Network Function Deployment with Balanced Server and Link Resources in Tree Topologies

Yang Chen and Jie Wu

Center for Networked Computing

Temple University, USA

1. Introduction

- Network Function Virtualization (NFV)
  - Virtualizing network functions into software modules

- Virtualized Network Function (VNF)

  software implementation of network services
  - Improve performance:
    - Web proxy, load balancer
  - Enhance security:
    - Firewall, IDS/IPS
  - Examples:

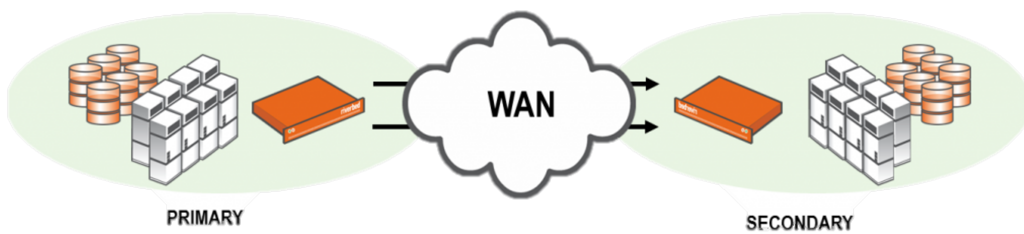
Web Proxy

Firewall

NAT

VNF Traffic Changing Effects [1]

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



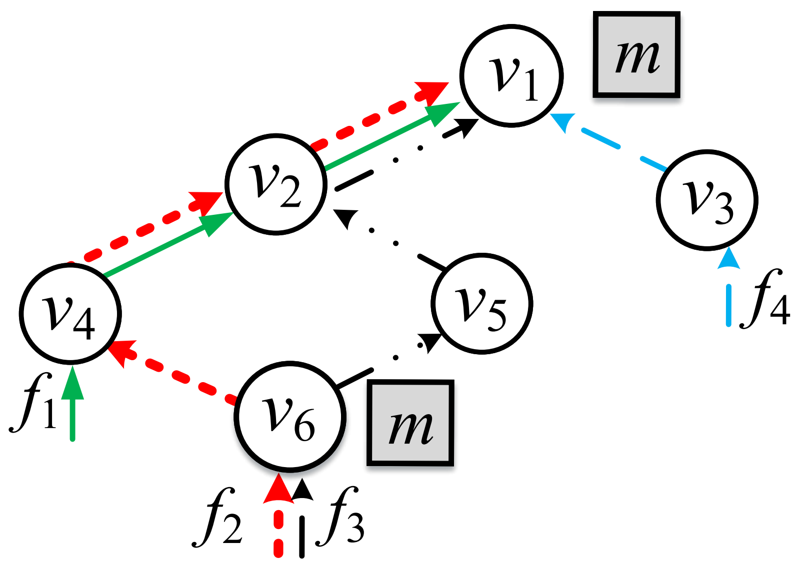
  - BCH(63,48) encoder: 130% (expanding)

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

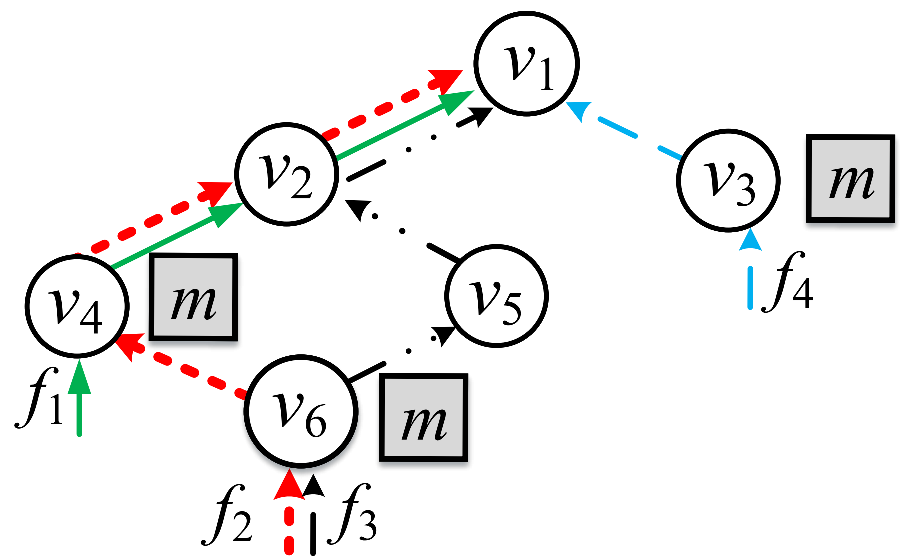
A motivating example

Traffic-diminishing ratio of VNF m: 0.5

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



(a) Two VNFs allowed.



(b) Three VNFs allowed.

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 to be processed
  - Deploy a limited number of a single type of VNFs

3. Problem Formulation

A mathematical optimization problem on minimizing total flow bandwidth consumption

$$\min \quad b(\mathcal{P}) = \sum_{f \in F} b(f) = \sum_{f \in F} \sum_{e \in p_f} b_e(f) \qquad (1)$$

$$\text{s.t.} \quad |\mathcal{P}| = \sum_{v \in V} m_v \leq k \qquad \forall m \in M \quad (2)$$

$$\sum_{v \in p_f} f_v = 1 \qquad \forall f \in F \quad (3)$$

$$f_v \leq m_v \qquad \forall v \in V \quad (4)$$

$$m_v = \{0, 1\}, f_v = \{0, 1\} \qquad \forall v \in V \quad (5)$$

4. Solution for trees

- Lowest Common Ancestor (LCA)
  - LCA of two vertices v and w in an acyclic graph G is the lowest vertex that has both v and w as descendants
- Solution
  - Greedy Solution for Trees (GST)
- Steps
  - Deploy one VNF on each leaf
  - Merge two VNFs with the minimum difference of the total bandwidth value when we delete two VNFs on v and w and place one on LCA(v,w)
  - Until total number of deployed VNFs no more than k

4. Solution for trees (cont'd)

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

5. Solution for DAGs

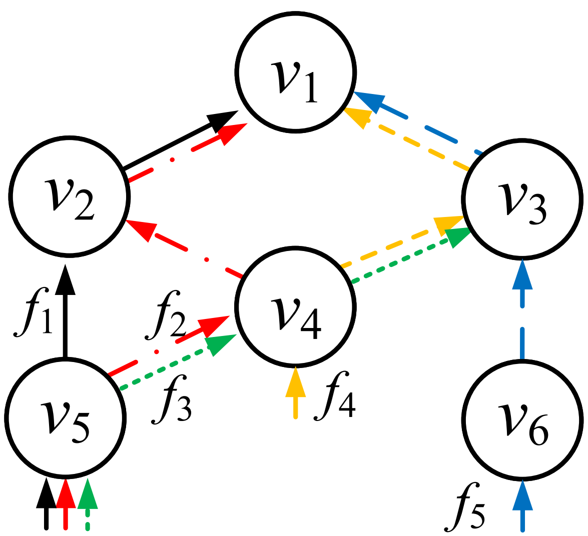
- Directed Acyclic Graph (DAG)
  - A finite directed graph with no directed cycles
  - A tree is a special case of a DAG
- Solution
  - Directed Acyclic Graph Technique (DAGT)
- Steps
  - Deploy one VNF on each all sources of flows
  - Sort all vertices in topological order level by level corresponding to all flows' paths
  - Apply Alg. GST for vertices with the same order

5. Solution for DAGs (cont'd)

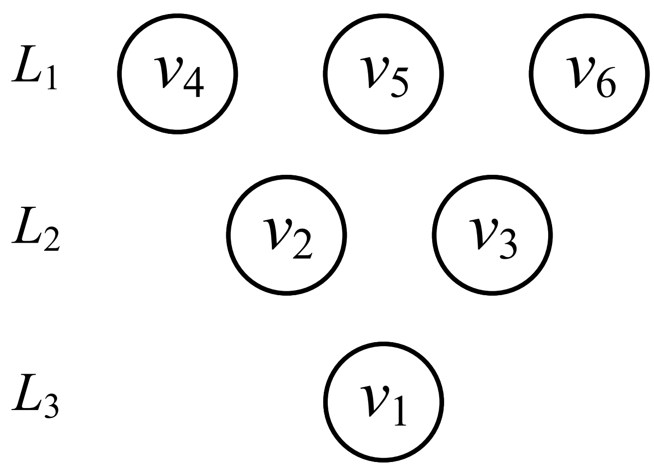
- Time complexity
  - $O(|V|^3 \log |V|)$
  - $|V|$: #vertices

Traffic-diminishing ratio of VNF m: 0.5

Initial flow rate: All are 1



(a) A DAG topology.



(b) Topological order.

4. Solution for general topologies

- Decrement function
  - Decrement of the total bandwidth consumption by a deployment plan
- Marginal decrement
  - Additional bandwidth decrement of processing flows by deploying VNFs on a new subset of vertices beyond current deployment
- Decrement function is submodular
  - Proof insight
    - More VNFs are placed, the less bandwidth consumption is, since each flow can be processed no later than the previous placement

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 $\frac{1}{1-\frac{1}{e}}$
- Time complexity
  - $O(k|V| \log |V|)$
  - $|V|$: #vertices
  - k: limited #VNFs

7. Simulation

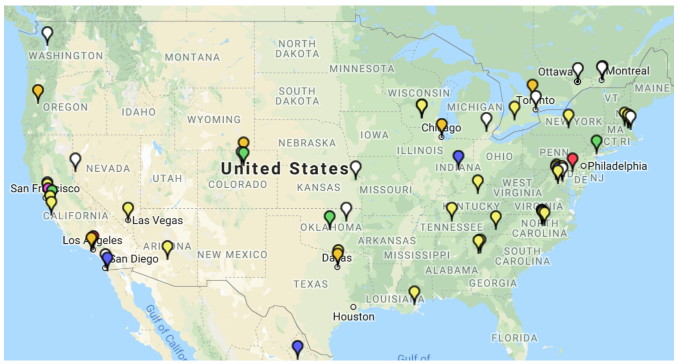
- Comparison algorithms
  - Random
    - randomly deploy VNFs until it places k VNFs
  - Best-effort
    - place one VNF on the vertex, which can reduce the total bandwidth of flows mostly, until it places k VNFs
- Our proposed algorithms
  - Tree: Alg. GST
  - DAG: Alg. DAGT
  - General topo: Alg. GTP

Settings

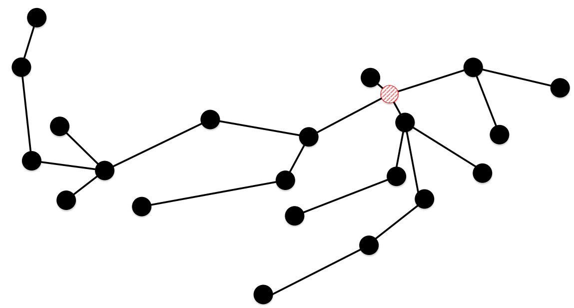
- Variables
  - VNF number constraint k
    - Default value: k = 8 for tree, k = 11 for DAG, k = 17 for the general topo
  - Traffic-changing ratio
    - Default value: 0.5
  - Flow density
    - Default value: 0.5
  - Topology size
    - Default value: 22 for tree, 30 for DAG, 36 for a general topo
  - Topology kind
- Metric
  - Total bandwidth consumption of all flows

Settings

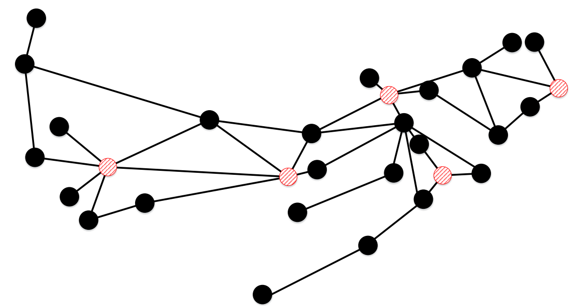
- Topology



(a) The Archipelago (Ark) Infrastructure.



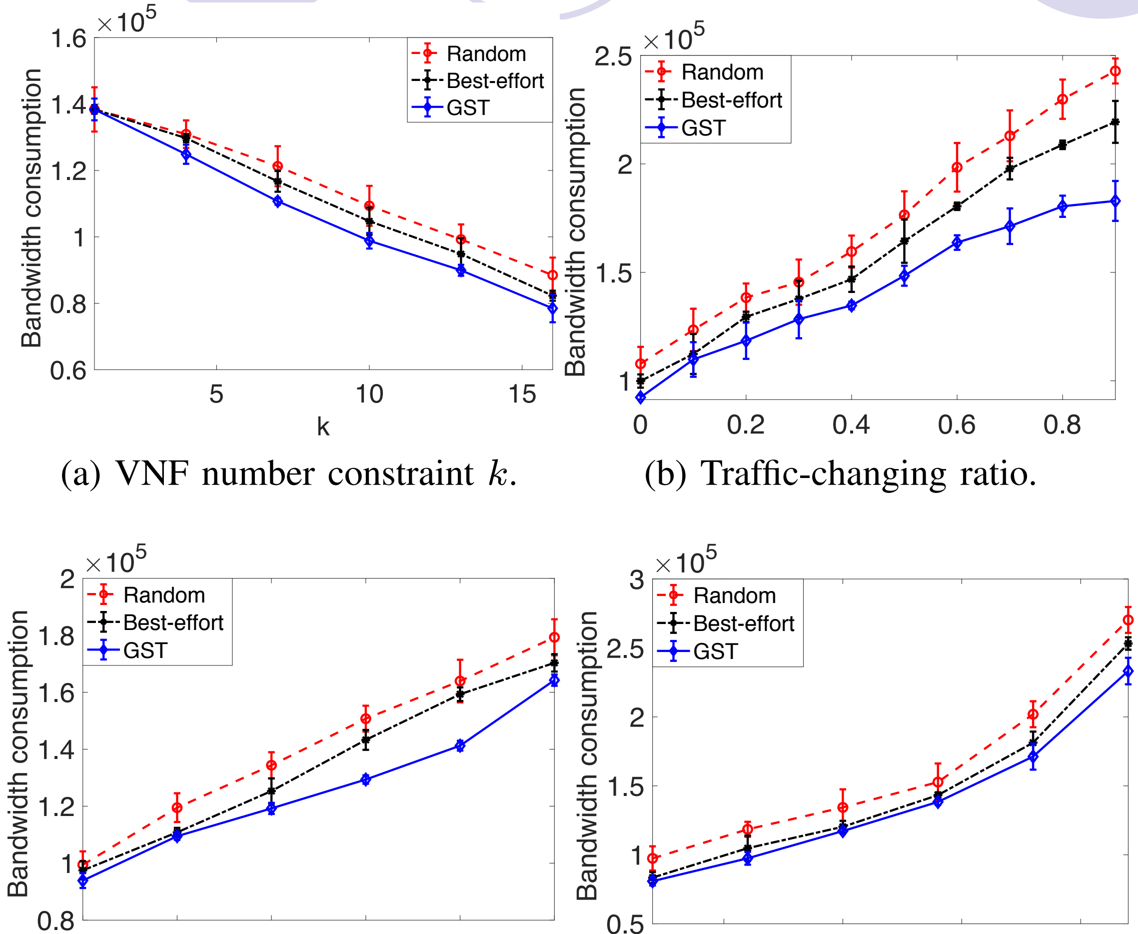
(b) Tree topology (subgraph of (a)).



(c) DAG topology (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



(a) VNF number constraint k.

(b) Traffic-changing ratio.

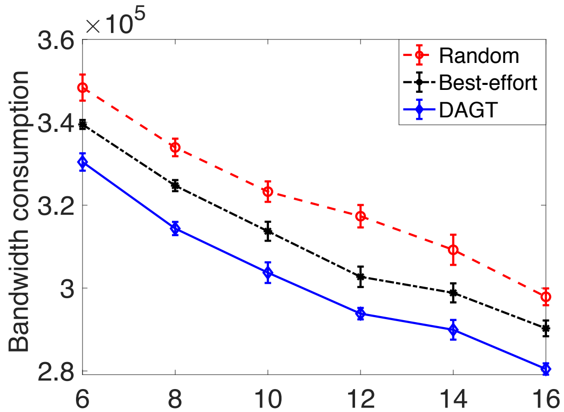
(c) Flow density.

(d) Topology size.

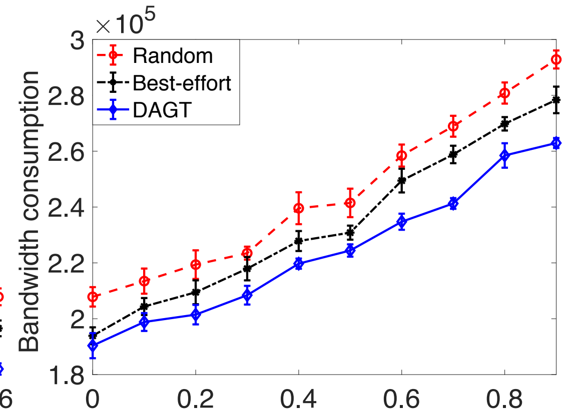
Tree Topology

- Alg. GST 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

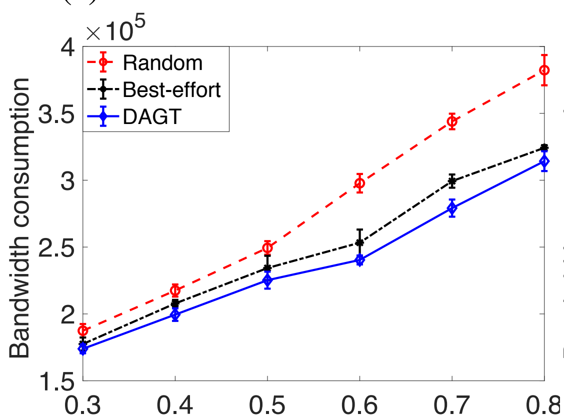
Simulation results of DAG(cont'd)



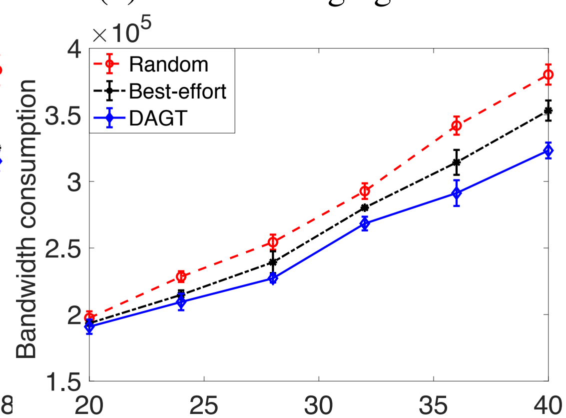
(a) VNF number constraint k.



(b) Traffic-changing ratio.



(c) Flow density.

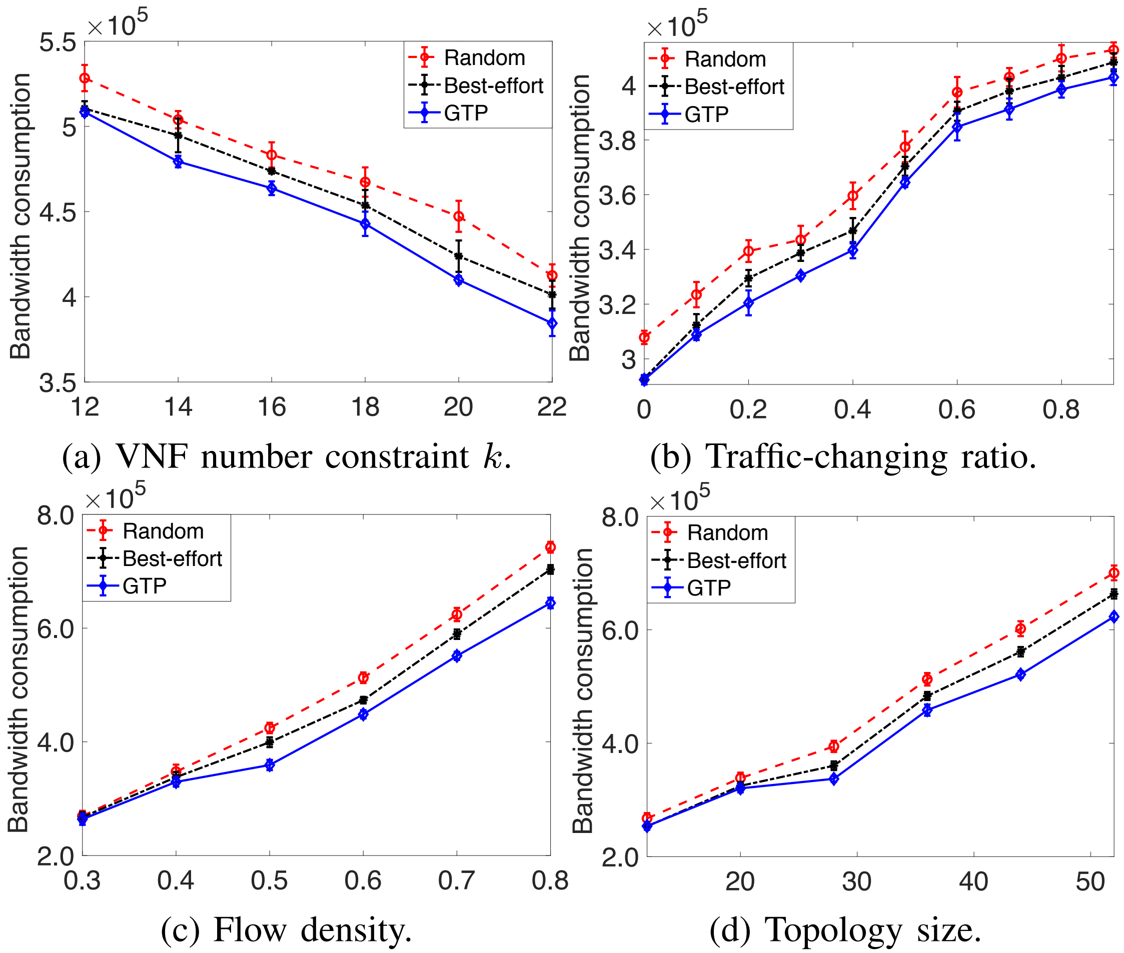


(d) Topology size.

DAG Topology

- Bandwidth consumption is almost three times of tree.
- In Fig. b, bandwidth of DAGT is 85.1% of Random and 92.0% of Best-effort.
- Deviation of DAGT in fig. c is smaller than in last slide
- When the topology size is larger than 35, DAGT performs even better.

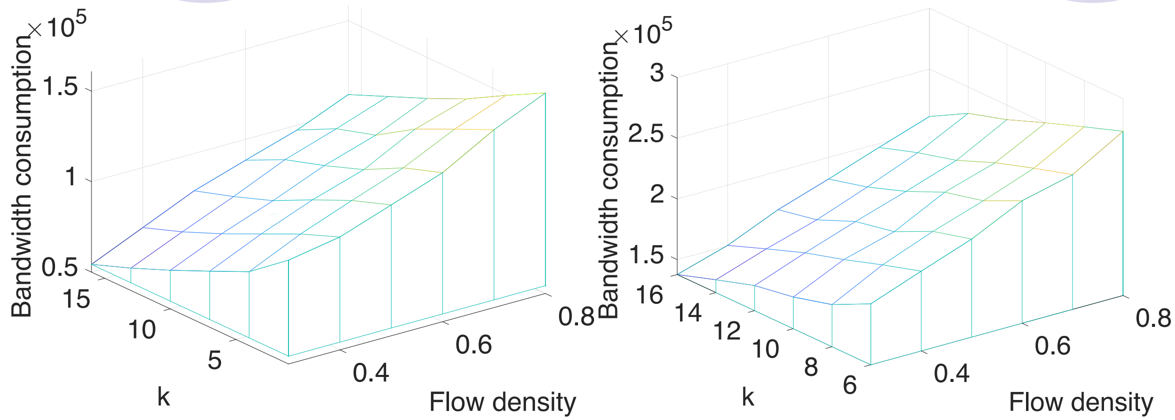
Simulation results of general topo (cont'd)



General Topology

- Error bars become shorter than last two slides
- 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
- Alg. GTP becomes larger when the topology size increases

Simulation results (cont'd)



(a) Tree. (b) DAG.

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 DAG

Conclusion and Future Work

- Problem
  - Using limited number of traffic-diminishing VNFs process all flows
- Objective
  - Minimize total bandwidth consumption
- Solutions
  - Tree: optimal
  - Directed Acyclic Graph: efficient heuristic
  - General graph: performance-guaranteed
- Future Work
  - Traffic-expanding VNFs
  - Service chain: an ordered set of multiple VNFs

Questions contact:

Yang Chen (yang.chen@temple.edu)