Improving MPTCP Performance by Enabling Sub-Flow Selection over a SDN Supported Network

Subhrendu Chattopadhyay[1] Samar Shailendra[2]
Sukumar Nandi[1] Sandip Chakraborty[3]

[1]IIT Guwahati , [2]TCS Research & Innovation , [3]IIT Kharagpur

October 16, 2018

Organization

A Motivating Statistics

- *"By 2021, 94 percent of workloads and compute instances will be processed by cloud data centers; 6% will be processed by traditional data centers"*[1]
  - CDN uses data centers
- Demands high bandwidth requirement for data centers
  - Data center topology allows multiple paths between nodes
  - Can exploit bandwidth aggregation
  - Bandwidth aggregation in data link layer causes management issues
- Bandwidth aggregation in transport layer
  - Multipath TCP (MPTCP)[2]

---

[1] *VNI Forecast Highlights Tool.* https://www.cisco.com/c/m/en_us/solutions/service-provider/vni-forecast-highlights.html.

[2] Alan Ford et al. *Architectural guidelines for multipath TCP development*. Tech. rep. IETF, RFC6824, 2011.

MPTCP Basics

- Advantages[3]
  - Improve throughput by aggregating bandwidth
  - Do no harm to the competing flows (TCP, SCTP etc.)
  - Balance congestion by offloading data via less congested paths
  - TCP like API for application transparency.[4]

[3] Costin Raiciu, Mark Handley, and Damon Wischik. *Coupled congestion control for multipath transport protocols*. Tech. rep. IETF, RFC6356, 2011.

[4] *MPTCP Application Interface Considerations*. `https://tools.ietf.org/html/draft-ietf-mptcp-api-07`.

MPTCP Basics

- Advantages[3]
  - Improve throughput by aggregating bandwidth
  - Do no harm to the competing flows (TCP, SCTP etc.)
  - Balance congestion by offloading data via less congested paths
  - TCP like API for application transparency.[4]

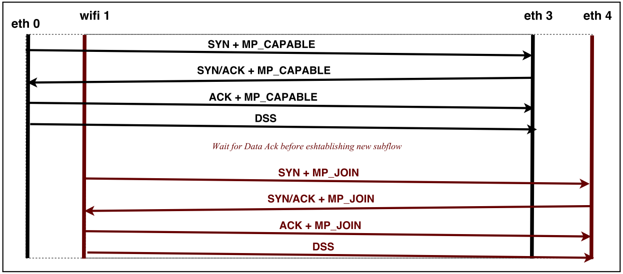

Figure: MPTCP connection initiation

MPTCP Architecture

Modules of MPTCP

- Path manager
  - **Full-Mesh**
  - ndiffports
- Segment scheduler
  - Round robin
  - **Lowest RTT first**
- Congestion control
  - LIA
  - OLIA
  - **BALIA**

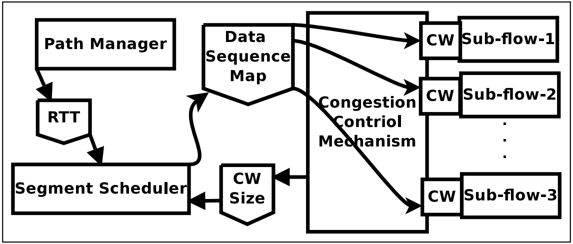

Figure: MPTCP Modules

Initial Experimentations

Test parameter setting (Previous work[5])

- MPTCP v0.90
- Full-mesh
- Lowest RTT first
- BALIA congestion control
  - Revisit: MPTCP objective
    - TCP friendliness
    - Responsiveness towards network changes
  - BALIA Pareto optimizes MPTCP principle

[5] Subhrendu Chattopadhyay et al. "Primary Path Effect in Multi-Path TCP: How Serious Is It for Deployment Consideration?". In: *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Mobihoc '17. Chennai, India: ACM, 2017, 36:1–36:2. ISBN: 978-1-4503-4912-3.

Initial Experimentations

- Variable no. of sub flows used
  - Sub flows have diversified path characteristics (Delay, Bandwidth)
- Bandwidth diversity
  - Increasing path bandwidths difference
    - **Observations:** Increase in total bandwidth pool
    - **Observations:** The average throughput decreases.
    - **Observations:** Increases no. of out of order segments
    - **Observations:** Increases file download time
- Delay diversity
  - Increasing path RTT difference
    - **Observations:** Decreases average throughput of MPTCP.
    - **Observations:** Increases no. of out of order segments
    - **Observations:** Increases file download time

Initial Experimentations

Understanding the results of[5]:

- Sub-flows with high disparity in end-to-end delay and bandwidth results in large number of out of order segments
- Increase in out of order segments results performance degradation due to spurious retransmission
- Full-mesh path manager is sub-optimal.

Analysis:

Out-of-order segments are the root cause. It creates *"HOL blocking"*. HOL blocking causes spurious retransmissions.

[5] Subhrendu Chattopadhyay et al. "Primary Path Effect in Multi-Path TCP: How Serious Is It for Deployment Consideration?". In: *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Mobihoc '17. Chennai, India: ACM, 2017, 36:1–36:2. ISBN: 978-1-4503-4912-3.

Problem Statement

What can we do about it?

Out-of-order segments can be estimated by the receiver buffer size. Reduction is receiver buffer signifies reduction in out-of-order segments. So, we control the size of receiver buffer size by choosing the subset of available sub-flows.

What we have?

- Set of sub-flows ($\mathcal{S} = \{S_1, S_2 \ldots S_n\}$)
- Delay characteristics of S_i ($Pr_i(X = r) = \Psi(\mu_i, \sigma_i, 0, \infty; X = r)$) Assume Gaussian
- Gross characteristics of S_i ($Q_i = \{b_i, l_i, \mu_i, \sigma_i\}$)

Problem Statement

What can we do about it?
Problem Formulation:
Given $\mathcal{S}$ sub-flows between source and destination and the path parameters $\vec{Q} = \{Q_i\}$ of sub-flows, we would like to obtain a sub-flow selection matrix I, such that the following optimization problem is solved.

$$\underset{I}{\text{maximize}} \quad Avg_{Th}(I)$$

subjected to:

$$RL(I) \leq RL_{max}$$

Problem Statement

What can we do about it?
Problem Formulation:

$$\underset{I}{\text{maximize}} \quad Avg_{Th}(I)$$

subjected to:

$$RL(I) \leq RL_{max}$$

- **Question 1:** How to find $Avg_{Th}(I)$ and $RL(I)$?
  - Using formal modeling of MPTCP system
  - Discrete Time Markov Chain (DTMC)
- **Question 2:** Who solves the optimization problem?
  - Hosts do not have end-to-end characteristics.
  - Use SDN.

DTMC

- Throughput modeling requires knowledge of congestion control method.
- We use BALIA

$$Y_i(t) = \frac{w_i(t)}{r_i} \qquad \alpha_i(t) = \frac{\max_k\{Y_k(t)\}}{Y_i(t)}$$

  - Algorithm:

$$w_i' = \begin{cases} \frac{Y_i(t)}{r_i\left(\sum_k Y_k(t)\right)^2}\left(\frac{1+\alpha_i(t)}{2}\right)\left(\frac{4+\alpha_i(t)}{5}\right) & \text{Success} \\ \frac{w_i(t)}{2}\min\{\alpha_i(t), 1.5\} & \text{Failure} \end{cases}$$

- Oscillation factor[a] increases responsiveness, but aggressive.
- Aggressiveness factor[b] controls the TCP friendliness.

[a] Oscillation factor: $Y_i(t)$

[b] Aggressiveness factor: $\alpha_i(t)$

DTMC

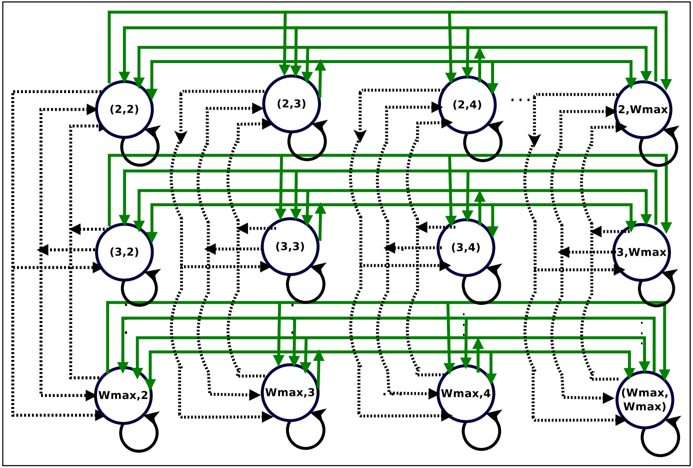

Figure: Markov Model for a MPTCP with 2 Sub-Flows

DTMC

States:

- CW size tuple
- Event CW change as state transition

Transition events and Probabilities:

- Successful transfer of segment via S_i (SS_i)
  - If $\max\{Y_k\} = Y_i$ (SS_{max_i})
  - If $\max\{Y_k\} \neq Y_i$ (SS_{max_m})
- Unsuccessful transfer of segment via S_i (SL_i)
  - If $\max\{Y_k\} = Y_i$ (SL_{max_i})
  - If $\max\{Y_k\} \neq Y_i$ (SL_{max_m})

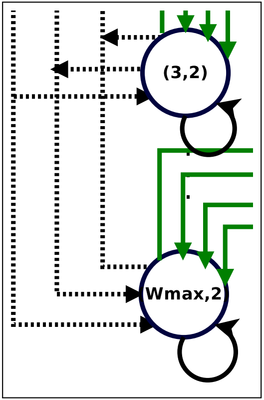

Figure: DTMC partial

DTMC

- **Model Outcome:**
  - Stationary distribution of DTMC
  - Average congestion window size
  - Average throughput and Average receiver buffer length

Model Verification

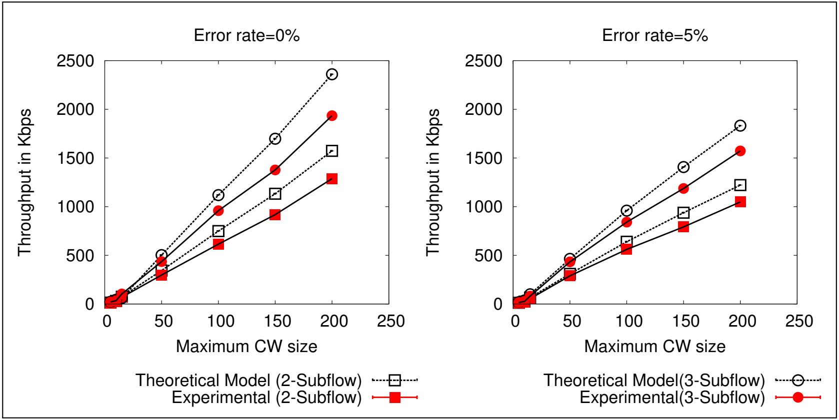

Figure: Throughput Comparison

Model Verification

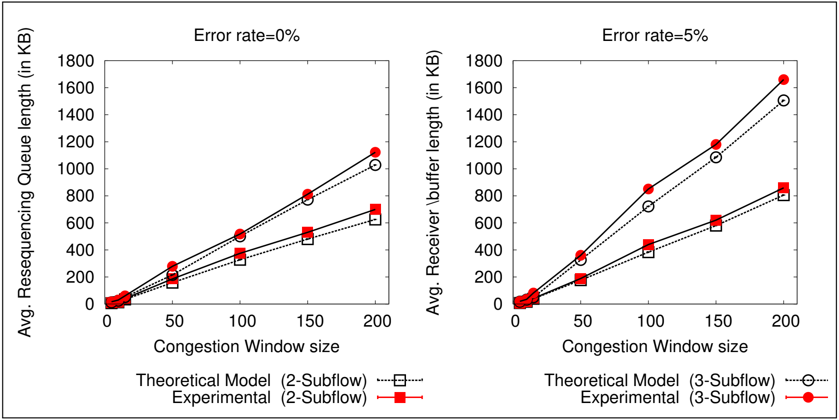

Figure: Receiver Buffer Size Comparison

Revisit Sub-flow Selection & Heuristic

Problem statement:
Given $\mathcal{S}$ sub-flows between source and destination and the path parameters $\vec{Q} = \{Q_i\}$ of sub-flows, we would like to obtain a sub-flow selection matrix I, such that the following optimization problem is solved.

$$\underset{I}{\text{maximize}} \quad Avg_{Th}(I)$$

subjected to:

$$RL(I) \leq RL_{max}$$

- 0-1 knapsack problem[6] and NP-hardness
  - Searching for a heuristic

[6] Harvey M. Salkin and Cornelis A. De Kluyver. "The knapsack problem: A survey". In: *Naval Research Logistics Quarterly* 22.1 (1975), pp. 127–144.

Revisit Sub-flow Selection & Heuristic

Algorithm 1 Heuristic for sub-flow selection (Pseudo code)

1: Input: Sub-flow path quality vector;
2: Output: Sub-flow selection;
3: Sort sub-flows based on high effective bandwidth $(b_i(1 - l_i))$ and low RTT (μ_i) (i.e $b_i(1 - l_i) + \frac{1}{\mu_i}$);
4: **for** $S_i \in \mathcal{S}$ **do**
5: Select sub-flow if calculated receiver buffer length obtained from DTMC is less than RL_{max}
6: **end for**
7: **return** $\vec{l}$

Revisit Sub-flow Selection & Heuristic

Algorithm 1 Heuristic for sub-flow selection (Algorithm)

Input: $\vec{Q}$;
Output: $\vec{I}$;
$\forall i : I_i \leftarrow 0$;
Sort $\vec{Q}$ based on $T_i \leftarrow b_i(1 - l_i) + \frac{1}{\mu_i}$;
{High effective bandwidth ($b_i(1 - l_i)$) and low RTT (μ_i) gets priority}
Find $\max_i(T_i)$; $I_i \leftarrow 1$;
for $j \leftarrow 2$ **to** n **do**
  $\vec{X} \leftarrow \vec{Q} \circ I$;
  $\mathcal{A} \leftarrow Avg_{Th}(\vec{X})$;
  $\mathcal{R} \leftarrow RL(\vec{X})$
  if $\mathcal{R} \leq RL_{max}$ **then**
    $I_j \leftarrow 1$;
  end if
end for
return $\vec{I}$

Implementation

We develop a SDN control plane application. [6]

- Tools used
  - Open-source MPTCP kernel module[7]
  - *Open vSwitch*[8]
  - *Mininet*[9]
  - *Tinydb*[10]
  - *POX controller*[11]
    - "flow_stat"
    - "L3_learning"
    - "host_tracker"

---

[6] https://github.com/subhrendu-subho/SDN_pathmanager

[7] *MultiPath TCP - Linux Kernel implementation*. https://multipath-tcp.org.

[8] OVS. *Open vSwitch*. http://openvswitch.org/.

[9] B Lantz et al. *Mininet-an instant virtual network on your laptop (or other pc)*. 2015.

[10] *Introduction-; TinyDB 3.2.1 documentation*. http://tinydb.readthedocs.io/en/latest/intro.html.

[11] *POX*. https://openflow.stanford.edu/display/ONL/POX+Wiki.

Implementation

We develop a SDN control plane application. [6]

- Tools used
- Development
  - MPTCP Path manager module
  - SDN application for sub flow selection

[6] https://github.com/subhrendu-subho/SDN_pathmanager

Implementation

We develop a SDN control plane application. [6]

- Tools used
- Development
- Event Handlers
  - Topology Update:
    - Invokes sub-flow selection module
    - Pro-actively notify path manager framework.
  - Packet In:
    - Find all available paths.
    - Invokes sub-flow selection module.

[6] https://github.com/subhrendu-subho/SDN_pathmanager

Results

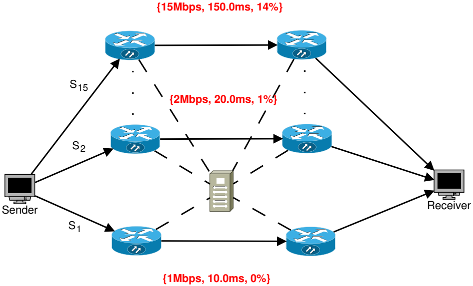

Figure: Topology

- 15 parallel paths
- The sender generates MPTCP supported HTTP flows destined towards receiver host.
- Traffic generated by transferring 100*MB* file.

Results

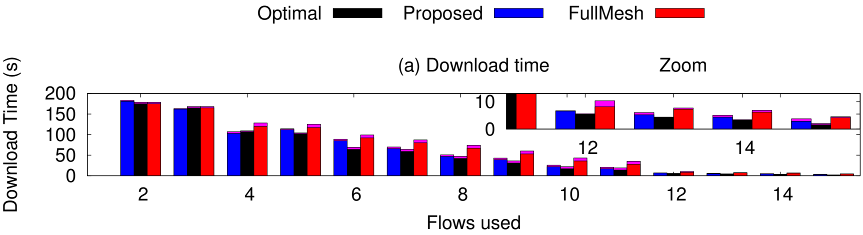

Figure: Flow Completion Time

Optimal Proposed FullMesh

- **Observations:**
  - Proposed method provides better performance.

Results

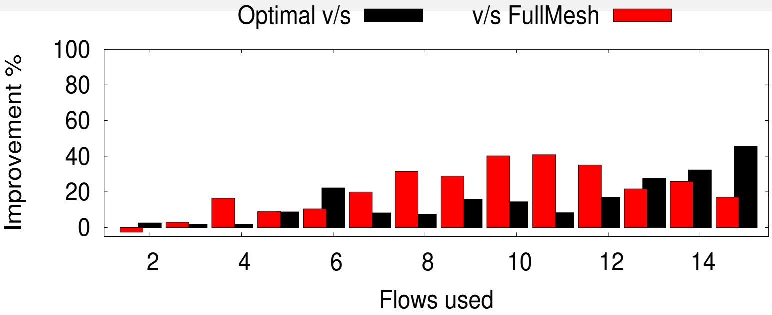

Figure: Flow Completion Time Comparison

- **Observations:**
  - Full mesh is better for 2 sub flows
  - Increased diversity provides better performance
  - Too much diversity reduces performance gain

Results

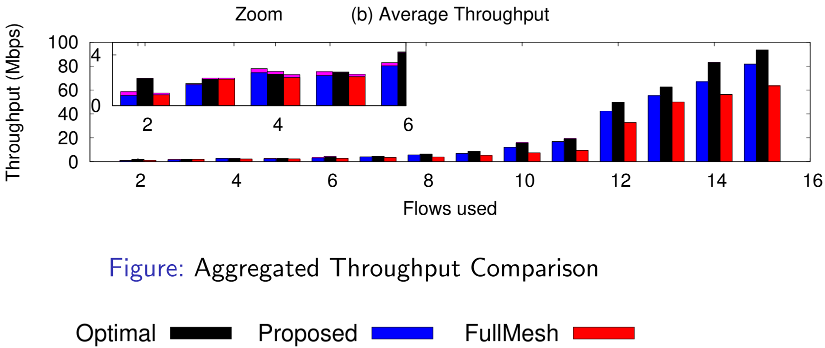

Figure: Aggregated Throughput Comparison

- **Observations:**
  - Effective increase in throughput from >6 sub-flows

Results

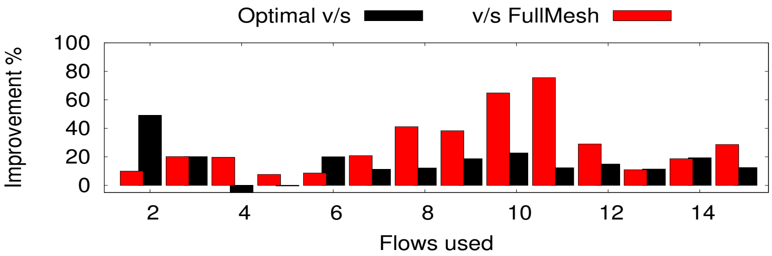

Figure: Aggregated Throughput Comparison

- **Observations:**
  - Throughput increase is closely follows optimal behaviour

Results

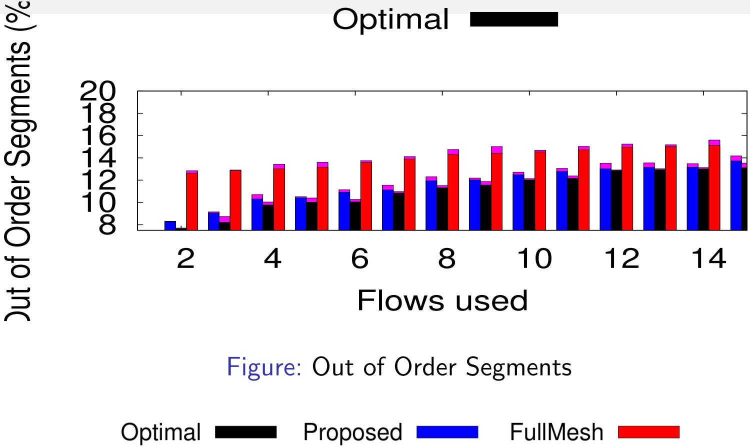

Figure: Out of Order Segments

- **Observations:**
  - Near optimal behaviour for proposed solution

Results

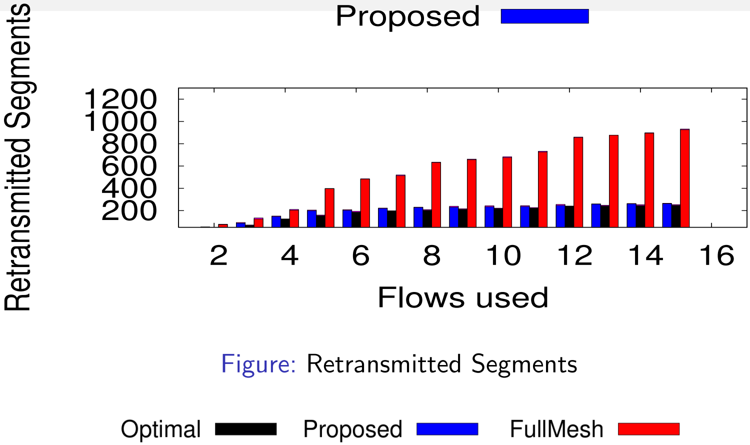

Figure: Retransmitted Segments

- **Observations:**
  - Near optimal behaviour for proposed solution

Results

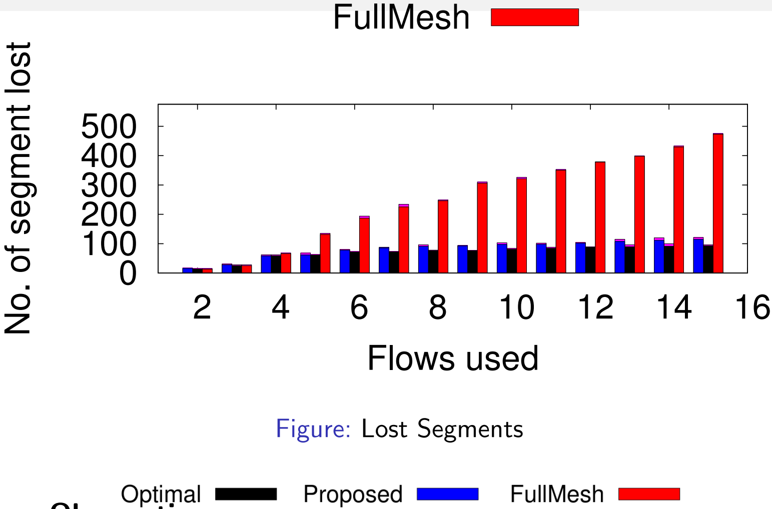

Figure: Lost Segments

- **Observations:**
  - Near optimal behaviour for proposed solution

Results

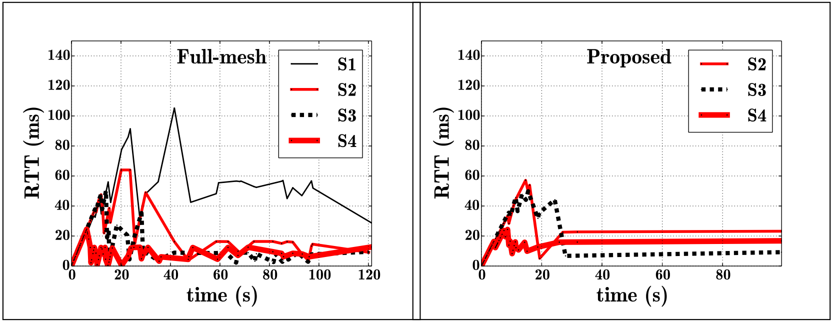

Figure: RTT Variations

- **Observations:**
  - Less fluctuations between the inter sub flow segments

Results

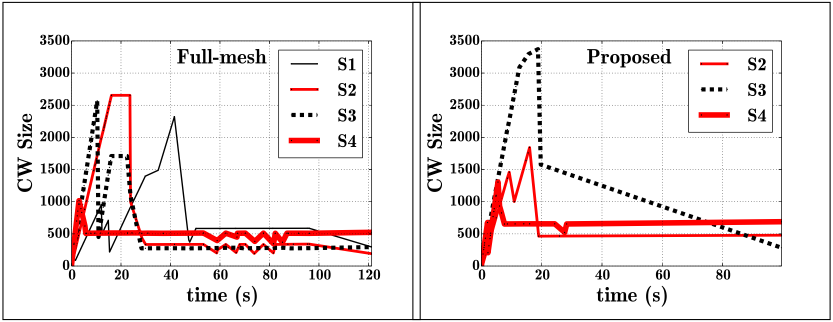

Figure: Congestion Window Variation

- **Observations:**
  - Can reach higher congestion window size due to less spurious transmission
  - Increase in effective aggregated throughput

Summary

- We formulate an irreducible and aperiodic DTMC to model the aggregated throughput prediction of a MPTCP flow with the end-to-end path characteristics of a given set of sub-flows.
- Based on the predicted throughput from the estimator model, we develop an optimization framework to find out the optimal set of sub-flows that can maximize the aggregated throughput for a given MPTCP flow.
- The SDN controller executes this optimization framework and schedules the sub-flows accordingly.

Conclusion

- MPTCP sub-flow management framework for enterprise data center network.
- Increases in-order delivery of segments and prevents HOL blocking
- Closely approximates the underlying *NP*-hard problem
- Future Work:
  - Can we generalize it for multi-homed network?
  - Can we use distributed SDN control plane application?

Thank You