On Balancing Middlebox Set-up Cost and Bandwidth Consumption in NFV

Jie Wu

Center for Networked Computing

Temple University, USA

Roadmap

1. Introduction of Middlebox
2. Middlebox Placement Problems
3. Traffic Changing Effects
4. Our Model and Solutions
5. Simulation
6. Conclusion and Future Work

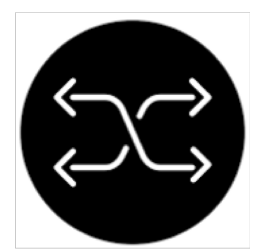
1. Introduction of Middlebox

- Network Function Virtualization (NFV)
  - Technology of virtualizing network functions into software building blocks

- Middlebox: software implementation of network services
  - Improve the network performance:
    - Web proxy and video transcoder, load balancer, …
  - Enhance the security:
    - Firewall, IDS/IPS, passive network monitor, …
- Examples

Web Proxy

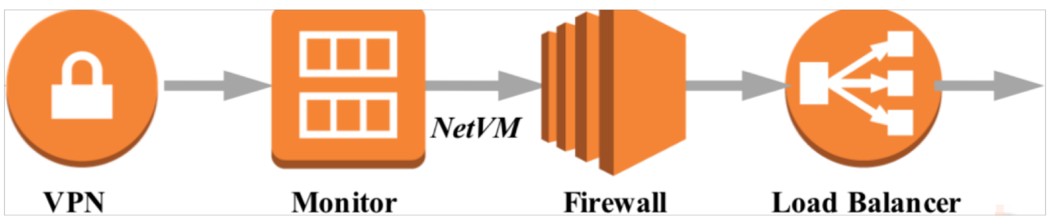
Firewall



NAT

Middlebox Dependency Relations [1]

- Multiple middleboxes may/may not have a serving order
  - Examples
    - Firewall usually before Proxy
    - Virus scanner either before or after NAT gateway
- Categories
  - Non-ordered middlebox set
  - Totally-ordered middlebox set (service chain)



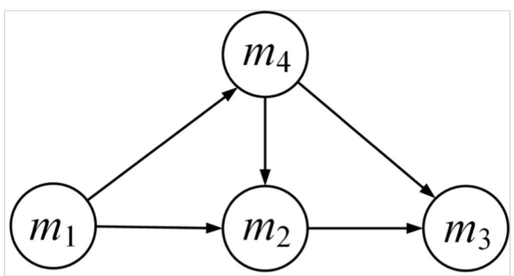
  - Partially-ordered middlebox set

[1] Dynamic Service Function Chaining in SDN-Enabled Networks with Middleboxes (ICNP '16)

2. Middlebox Placement Problems

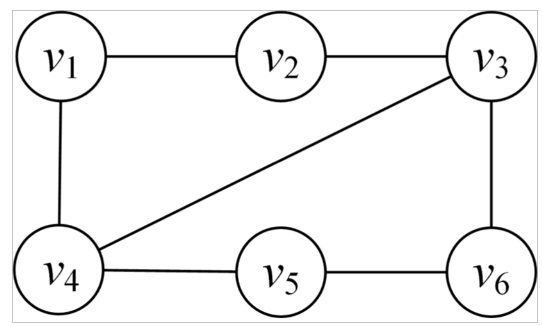
- Graph embedding [2]
  - Middlebox graph, G_m, of multiple service chains that needs to be embedded in a give network graph, G_n.

Virtual network



Embedding

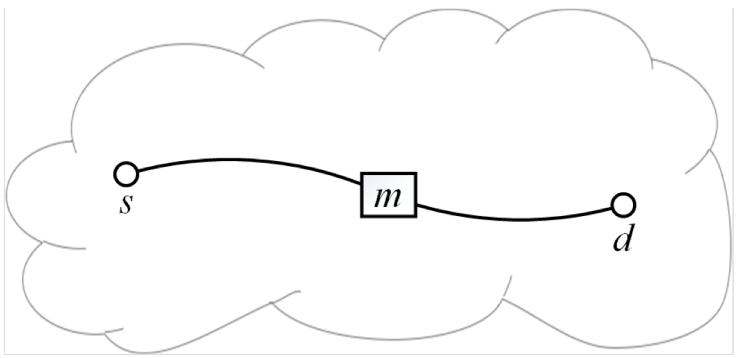
Physical network

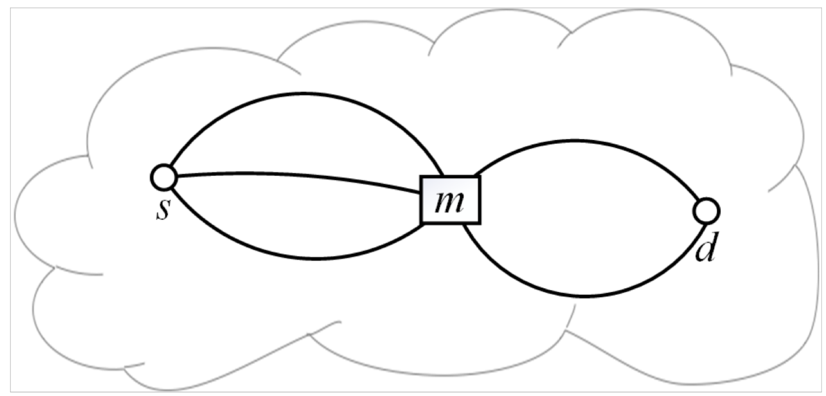


[2] Charting the Complexity Landscape of Virtual Network Embedding (IFIP '18)

Middlebox Placement Problems (cont'd)

- Graph flow routing [3]
  - Shortest path or maximum flow between a given source and destination that have to go through a given middlebox in G_n.

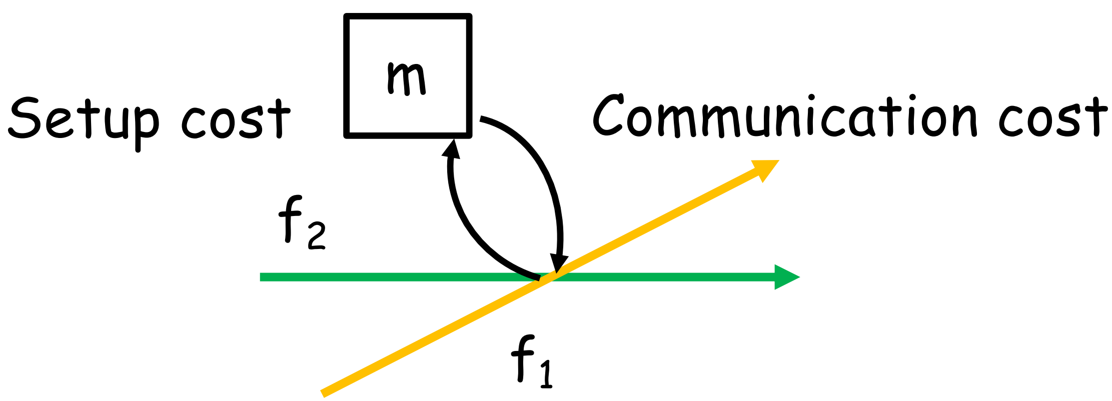




[3] Provably Efficient Algorithms for Joint Placement and Allocation of Virtual Network Functions (INFOCOM '17)

Middlebox Placement Problem (cont'd)

- Facility allocation [4]
  - Optimal placement of facilities (i.e., middlebox) to minimize transportation costs (i.e., traffic, including detour traffic from flows to middleboxes).
- Cost

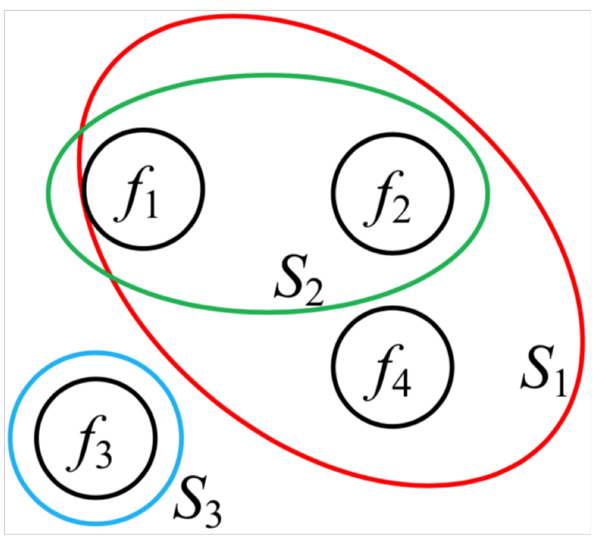


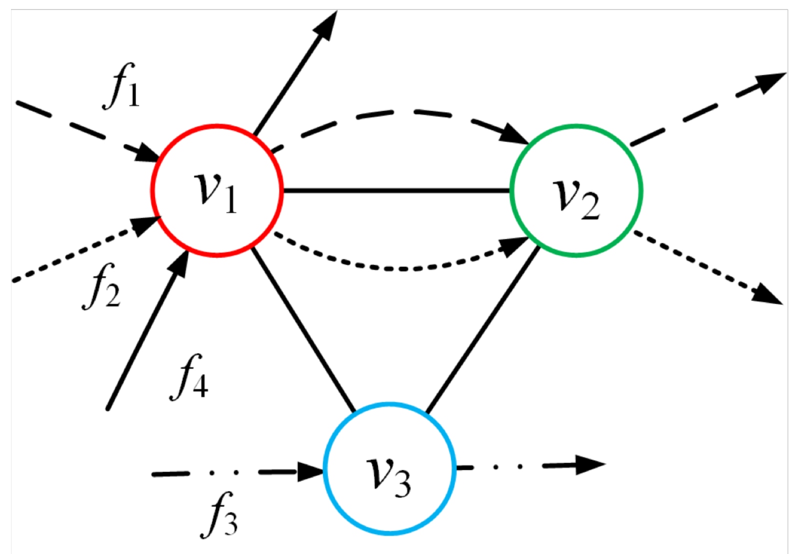
- Objective
  - Minimizing sum of middlebox setup cost and communication cost

[4] Near Optimal Placement of Virtual Network Functions (INFOCOM '15)

Middlebox Placement Problems (cont'd)

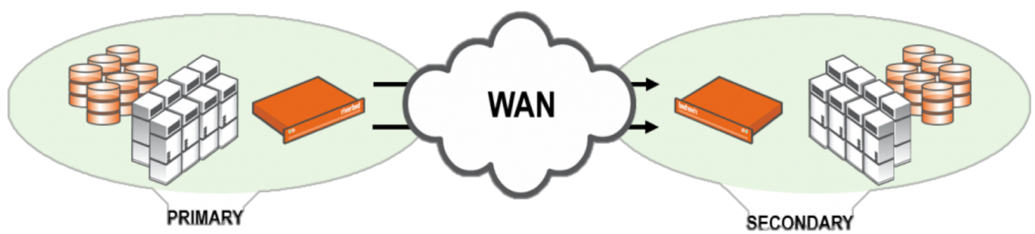
- Set covering
  - Minimize the number of middleboxes used to cover all flows.
  - NP-hard





3. Traffic Changing Effects [5]

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



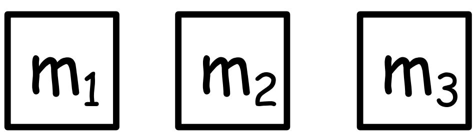
  - BCH(63,48) encoder: 130% (expanding)

- Objetive: minimizing total traffic

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

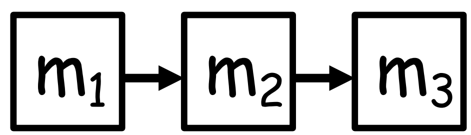
Service Chain Models

- Objective
  - Minimizing the total bandwidth consumption
- Solutions
  - Consider traffic-changing effects
  - Place middleboxes for a single flow



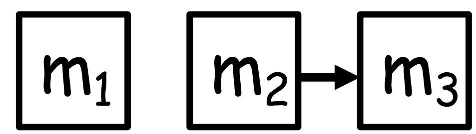
Non-ordered

(Optimal greedy: sort traffic-changing ratios in increasing order)



Totally-ordered

(Optimal DP: latter middleboxes must be after front ones)



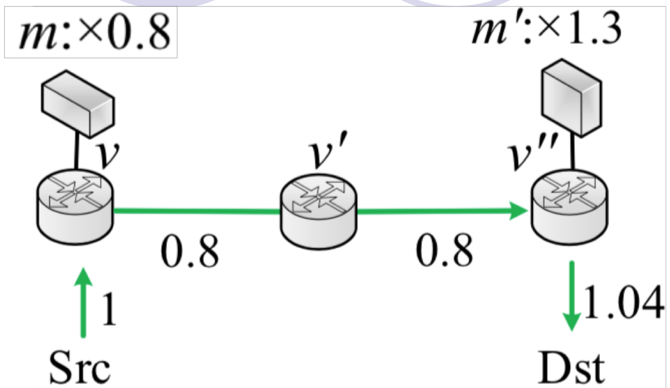
Partially-ordered

(NP-hard: reduced from the Clique Problem)

4. Our Model and Solutions

- Problem
  - Placing middleboxes to satisfy all flows' network service requests
- Network service requests
  - Multiple middleboxes
    - Middlebox set with or without dependency relations
- Cost
  - Middlebox setup
    - Sum of middlebox setup cost (amortized over a period of time)
  - Bandwidth consumption
    - Sum of each flow's bandwidth consumption cost on each link
- Objective
  - Minimizing total cost of middlebox setup and bandwidth consumption

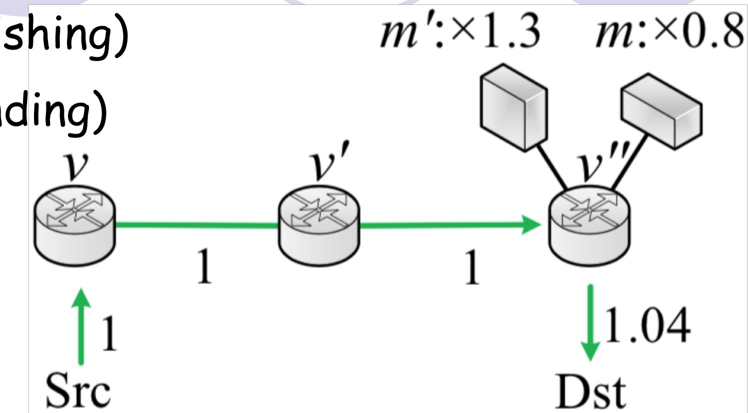
A Motivating Example



Independent middleboxes

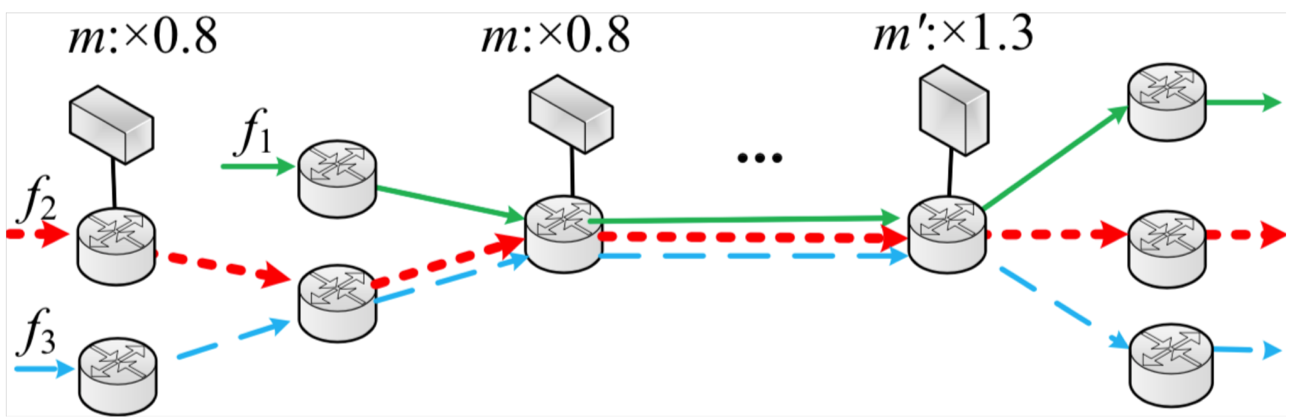
m: 0.8 (diminishing)

m′: 1.3 (expanding)



Dependent middleboxes: m′ before m

Red flow with high rate



A flow covered by multiple middleboxes

(When additional setup cost is less than the reduced bandwidth consumption)

Problem Formulation

- Middlebox setup cost
  - $$c_1 = \sum_{m \in M} \sum_{v \in V} c_m$$
    - c_m: unit setup cost of middlebox m
- Bandwidth consumption cost
  - $$c_2 = \sum_{f \in F} \sum_{e \in p_f} w(b_f^e)$$
    - $w(b_f^e)$: bandwidth cost function of flow f on link e
  - $$b_f^e = r_f \prod_m \lambda_m$$
    - r_f: initial traffic rate of flow f
    - λ_m: traffic-changing ratio of middlebox m
- Objective
  - Minimizing c_1+c_2

Problem Formulation (cont'd)

- Translog bandwidth cost function on each link[6]

$$w(b_f^e) = \log(b_f^e) = \log(r_f \prod \lambda_m) = \log(r_f) + \sum \log(\lambda_m)$$

- Reasons
  - Widely used in Cisco EIGRP and OSPF protocols
  - Log-linear for easy calculation
- The weight of setup cost and bandwidth consumption
  - Adjusting the traffic-changing ratios and unit setup costs of middleboxes

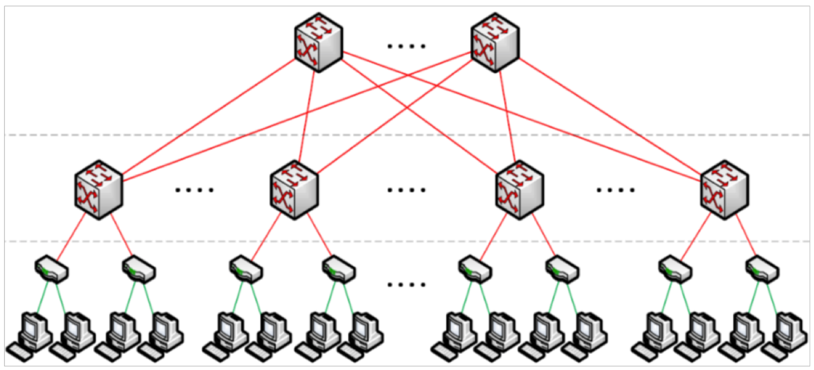
[6] Computer Networking: A Top-Down Approach (Book)

Overview

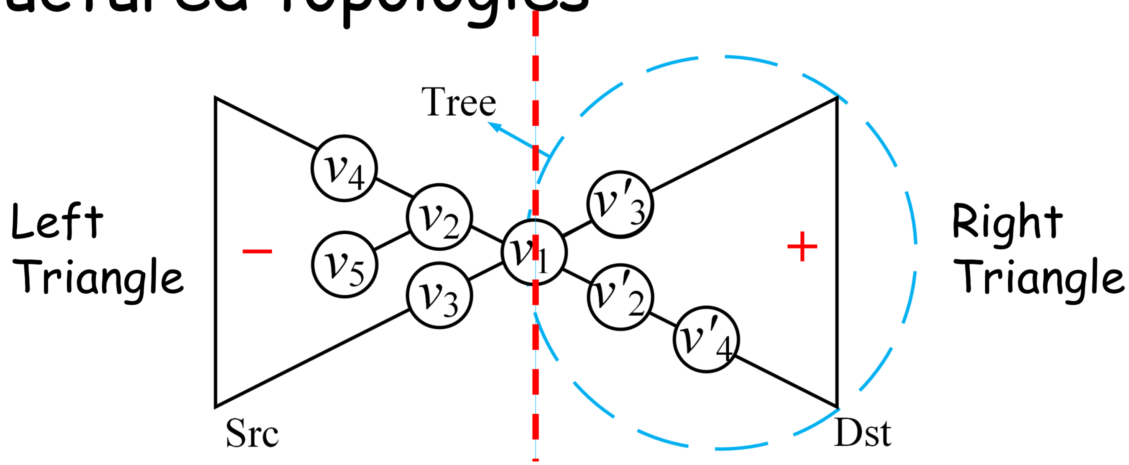
- Optimal solutions for homogeneous flows
  - Single middlebox
    - Greedy
  - Non-ordered middlebox set
    - Greedy
  - Totally-ordered middlebox set
    - Dynamic Programming
- Performance-guaranteed solution for heterogeneous flows

Topology Structure

- We focus on tree-structured topologies

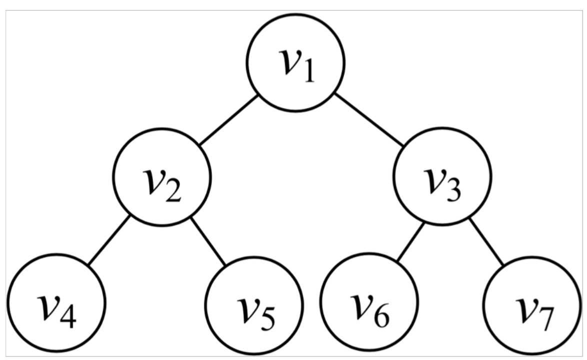


Tree-based data centers

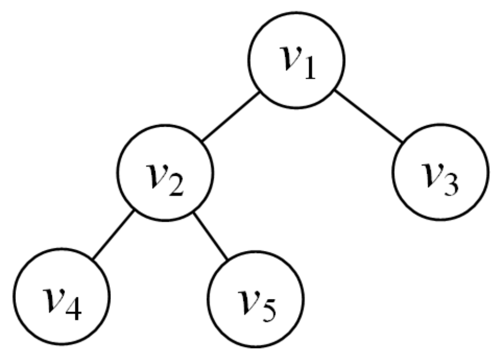


Double-tree structure

Each triangle is mostly a perfect or complete tree



Perfect tree

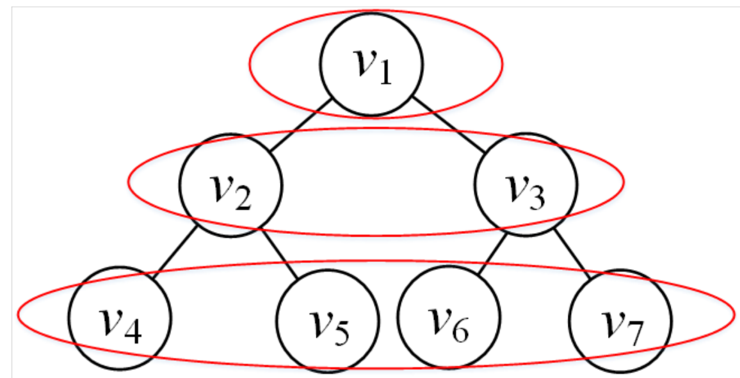


Complete tree

Placing a Single Middlebox

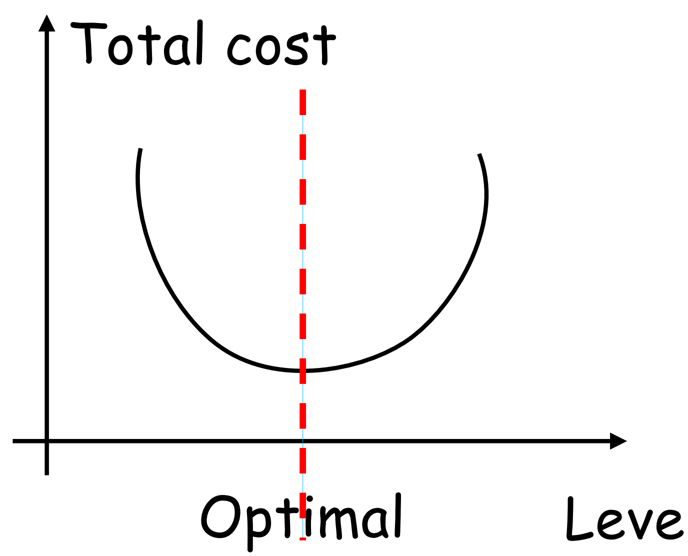
Solution

- Local Greedy Algorithm (LGA)



Steps

- Calculate the total cost of placing middleboxes in a level
- Select the level with the minimum total cost
- Convex function: sufficient to select the local minimum



Placing a Single Middlebox (cont'd)

- Time complexity (|V|: #node)
  - O(|V|) (O(log|V| for perfect tree)
- Optimal for perfect tree topologies
  - Symmetry of placement
  - No multiple "coverage" situation
- Also optimal for complete tree topologies
  - Also no multiple "coverage" situation

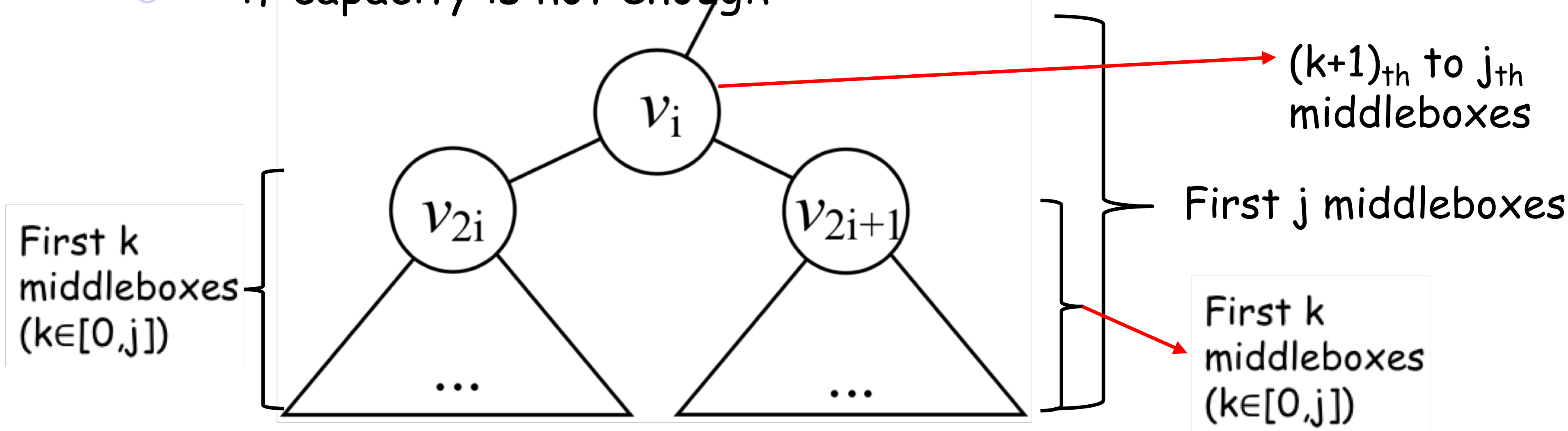
Placing Multiple Middleboxes

Non-ordered middlebox set placement

- Solution
  - Combined Local Greedy Algorithm (CLGA)
- Insight
  - Place each middlebox independently by applying LGA
- Time complexity ($|V|$: #node, $|M|$: #middlebox)
  - $O(|M|\ |V|)$ or $O(|M| \log |V|)$
- Optimal for perfect and complete trees

Totally-ordered Middlebox Set Placement

- Solution: Dynamic Programming (DP)
- Works for infinite and finite vertex capacity
- OPT(i, j)
  - Minimum cost of subtree with root v_i when placing first j middleboxes in the set
  - ∞ if capacity is not enough



Dynamic Programming Formulation

- Left triangle

$$\mathrm{OPT}(i,j) = \begin{cases} \min\limits_{0 \le k \le j} \{\mathrm{OPT}(2i,k) + \mathrm{OPT}(2i+1,k) \\ \quad + \sum\limits_{k<l\le j} c_l + \lfloor \log i \rfloor \sum\limits_{k<l\le j} \log \lambda_l\}, & 1 \le i \le \lfloor \frac{n}{2} \rfloor. \\ \sum\limits_{1\le l\le j} c_l + \lfloor \log i \rfloor \log r_f + \lfloor \log i \rfloor \sum\limits_{1\le l \le j} \log \lambda_l, & \lfloor \frac{n}{2} \rfloor < i \le n. \\ \infty & \text{if not enough node capacity.} \end{cases}$$

Left subtree

Right subtree

Changing rate consumption

Base bandwidth consumption

Newly placed middleboxes

- Right triangle
  - Similar to the left triangle's formulation

Totally-ordered Middlebox Set Placement (cont'd)

Insights

- The optimal placement with root v_i by placing first j and its two subtrees by placing no more than j middleboxes

Perfect tree

- Transformed to a line
- Similar to a single flow placement

Complete tree

- No multiple "coverage" situation

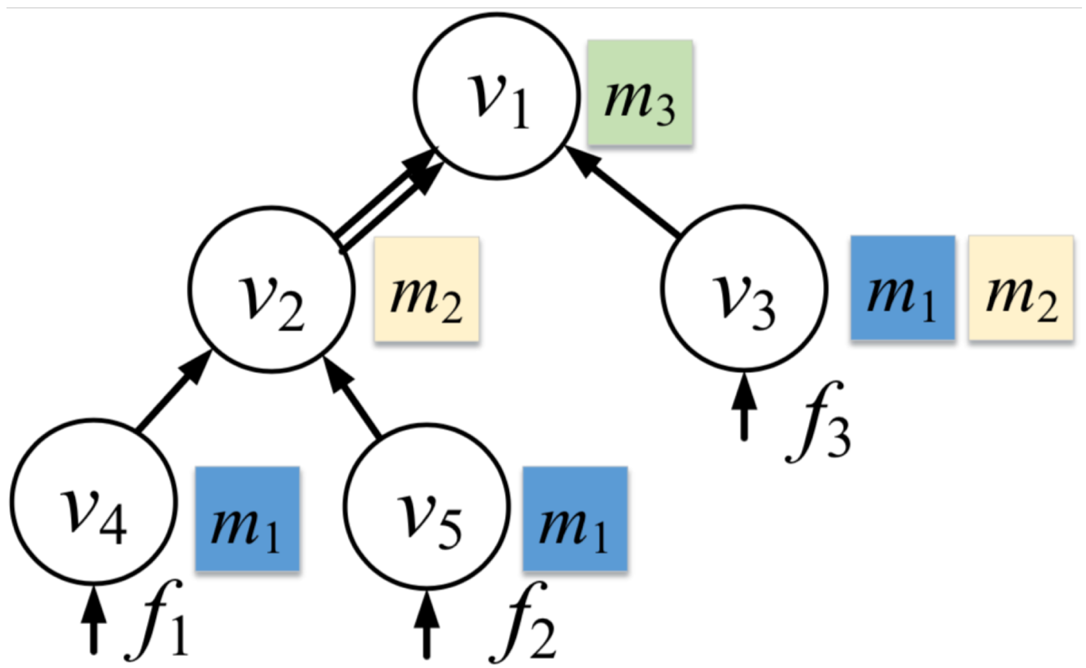
Time complexity (|V|: #node, |M|:#middlebox)

- $O(|M|^3 |V|)$ or $O(|M|^3 \log|V|)$

An Example

	m_1	m_2	m_3
Traffic-changing ratio	0.5	0.8	1.1
Setup cost	0.2	0.4	0.3

- Dependency relations
  - m_1->m_2->m_3
- Initial traffic rate
  - r_1=r_2=r_3=1

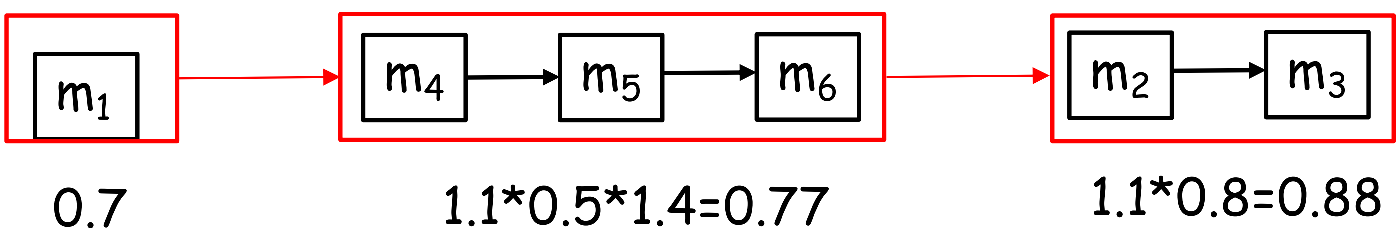


Partially-ordered Middlebox Set Placement

- NP-hard even for a single flow [2]
- One heuristic solution
  - Insight
    - Transform into a totally-ordered middlebox set
  - Steps (λ: traffic-changing ratio)
    - Treat middleboxes with dependencies as a single middlebox
    - Sort middleboxes in increasing order of λ
  - Example
    - Middlebox set

	m_1	m_2	m_3	m_4	m_5	m_6
λ	0.7	1.1	0.8	1.1	0.5	1.4

    - Dependency relationship: $m_2 \rightarrow m_3$, $m_4 \rightarrow m_5 \rightarrow m_6$



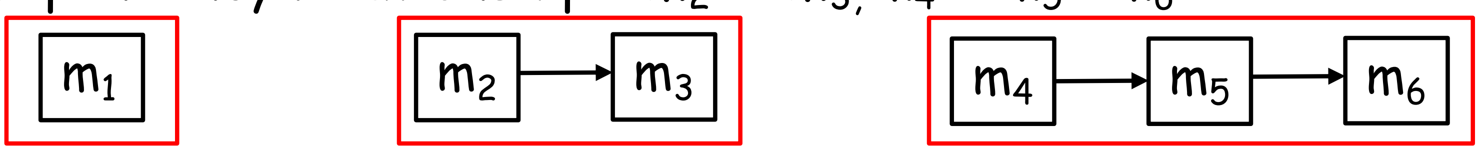
[2] Traffic aware placement of interdependent NFV middleboxes (INFOCOM '17)

Partially-ordered Middlebox Set Placement (cont'd)

- Another heuristic solution
  - Insight
    - Transform into a non-ordered middlebox set
  - Steps
    - Treat middleboxes with dependencies as a single middlebox by a topological order
    - No dependency relations among new middleboxes
  - Example
    - Middlebox set

	m_1	m_2	m_3	m_4	m_5	m_6
λ	0.7	1.1	0.8	1.1	0.5	1.4

    - Dependency relationship: $m_2 \rightarrow m_3$, $m_4 \rightarrow m_5 \rightarrow m_6$



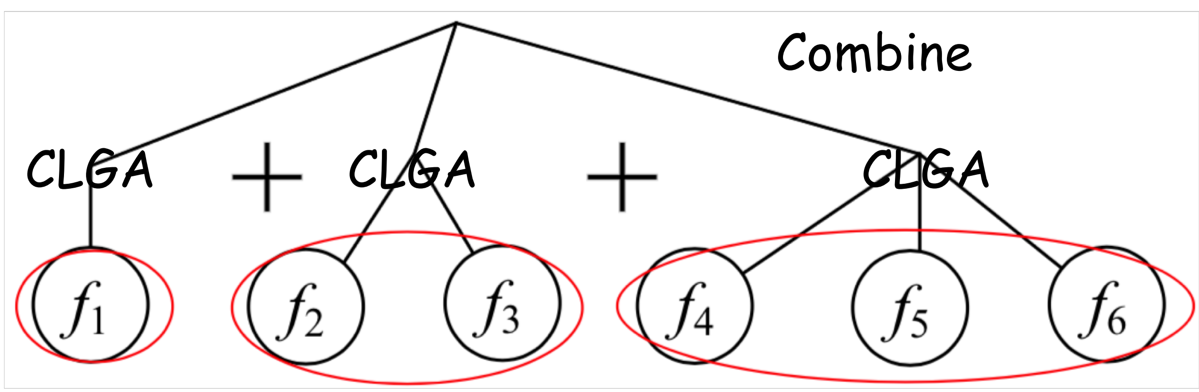
0.7

1.1*0.8=0.88

1.1*0.5*1.4=0.77

Handling Heterogeneous flows for Non-ordered Middlebox Set

- Group Flows by Initial Bandwidths (GFIB)
  - Group flows by initial traffic rates (r_f: f's traffic rate)
    - #group: $\lfloor \log_2 \frac{\max r_f}{\min r_f} \rfloor + 1$
    - The traffic rate range of the i^{th} group: $2^{i-1} \times \min r_f \leq r_f < 2^i \times \min r_f$
  - Treat flows in each group as homogeneous
  - Apply CLGA for each group
- An example



r_1=1 r_2=2 r_3=3 r_4=4 r_5=6 r_6=7

max r_f= 7
min r_f= 1
Group 1: [1,2)
Group 2: [2,4)
Group 3: [4,8)

Handling Heterogeneous Flows for Non-ordered Middlebox Set (cont'd)

- Time complexity

$$\max\{O(|V|\log|V|), O(|V|(\left\lfloor \log_2 \frac{\max r_f}{\min r_f} \right\rfloor + 1))\}$$

- Performance-guaranteed algorithm
  - Approximation ratio [6]: $\left\lfloor \log_2 \frac{\max r_f}{\min r_f} \right\rfloor + 1$

[6] On Optimal Scheduling of Multiple Mobile Chargers in Wireless Sensor Networks (MSCC '14)

4. Simulation

- Our algorithms
  - LGA
    - Single middlebox
    - Select the level with the minimum cost
  - CLGA
    - Non-ordered middlebox set
    - Apply LGA independently
  - DP
    - Totally-ordered middlebox set
    - Dynamic programming
  - GFIB
    - Heterogeneous flows
    - Group flows by initial traffic rates
    - Combine placement by applying CLGA for each group

5. Simulation

- Comparison algorithms
  - Random-fit
    - Randomly place middleboxes until all flows are satisfied
  - NOSP [2]
    - For single middlebox or non-ordered middlebox set
    - Place middleboxes for each flow independently
  - TOSP [2]
    - For totally-ordered middlebox set with or without vertex capacity
    - Dynamic programming based algorithm for each flow independently

[2] Traffic aware placement of interdependent NFV middleboxes (INFOCOM '17)

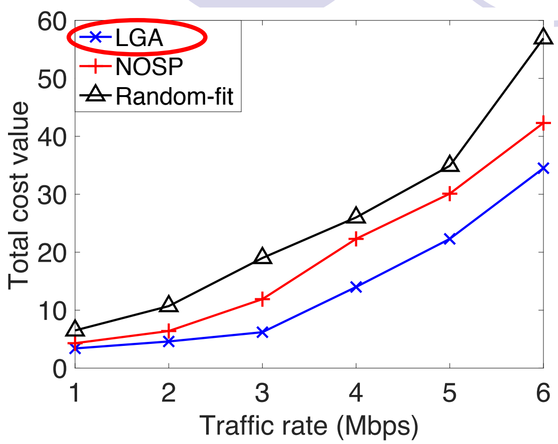
Settings

- Topology
  - Perfect 5-layer binary tree for each triangle
- Facebook data center traffic trace
  - Single-flow initial traffic rate: 1~6 Mb
- Middlebox set

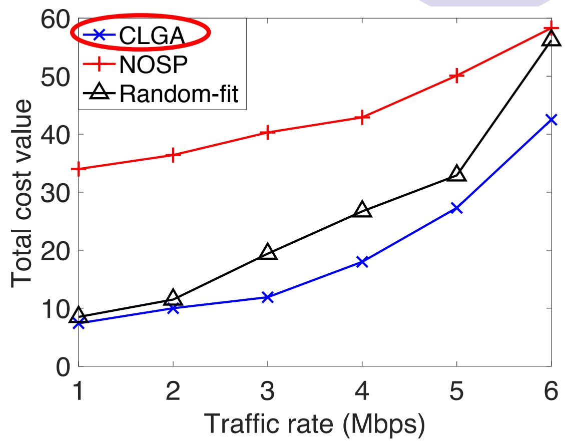
	m_1	m_2	m_3	m_4
Traffic-changing ratio	0.7	0.8	1.1	1.2
Set-up cost	0.4	0.6	0.2	0.8

- Dependency relationship
  - m_2 -> m_3 -> m_1 -> m_4

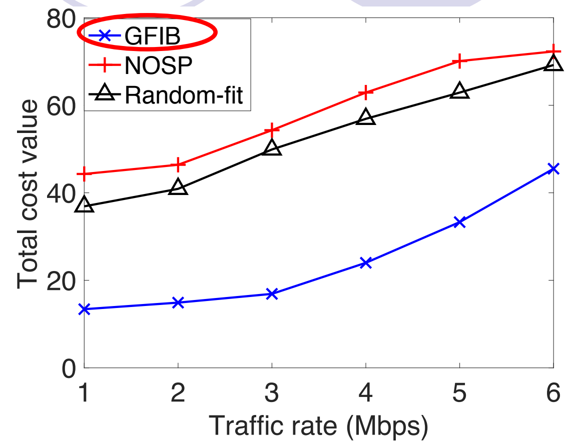
Simulation Results



Single middlebox (LGA)



Non-ordered middlebox set (CLGA)

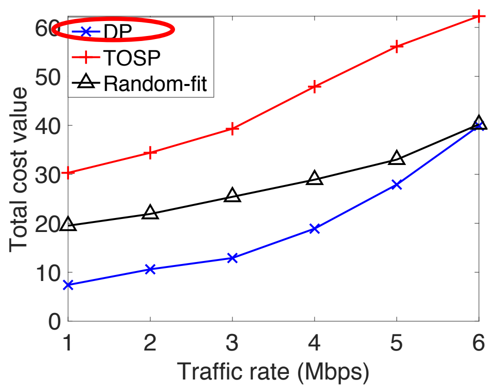


Bandwidth heterogeneity (GFIB)

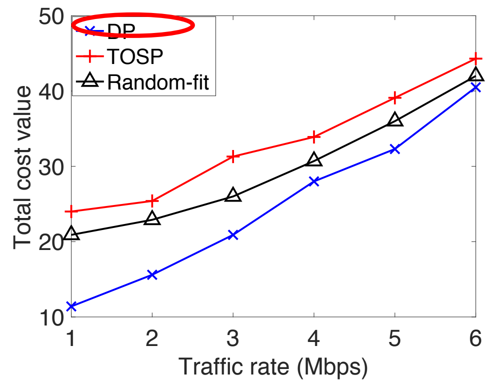
- LGA costs 20.3% less than NOSP and 35.1% less than Random-fit.
- CLGA performs the best even with heavy traffic.
- For heterogeneous flows, GFIB saves about 36.9% and 34.0% compared to NOSP and Random-fit.

Simulation Results (cont'd)

Totally-ordered middlebox set with (2) and without vertex capacity



Without vertex capacity



With vertex capacity

Totally-ordered middleboxes	Total cost	Set-up cost
$m_2 \to m_3 \to m_1 \to m_4$	20.9	10.4
$m_3 \to m_1 \to m_2 \to m_4$	23.7	12.0
$m_1 \to m_4 \to m_3 \to m_2$	22.8	9.6
$m_1 \to m_2 \to m_3 \to m_4$	11.9	4.4
$m_4 \to m_3 \to m_2 \to m_1$	24.7	10.2

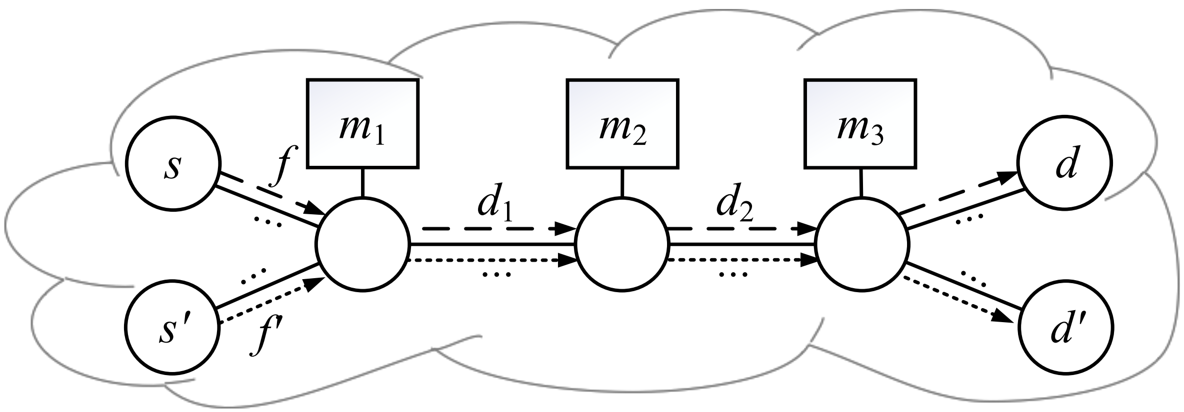
Middlebox order effect at 3 Mbps (DP)

- The total cost is larger than the non-ordered middlebox set.
- Limited vertex capacity increases the minimum cost.

5. Conclusion and Future Work

- Middlebox constraints
  - Traffic-changing effects, dependency relations, and flow sharing
- Middlebox placement
  - Balancing middlebox set-up cost and bandwidth consumption
- Tree-structured topologies
  - Optimal algorithms for homogeneous flows
  - Performance-guaranteed algorithm for heterogeneous flows
- Future work
  - General tree-structure and other topologies

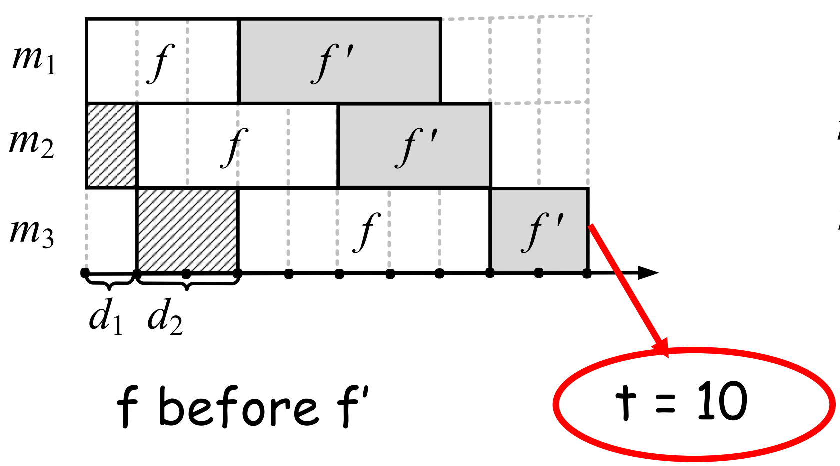
Future Work: Other Chain Models

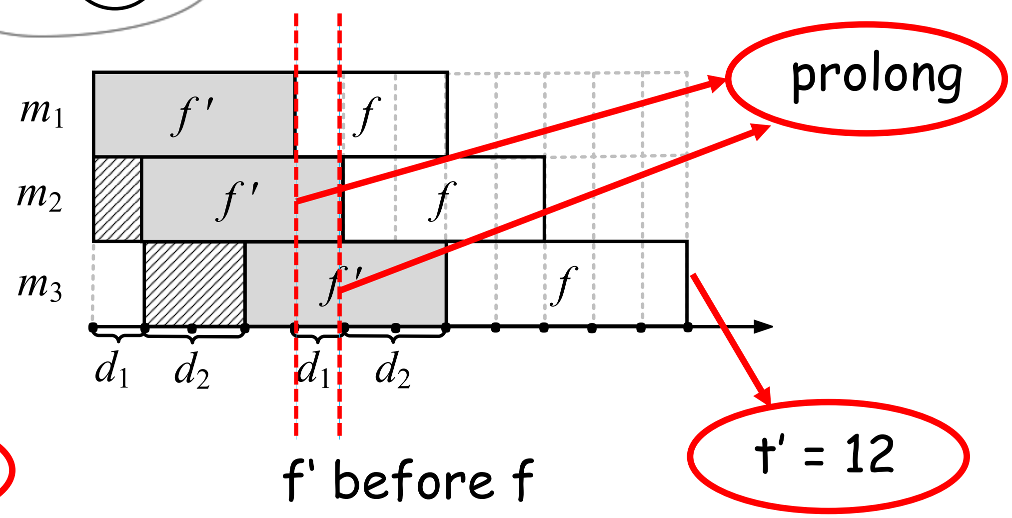


$d_1 = 1$ $d_2 = 2$

Processing time

Flows \ Middleboxes	m_1	m_2	m_3
f	3	4	5
f'	4	3	2





- Minimizing the makespan (similar to flow shop)
- Minimizing the average completion time

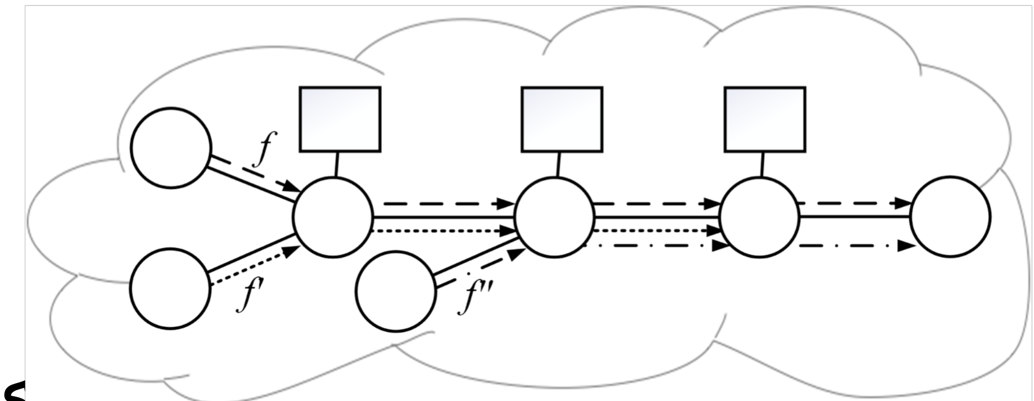
Q & A

Y. Chen and J. Wu, " NFV Middlebox Placement with Balanced Set-up Cost and Bandwidth Consumption," *Proc of ICPP*, August 13-16, 2018.

Y. Chen, J. Wu, and B. Ji, " Virtual Network Function Deployment in Tree-structured Networks," *Proc. of ICNP*, September 24-27, 2018.

Other Service Chain Models

- One box with different volumes/costs



- Solutions

Topo \ Type	Heterogeneous (NP-hard)		Homogeneous	
Tree	DP	Optimal	DP	Optimal
		$O(\|V\|^4 \times (c_{max} \times w_{max})^3)$		$O(\|V\|^4 \times (c_{max})^3)$
Line	Greedy	Approximate	Greedy	Optimal
		$O(\|V\|^2 \times \|M\| \times c_{max})$		$O(\|V\| \times c_{max})$

V: set of vertices

M: set of box types (<10)

c_{max}: Max box number per vertex (<30)

w_{max}: Max box cost scale (tunable)