Minimizing Transmission and Processing Delay in a NFV-based Network

Yang Chen and Jie Wu
Center for Networked Computing
Temple University, USA
1. Introduction

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

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

- **Service chain**
  - Multiple middleboxes in a specific processing order
  - Example

![Diagram showing network functions: VPN, Monitor, Firewall, Load Balancer]
2. Our Model

- **Problem**
  - Flow contention on a service chain

- **Flow communication latency behaviors**
  - Middlebox processing time
    - Distinct value for different flows on different middleboxes
  - Link transmission delay
    - Constant value for all flows on a single link

- **Objective**: minimizing flow completion time in two aspects
  - Minimize the makespan (longest flow completion time)
  - Minimize the average flow completion time
launch the middlebox software within about 30ms, which improves the running efficiency of virtual middleboxes by making it feasible to dynamically schedule the middleboxes. Middlebox placement problem of choosing service locations manoeuvres through NFV traffic and allows the flow order to incorporate SDN and NFV in concerted ecosystems [3]. With Defined Networking (SDN) emerges, so does a tendency to chain.

Two aspects respectively: the makespan (the longest completion transmission and processing delay model to formulate latency behaviors and control the distance cost between flows' paths and middleboxes and processed in a fixed order. We are given a set of unsplittable jobs that are sensitive to TCP packet ordering (e.g., e.g., etc.) of middleboxes according to online changing traffic. However, they pay little attention to the flow scheduling in middlebox middleboxes placement and path selection. Li et al. [20] provide an optimal solution when all jobs can be scheduled perfectly. Unfortunately, this is an NP-hard problem, harder to obtain. Minimizing the average completion time with regard to the average completion time objective are much more important in latency-sensitive applications that are sensitive to TCP packet ordering (e.g., e.g., etc.).

A classic problem, flow shop [16], inspires our work. Flow shop is an ordered middlebox set, where each flow needs to be processed in a fixed order. We are given a set of unsplittable flows that are processed in a fixed order. We are given a set of unsplittable flows that are processed in a fixed order. We are given a set of unsplittable flows that are processed in a fixed order. We are given a set of unsplittable flows that are processed in a fixed order.

Fig. 2: Different scheduling orders.

Fig. 3: A service chain.

Middleboxes

\[
\begin{array}{c|c|c|c}
\text{Flows} & m_1 & m_2 & m_3 \\
\hline
f & 3 & 4 & 5 \\
\hline
f' & 4 & 3 & 2 \\
\end{array}
\]

Link transmission delay 
\[
d_1 = 1 \quad d_2 = 2
\]

Processing time

\[
f \quad f'
\]

\[
f \quad f'
\]

\[
f \quad f'
\]

f before f' \quad t = 10

f' before f \quad t' = 12

prolong

\[
f \quad f'
\]

\[
f \quad f'
\]

\[
f \quad f'
\]
3. A Service Chain with Two Middleboxes

- **Objective:** minimizing makespan

- **Solution**
  - Two Set Order Schedule (TSOS)

- **Solution steps**
  - Sort flows in decreasing order of $p_2 - p_1$

- **Insight**
  - Inspired by the classic flow shop\[^1\] problem
    - Optimal solution for two machines
  - Make the second middlebox not idle
  - Smallest completion time extension for the last flow

3. A Service Chain with Two Middleboxes

- **Objective**: minimizing average completion time
- **Solution**
  - Pairwise Schedule (PS)
- **Solution steps**
  - Sort flows in increasing order of \( \max_f \{p^f_1, p^f_2\} \)
  - For flows with same \( \max_f \{p^f_1, p^f_2\} \)
    - Select flows with \( \max_f \{p^f_2 - p^f_1\} \) and \( \max_f \{p^f_1 - p^f_2\} \) as a pair
- **Pair flows and processing order illustrations**

(a) A pair.
(b) A pair with a delay.
4. A Service Chain with Multiple Middleboxes

- **Objective**: minimizing makespan
- **Problem complexity**: NP-hard
- **Solution**
  - Slope Heuristic Algorithm (SHA)
- **Solution insights**
  - Cut the service chain into two same-length parts
  - Each part as a “new” middlebox
  - Modification of processing times
  - Apply our proposed Alg. TSOS
- **Illustration**
  
  ![Diagram](image)

(a) Two ordered services. (b) Multiple ordered services.
4. A Service Chain with Multiple Middleboxes

- **Objective:** minimizing average completion time
- **Problem complexity**
  - Harder than makespan
  - NP-hard
- **Solution**
  - Pairwise Heuristic Schedule (PHS)
- **Solution insights**
  - Cut the service chain into two same-length parts
  - Each part as a “new” middlebox
  - Modification of processing times
  - Apply our proposed Alg. PS
5. Simulation

- **Comparison algorithms**
  - Random
    - Rank flow randomly
  - SPT
    - Rank flows by total processing times in increasing order
  - LPT
    - Rank flows by total processing times in decreasing order

- **Our algorithms**

<table>
<thead>
<tr>
<th></th>
<th>Two middleboxes</th>
<th>Multiple middleboxes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makespan</td>
<td>TSOS</td>
<td>PS</td>
</tr>
<tr>
<td>Avg completion time</td>
<td>SHA</td>
<td>PHS</td>
</tr>
</tbody>
</table>
Simulation settings

- Facebook data center flow distribution
  - #flow: ranging from 1000 to 6000

- Service chain
  - Include two or six middleboxes
  - Flow processing time ranging from 2 to 10

- Link transmission delay
  - Ranging 1 to 10
Simulation Results

A service chain with two middleboxes

- TSOS achieves the smallest makespan because of its optimality in a service chain with only two middleboxes
- PS has the lowest average completion time
- The total processing time is important for minimizing the average completion time
Simulation Results

A service chain with six middleboxes

- Performance difference is not obvious
- Average completion time is larger than two middleboxes with the same number of flows
- SHA performs best in makespan while PHS is the best in average completion time
6. Conclusion and Future Work

- Flow contention on the same service chain
- Objectives on flow completion times
  - Makespan
  - Average completion time
- Solutions
  - With only two middleboxes
    - optimal solutions
  - With multiple (>2) middleboxes
    - heuristic solutions
- Future Work
  - Performance-guaranteed solution
  - Statistic processing time model
Questions contact:
Yang Chen (yang.chen@temple.edu)