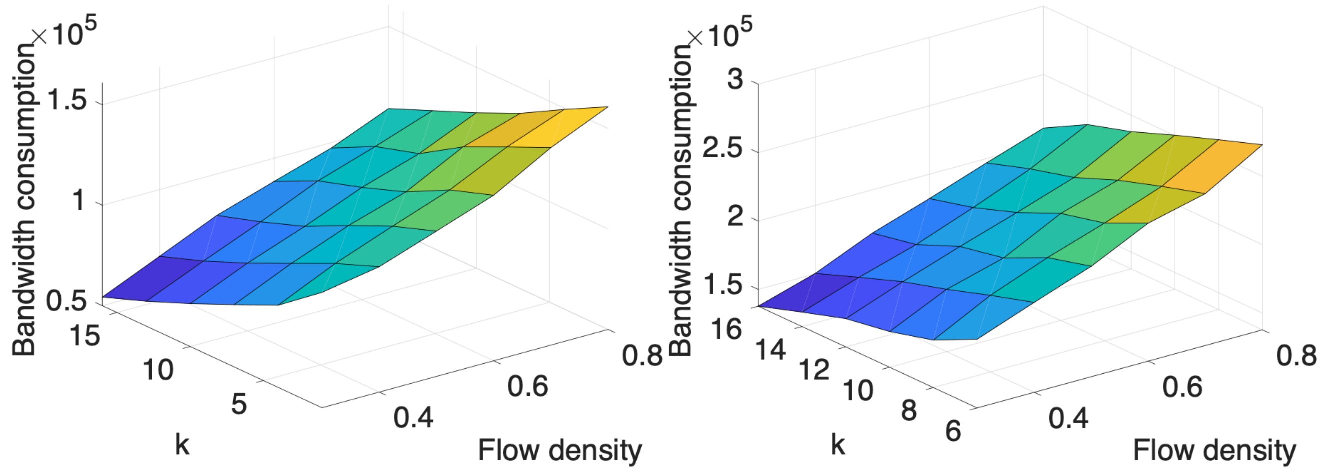
Simulation results of general topology



General Topology

- Alg. GTP always consumes the smallest bandwidth
- Error bars become shorter
- Bandwidth consumption increases faster in fig. b when ratio ranges from 0.4 to 0.6
- When flow density is lower than 0.4 in fig. c, little difference among three algorithms

Simulation results (cont'd)



(a) Tree. **(b) General topology.**

Spam Filter (Traffic diminishing ratio: 0)

- Flow density plays a more important role in affecting the total bandwidth consumption
- When flow density doubles from 0.3 to 0.6, bandwidth consumption in tree increases 30.2%, while increment is only 25.6% in general topo

Conclusion and Future Work

- Problem
  - Deploy a limited number of traffic-diminishing VNFs
  - All flows get processed
- Objective
  - Minimize total bandwidth consumption
- Solutions
  - Tree: optimal and greedy
  - General graph: performance-guaranteed
- Future Work
  - Traffic-expanding VNFs
  - Service chain: an ordered set of multiple VNFs

Optimizing Flow Bandwidth Consumption with Traffic-diminishing Middlebox Placement

Thank you!

Questions?