Congestion Control in Converged Ethernet with Heterogeneous and Time-Varying Delays

Wenxue Cheng, Wanchun Jiang (CSU), Tong Zhang,
Bo Wang, Kun Qian, Fengyuan Ren

NNS Group @ Tsinghua University

IEEE/ACM IWQoS 2017, VILANOVA I LA GELTRÚ, SPAIN

Background

Converged Ethernet (CE)

Storage Area Networks (SAN)
Deterministic, in-order, guaranteed delivery to be sent to/from storage devices.

Local Area Networks (LAN)
Traditional TCP/IP based Ethernet network for best effort data communications.

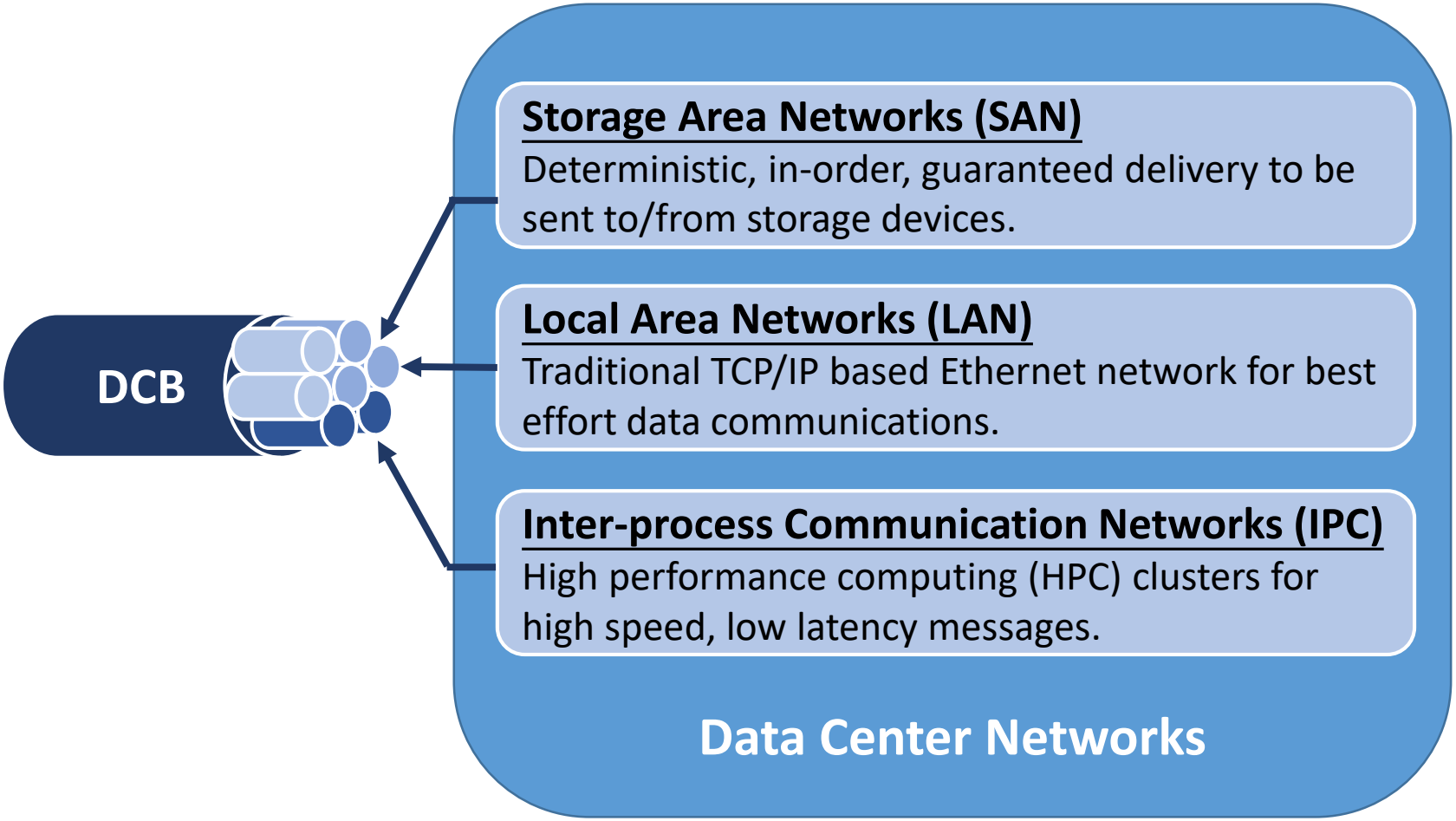
Inter-process Communication Networks (IPC)
High performance computing (HPC) clusters for high speed, low latency messages.

Data Center Networks

Background

Converged Ethernet (CE)

Enhance Ethernet as a unified fabric in data center.



Background

Congestion Control in CE

- Indispensable for losslessness and high utilization.
- Deployed in link layer

Background

Congestion Control in CE

- Indispensable for losslessness and high utilization.
- Deployed in link layer

- Quantified Congestion Notification (QCN)
  - Recommended in standard draft of DCB.
  - Implemented in devices.
  - Extended as DCQCN to support RoCEv2

Background

Congestion Control in CE

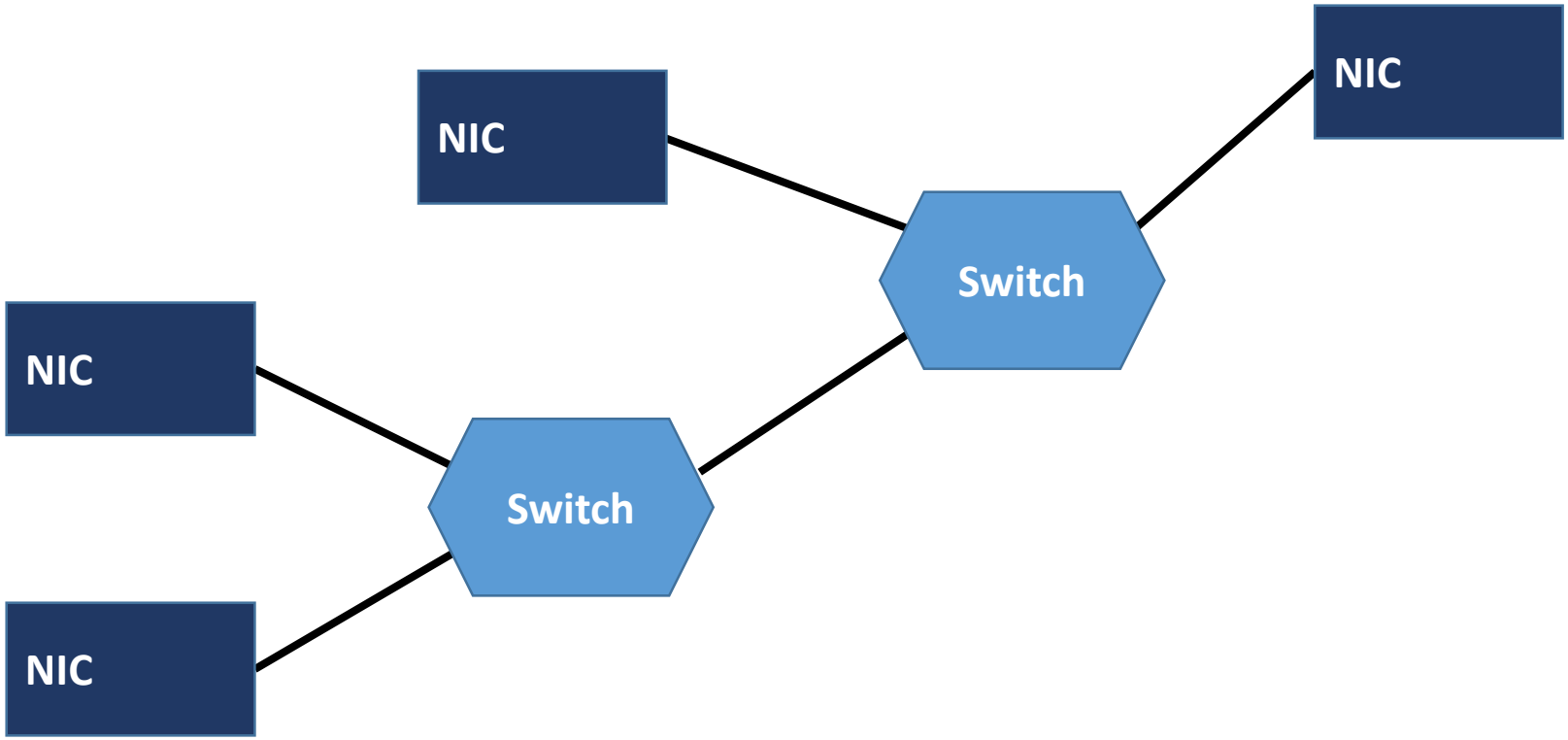
- Indispensable for losslessness and high utilization.
- Deployed in link layer

- Quantified Congestion Notification (QCN)
  - Recommended in standard draft of DCB.
  - Implemented in devices.
  - Extended as DCQCN to support RoCEv2

- Sliding Mode Congestion Control (SMCC) [Infocom'12]
  - Configuration-sensitive issue of QCN
  - Slide mode motion to be robust to the changes of system parameters and network configurations.

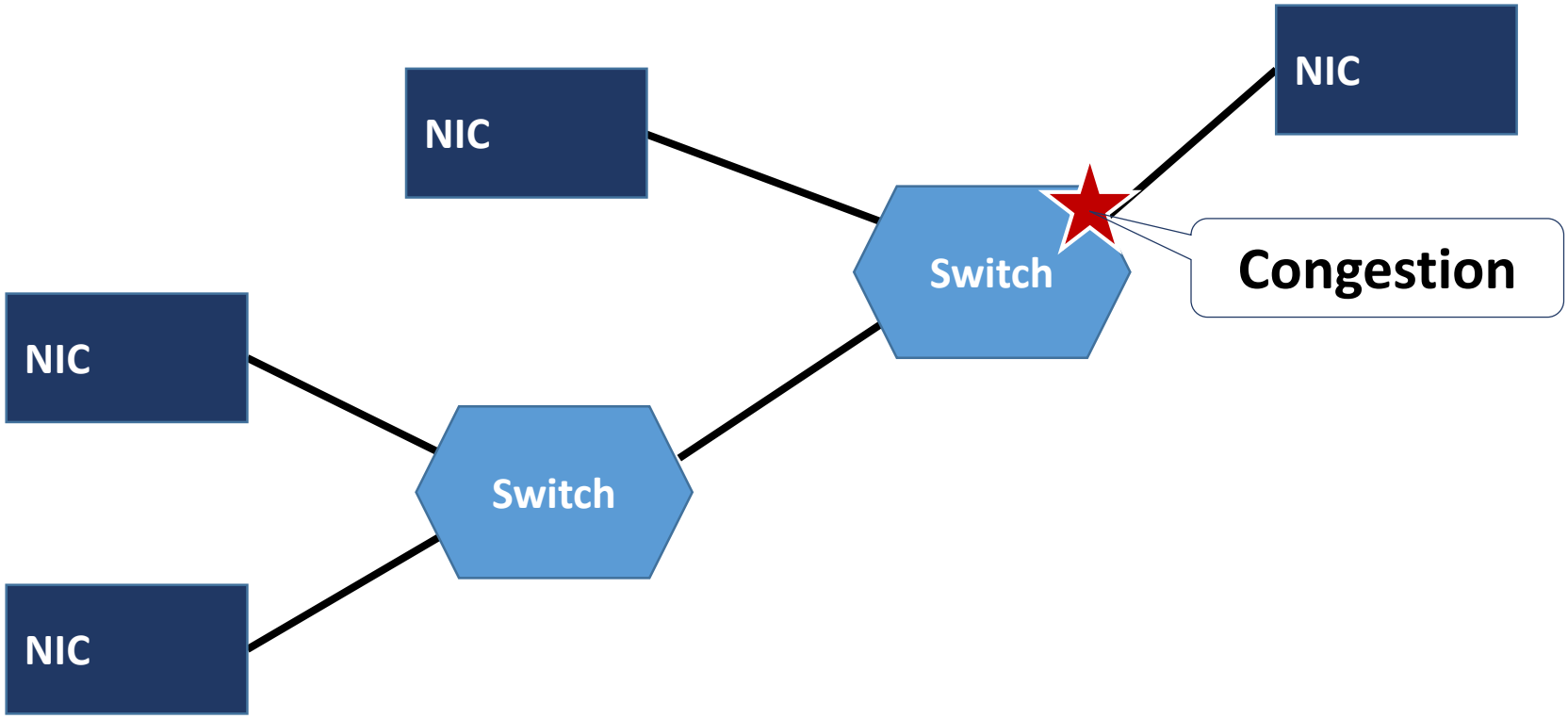
Background

Congestion Control in CE



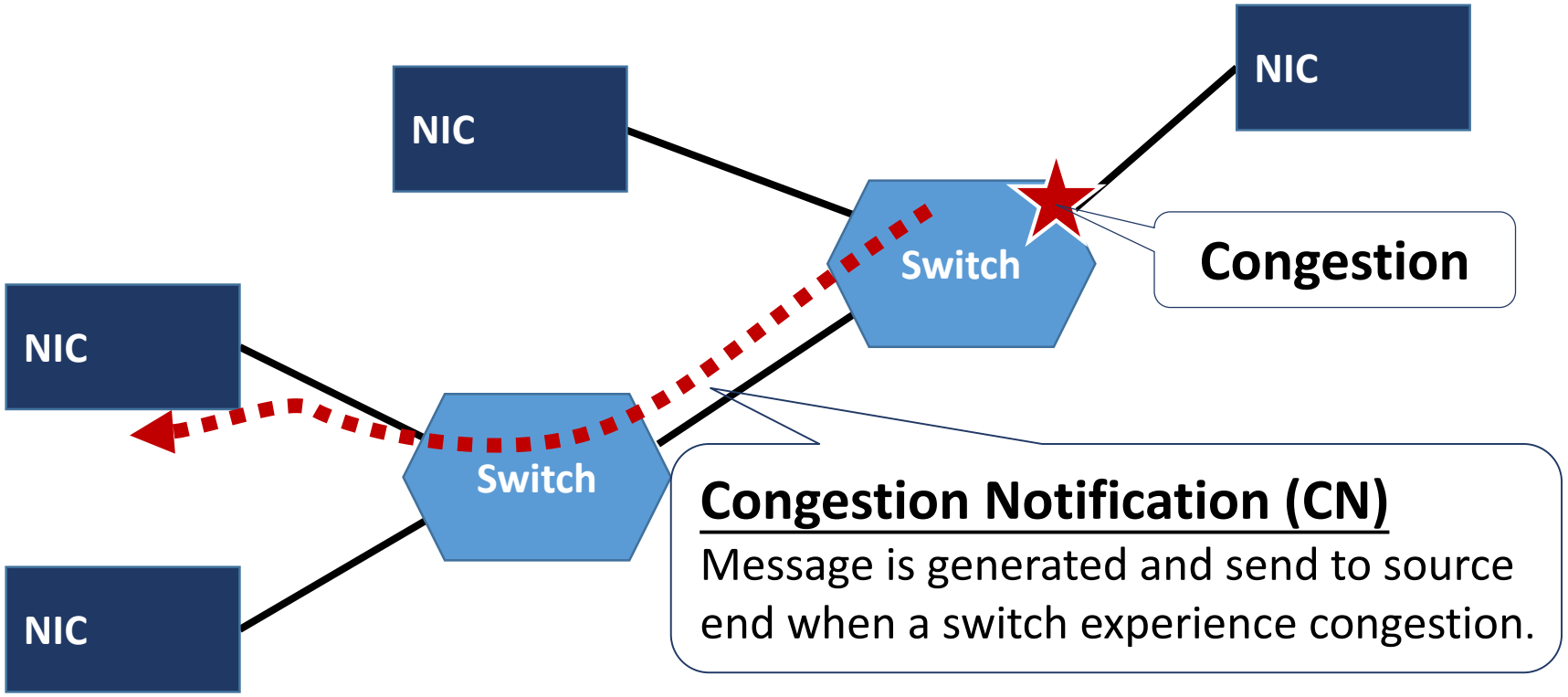
Background

Congestion Control in CE



Background

Congestion Control in CE

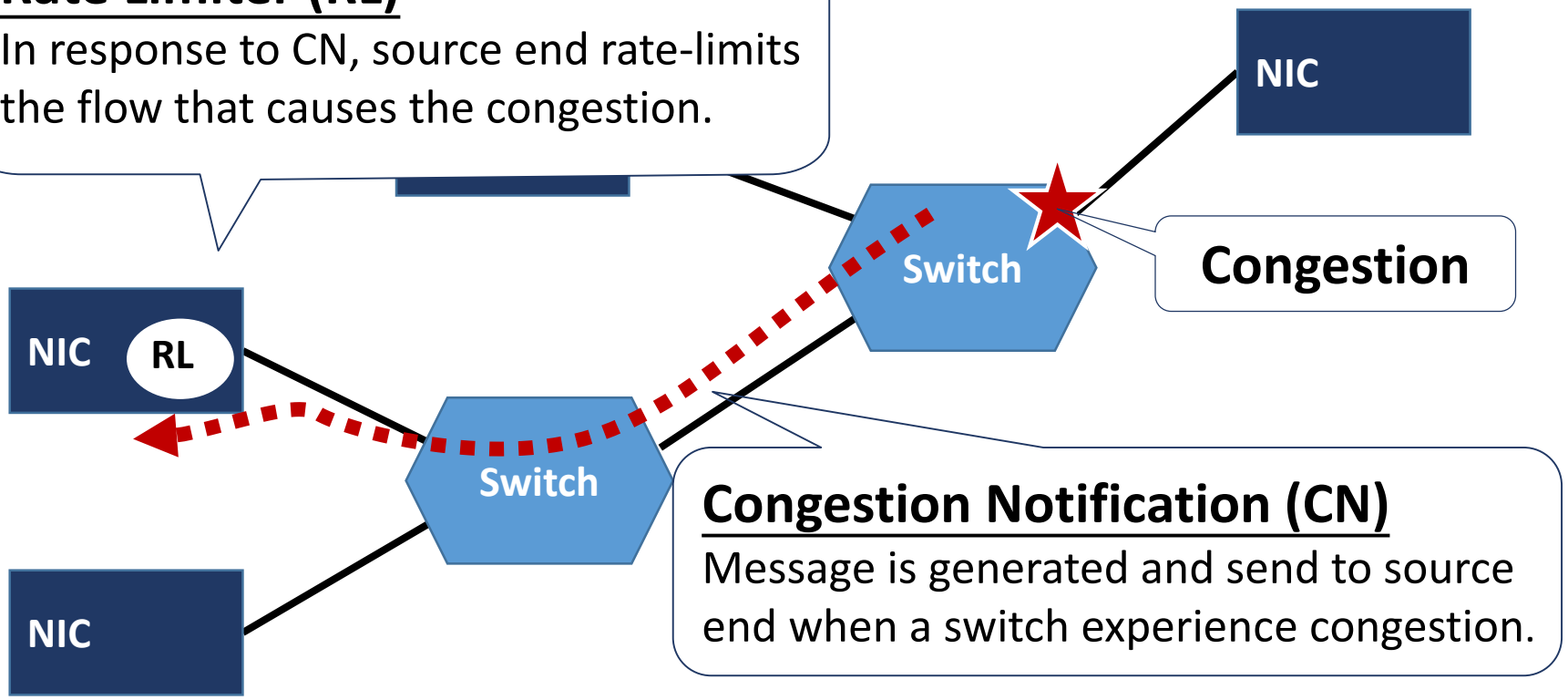


Background

Congestion Control in CE

Rate Limiter (RL)

In response to CN, source end rate-limits the flow that causes the congestion.



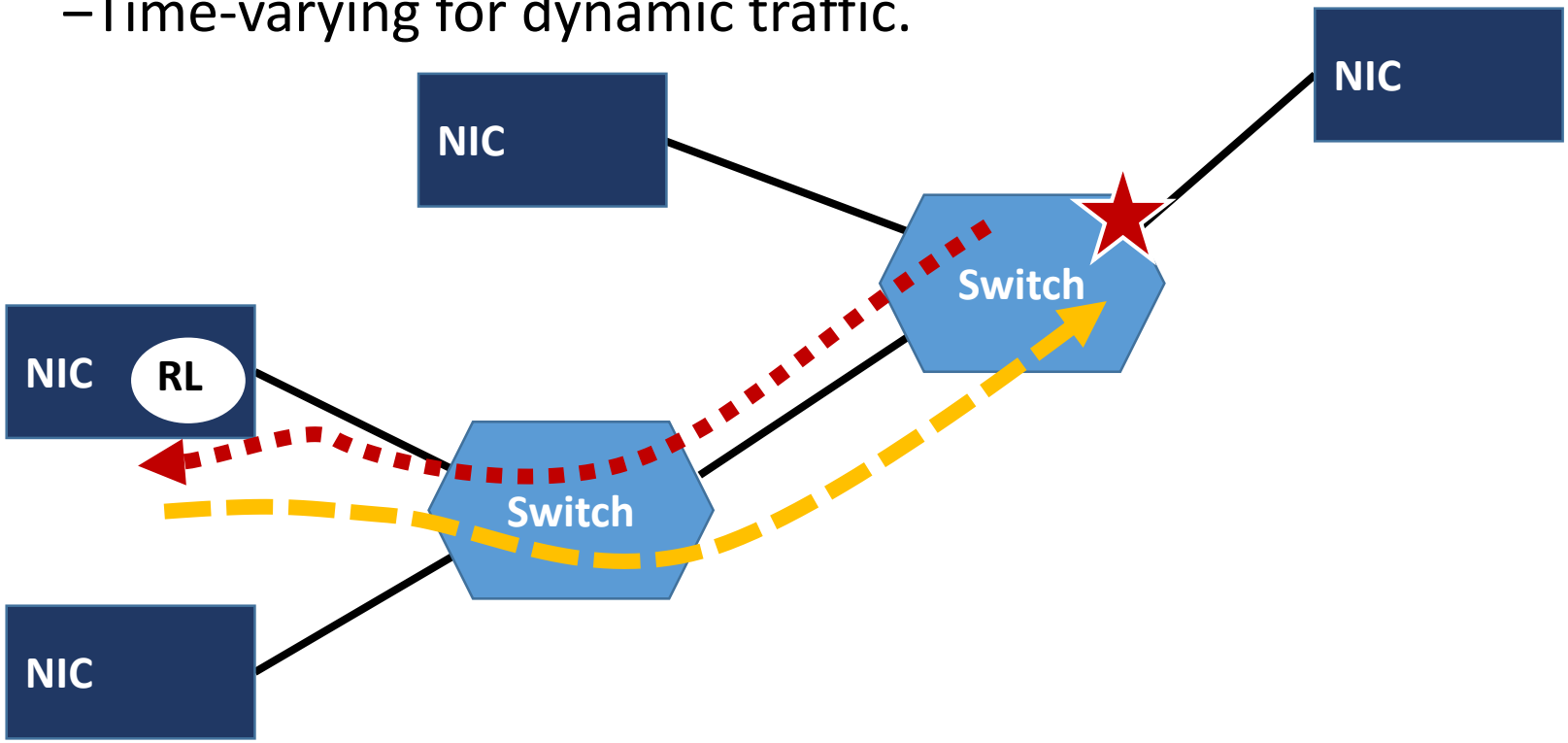
Congestion Notification (CN)

Message is generated and send to source end when a switch experience congestion.

Background

Heterogeneous and Time-Varying Delays

- Delays in the feedback loop
  - –Heterogeneous for different sources.
  - –Time-varying for dynamic traffic.

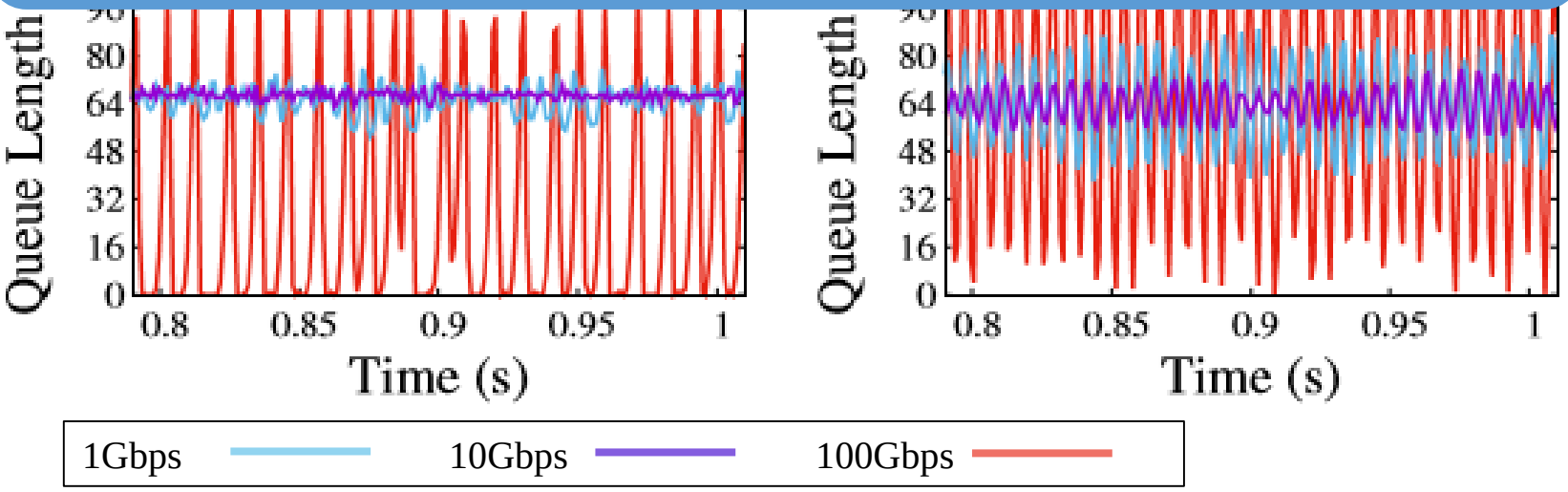


Background

Negative Impact of Delays

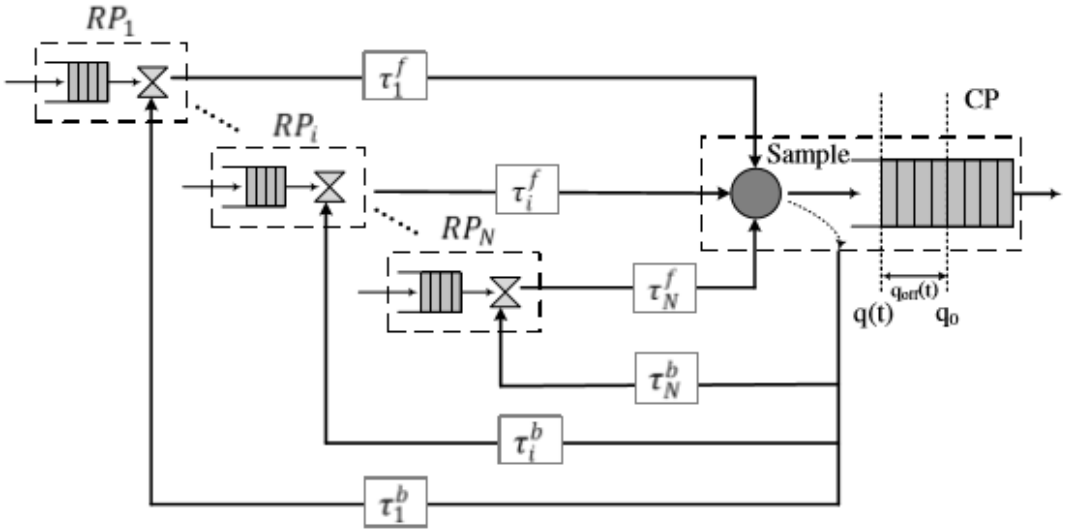
- Obstruct sources from obtaining the right congestion status timely
  - Error rate adjustment ➔ Overflow or Underflow
  - Heterogeneous and time-varying delays ➔ Complex
  - High speed Ethernet link ➔ More serious

Mitigate the negative impacts of delays



Model

Heterogeneous and Time-Varying Delays



The congested switch samples the packets

➔ Take the sample period T as slot

The k_{th} feedback is sent to source i

- Backward delay: $m_i^b(k) = \lceil \tau_i^b(k)/T \rceil$
- Forward delay: $m_i^f(k) = \lceil \tau_i^f(k)/T \rceil$
- Total delay: $m_i(k) = m_i^b(k) + m_i^f(k)$

Model

Rate Adjustments

k

Switch generates a feedback

$$F_b(k) = f\left(Q_f(k), Q_v(k), \cdots\right)$$

$k + m_i^b(k)$

Source i adjusts its rate according to $F_b(k)$

$$\Delta r_i\left(k + m_i^b(k)\right) = u\left(F_b(k)\right)$$

$k + m_j^b(k)$

Source j doesn't receive $F_b(k)$

$$\Delta r_j\left(k + m_j^b(k)\right) = 0$$

$k + m$

$m = \max\{m_i(k)\}$, incoming rate at switch

$$\begin{aligned}\Delta R(k+m) &= \sum \Delta r_j\left(k + m - m_j^f(k)\right) \\ &= \sum \Delta r_j\left(k + m_j^b(k)\right) + \varepsilon(k) \\ &= u\left(F_b(k)\right) + \varepsilon(k)\end{aligned}$$

Time/T

$$\varepsilon(k) = \sum \Delta r_j\left(k + m - m_j^f(k)\right) - \sum \Delta r_j\left(k + m_j^b(k)\right)$$

Model

Evolution of Switch Queue

- Q_f: Queue length offset to the target value q_0
- Q_v: Queue length variance

$$\begin{cases} \Delta Q_f(k+m) = Q_v(k+m) \\ \Delta Q_v(k+m) = T\big(u(k)+\varepsilon(k)\big) \end{cases}$$

$$\begin{cases} Q_f(k+m) = Q_f(k) + mQ_v(k) + T\sum_{i=1}^{m} i[u(k-i) + \varepsilon(k-i)] \\ Q_v(k+m) = Q_v(k) + T\sum_{i=1}^{m}[u(k-i) + \varepsilon(k-i)] \end{cases}$$

$$\begin{cases} \widehat{Q_f}(k) = Q_f(k) + mQ_v(k) + T\sum_{i=1}^{m} iu(k-i) \\ \widehat{Q_v}(k) = Q_v(k) + T\sum_{i=1}^{m} u(k-i) \end{cases}$$

$$\begin{cases} \Delta\widehat{Q_f}(k) = \widehat{Q_v}(k) + mT\varepsilon(k-m) \\ \Delta\widehat{Q_v}(k) = T\big(u(k)\big) + T\varepsilon(k-m) \end{cases}$$

Model

New Congestion Detector

$$\begin{cases} \widehat{Q_f}(k) = Q_f(k) + mQ_v(k) + T\sum_{i=1}^{m} iu(k-i) \\ \widehat{Q_v}(k) = Q_v(k) + T\sum_{i=1}^{m} u(k-i) \end{cases}$$

$$\begin{cases} \Delta\widehat{Q_f}(k) = \widehat{Q_v}(k) + mT\varepsilon(k-m) \\ \Delta\widehat{Q_v}(k) = T\big(u(k)\big) + T\varepsilon(k-m) \end{cases}$$

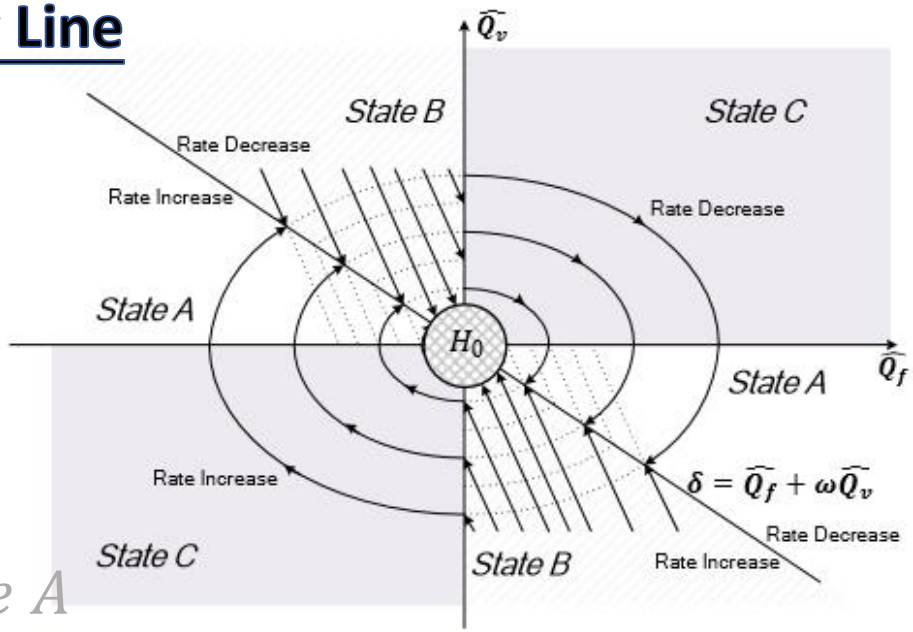
- $\left(\widehat{Q_f}(k), \widehat{Q_v}(k)\right)$ estimates the real evolution of the switch queue length with the impact of delays.
- $\left(\widehat{Q_f}(k), \widehat{Q_v}(k)\right)$ can be calculated at time k.
- $\left(\widehat{Q_f}(k), \widehat{Q_v}(k)\right) = (0,0)$ is the stable state.
- The disturbance ξ(k − m) is limited

Design

Sliding Mode Control Method

Step 1: Determining Boundary Line

- Quadrant I ➔ Overflow
- Quadrant III ➔ Underflow
- Quadrant II/IV
  - $\delta > 0$ ➔ Overflow
  - $\delta < 0$ ➔ Underflow



Step 2: Developing Rules Fulfilling Sliding Mode Motion

- $u(k) = \begin{cases} -a\widehat{Q_f}(k), \ in\ state\ A \\ -b\widehat{Q_v}(k), \ in\ state\ B \end{cases}$
- Satisfy $\delta(k) * \Delta\delta(k) < 0$

Step 3: Developing Reaching Process

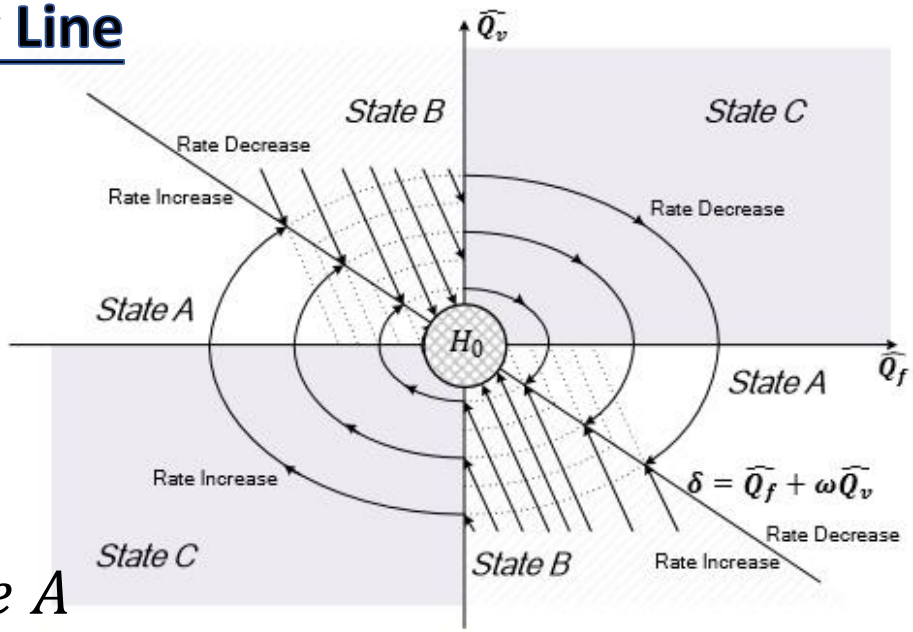
- $u(k) = -c\widehat{Q_f}(k), in\ state\ C$

Design

Sliding Mode Control Method

Step 1: Determining Boundary Line

- Quadrant I ➔ Overflow
- Quadrant III ➔ Underflow
- Quadrant II or IV
  - $\delta > 0$ ➔ Overflow
  - $\delta < 0$ ➔ Underflow



Step 2: Developing Rules Fulfilling Sliding Mode Motion

- $u(k) = \begin{cases} -a\widehat{Q_f}(k), \ in\ state\ A \\ -b\widehat{Q_v}(k), \ in\ state\ B \end{cases}$
- Satisfy $\delta(k) * \Delta\delta(k) < 0$

Step 3: Developing Reaching Process

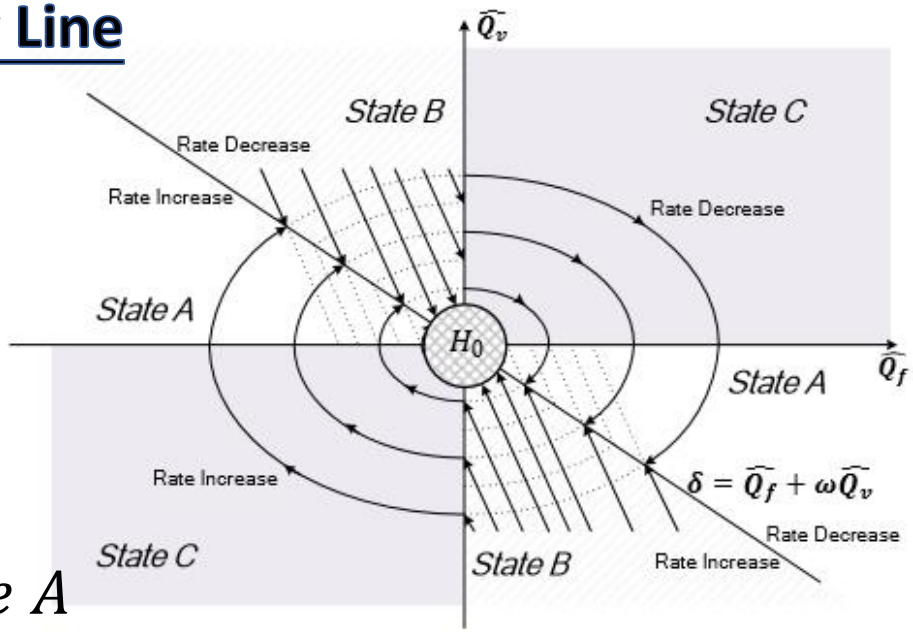
- $u(k) = -c\widehat{Q_f}(k), in\ state\ C$

Design

Sliding Mode Control Method

Step 1: Determining Boundary Line

- Quadrant I ➔ Overflow
- Quadrant III ➔ Underflow
- Quadrant II or IV
  - $\delta > 0$ ➔ Overflow
  - $\delta < 0$ ➔ Underflow



Step 2: Developing Rules Fulfilling Sliding Mode Motion

- $u(k) = \begin{cases} -a\widehat{Q_f}(k), \ in\ state\ A \\ -b\widehat{Q_v}(k), \ in\ state\ B \end{cases}$
- Satisfy $\delta(k) * \Delta\delta(k) < 0$

Step 3: Developing Reaching Process

- $u(k) = -c\widehat{Q_f}(k), in\ state\ C$

Design

Delay-Tolerant Sliding Model (DSM) Congestion Control Scheme

Switch

- $$F_b(k) = \begin{cases} -a\widehat{Q_v}(k), if\ \widehat{Q_v}(k) * \left(\widehat{Q_f}(k) + \omega\widehat{Q_v}(k)\right) < 0 \\ -b\widehat{Q_v}(k), if\ \widehat{Q_f}(k) * \left(\widehat{Q_f}(k) + \omega\widehat{Q_v}(k)\right) < 0 \\ -c\widehat{Q_f}(k), if\ \widehat{Q_f}(k) * \widehat{Q_v}(k) > 0 \end{cases}$$
- $$\begin{cases} \widehat{Q_f}(k) = Q_f(k) + mQ_v(k) + T\sum_{i=1}^{m} iF_b(k-i) \\ \widehat{Q_v}(k) = Q_v(k) + T\sum_{i=1}^{m} F_b(k-i) \end{cases}$$
- $$\begin{cases} m = \inf\{m \in N^+ : m_i(k) \leq m, \forall i, k\} \\ a \to \left[\frac{1}{p\ \max(Q_f + \omega Q_v) + \frac{m(m+1)}{2}} \frac{1}{T}\right]^- \\ b \to \left[\frac{1}{p\ \max(Q_v) + m} \frac{1}{T}\right]^- \\ c \to \left[\min\left\{\frac{1}{p\ \max(Q_v)} \frac{1}{T}, \frac{2}{T}\right\}\right]^- \\ \omega > m + p\ \max(Q_v) - 1 \end{cases}$$

Design

Delay-Tolerant Sliding Model (DSM) Congestion Control Scheme

Switch

- $$F_b(k) = \begin{cases} -a\widehat{Q_v}(k), if\ \widehat{Q_v}(k) * \left(\widehat{Q_f}(k) + \omega\widehat{Q_v}(k)\right) < 0 \\ -b\widehat{Q_v}(k), if\ \widehat{Q_f}(k) * \left(\widehat{Q_f}(k) + \omega\widehat{Q_v}(k)\right) < 0 \\ -c\widehat{Q_f}(k), if\ \widehat{Q_f}(k) * \widehat{Q_v}(k) > 0 \end{cases}$$
- $$\begin{cases} \widehat{Q_f}(k) = Q_f(k) + mQ_v(k) + T\sum_{i=1}^{m} iF_b(k-i) \\ \widehat{Q_v}(k) = Q_v(k) + T\sum_{i=1}^{m} F_b(k-i) \end{cases}$$
- $$\begin{cases} m = \inf\{m \in N^+ : m_i(k) \le m, \forall i, k\} \\ a \to \left[\frac{1}{p\ \max(Q_f + \omega Q_v) + \frac{m(m+1)}{2}} \frac{1}{T}\right]^- \\ b \to \left[\frac{1}{p\ \max(Q_v) + m} \frac{1}{T}\right]^- \\ c \to \left[\min\left\{\frac{1}{p\ \max(Q_v)} \frac{1}{T}, \frac{2}{T}\right\}\right]^- \\ \omega > m + p\ \max(Q_v) - 1 \end{cases}$$

NIC

- $r \leftarrow r + F_b$

Properties

Complexity

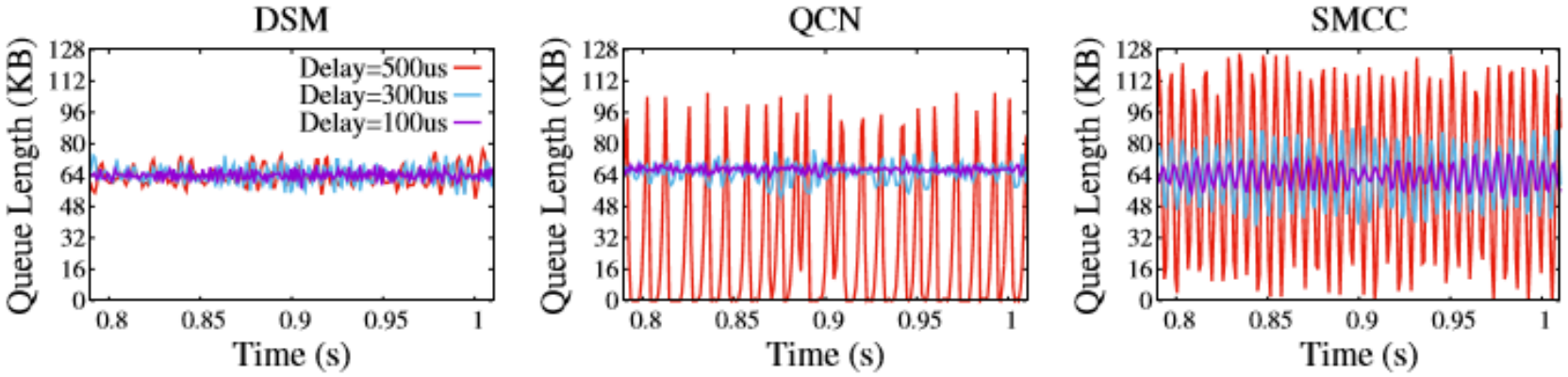
- Switch
  - Spatial complexity: $O(M)$
    - Memorize the last m feedbacks
  - Computing complexity: $O(1)$
    - Iteration Method

$$\begin{cases} S_1(k) = \sum_{i=1}^{m} F_b(k-i) \\ S_2(k) = \sum_{i=1}^{m} iF_b(k-i) \end{cases} \rightarrow \begin{cases} \widehat{Q_f}(k) = Q_f(k) + mQ_v(k) + TS_2(k) \\ \widehat{Q_v}(k) = Q_v(k) + TS_2(k) \\ S_1(k+1) = S_1(k) + F_b(k-m) \\ S_2(k+1) = S_1(k+1) + S_2(k) - mF_b(k-m) \end{cases}$$

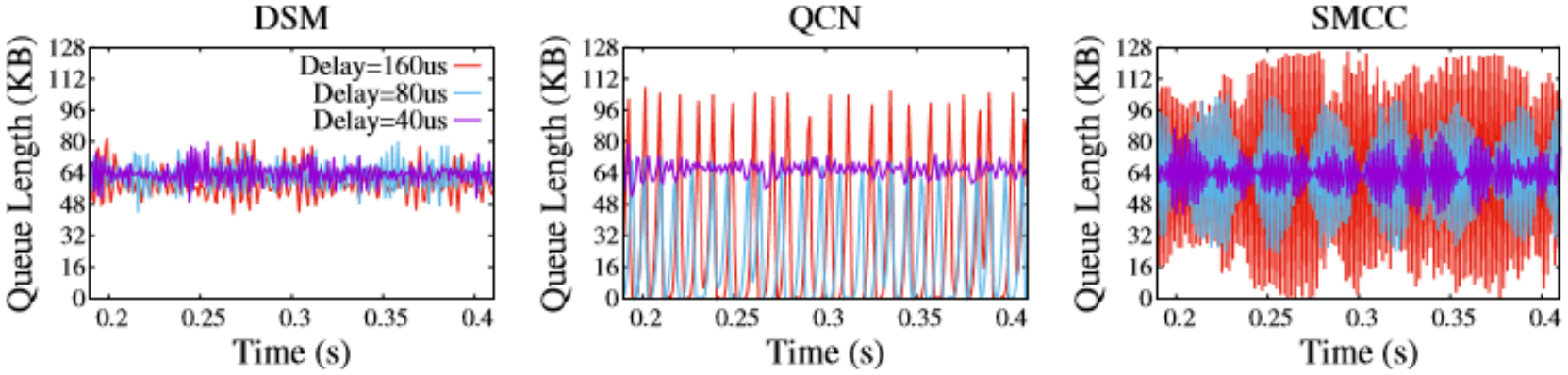
- NIC
  - Complexity: $O(1)$

Properties

Stability



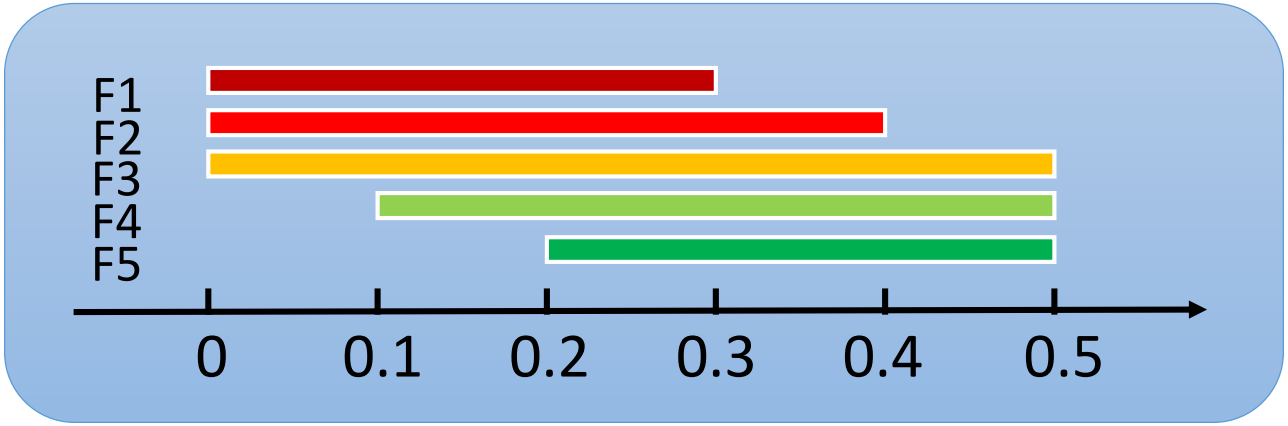
(a) Queue length evolutions in DSM, QCN and SMCC in 10*Gbps* Ethernet.

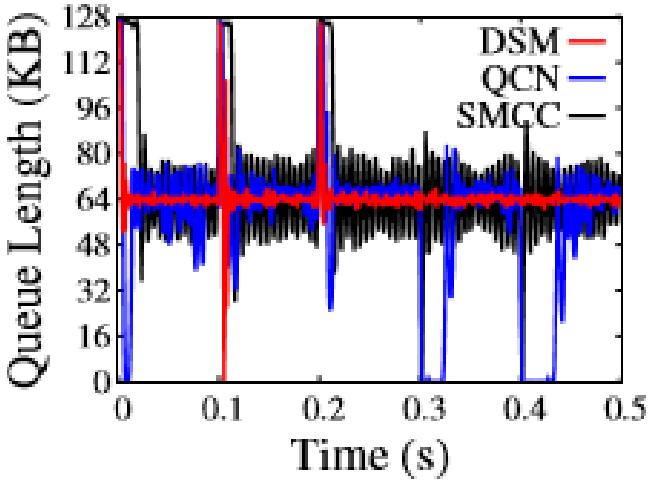


(b) Queue length evolutions in DSM, QCN and SMCC in 100*Gbps* Ethernet.

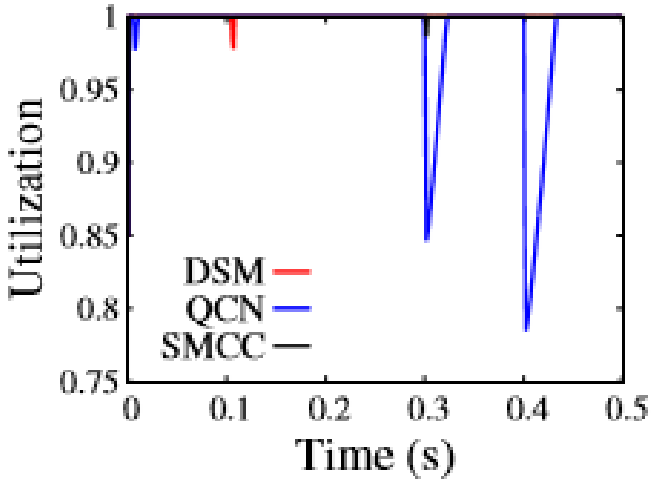
Properties

Responsiveness





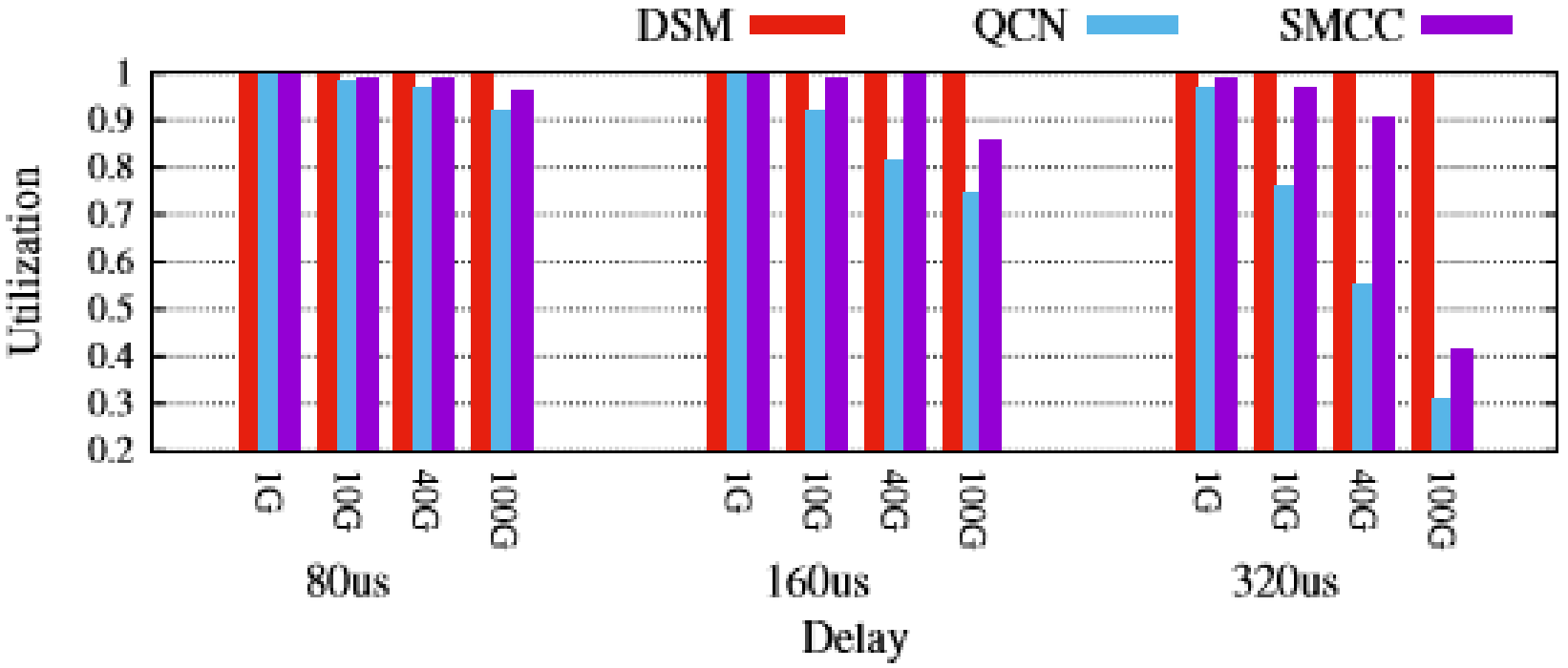
(a) Queue Length



(b) Utilization

Properties

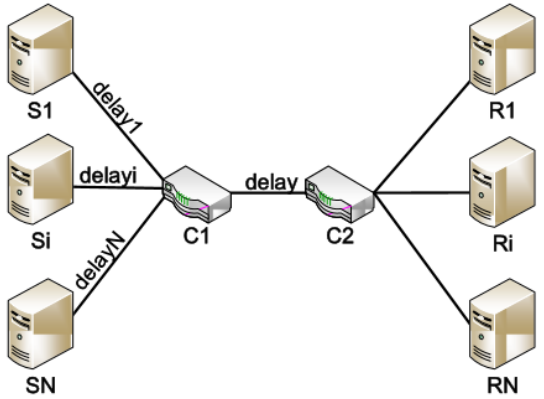
Adaptability

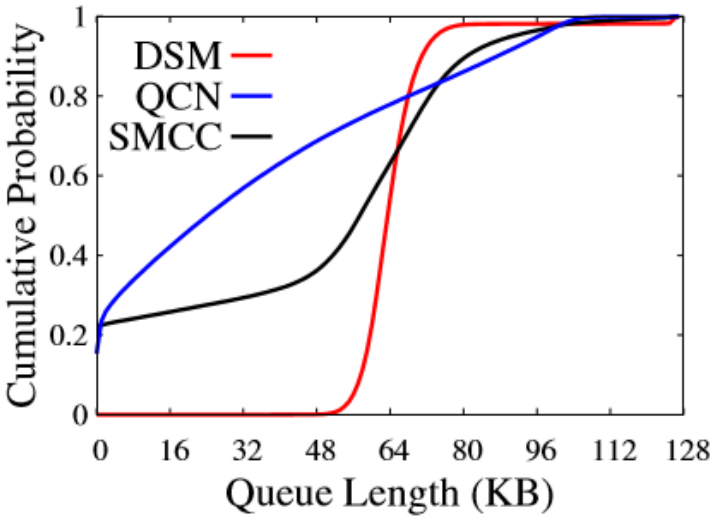


Evaluation

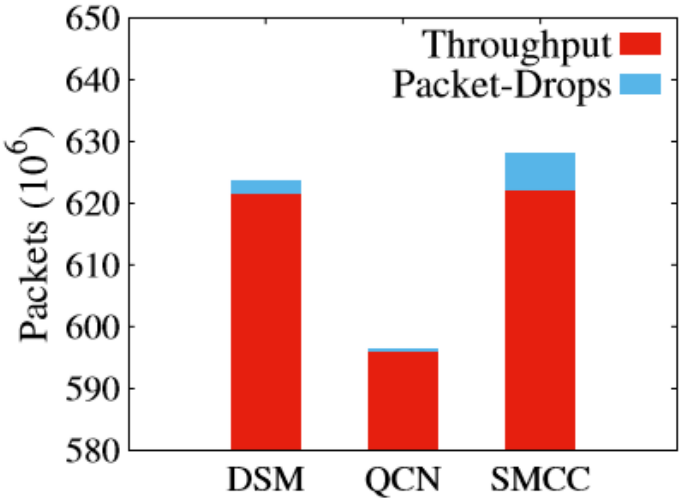
Heterogeneous and Time-Varying Delays

- Dumbbell topology
- 10Gbps links
- Random delays
- 100 times





(a) Queue Length

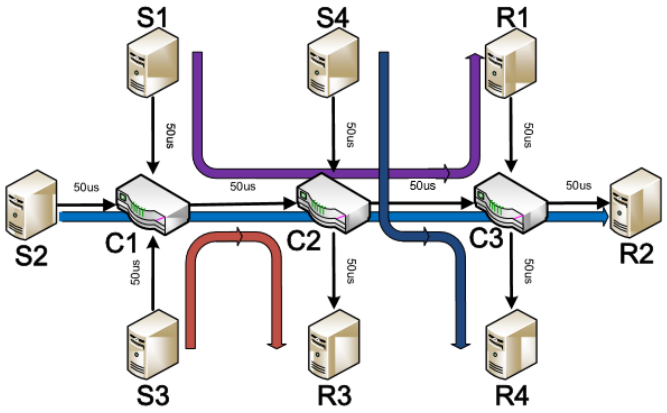


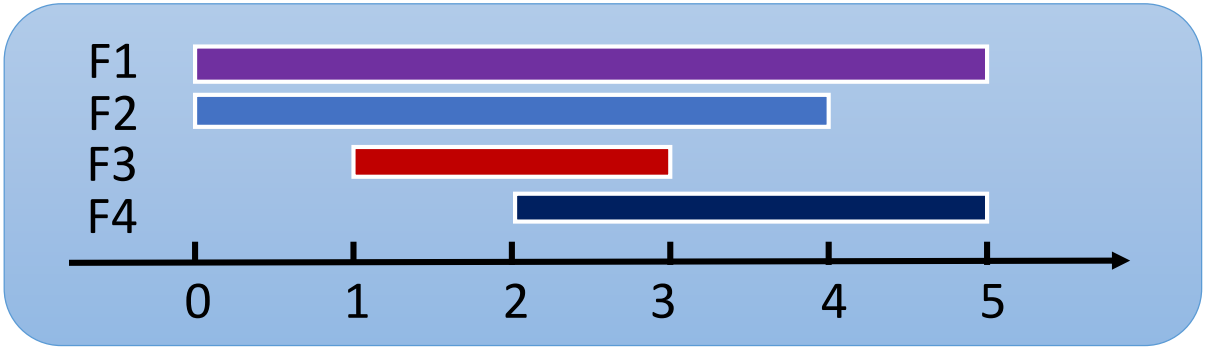
(b) Throughput and Packet-Drops

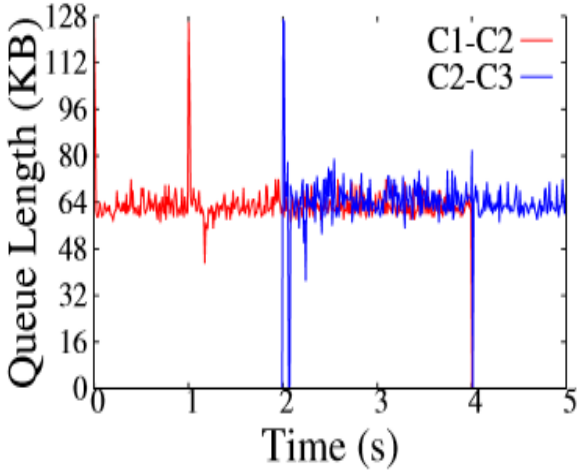
Evaluation

Multiple Bottlenecks Scenario

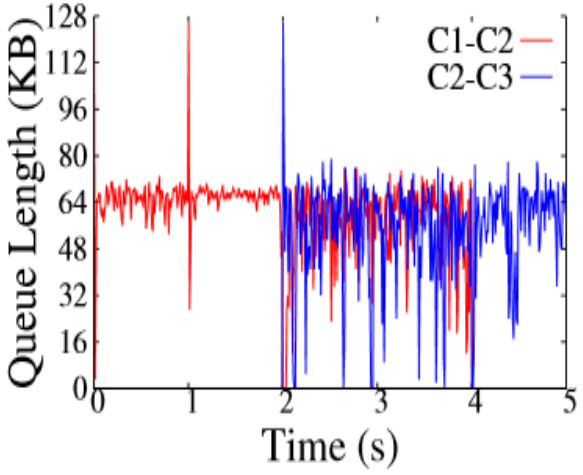
Long Flows



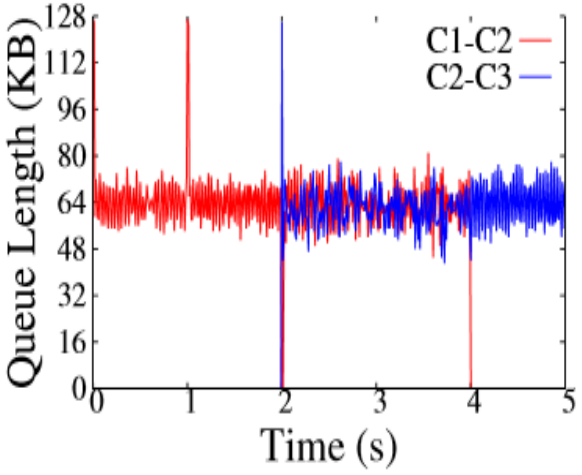




(a) DSM



(b) QCN



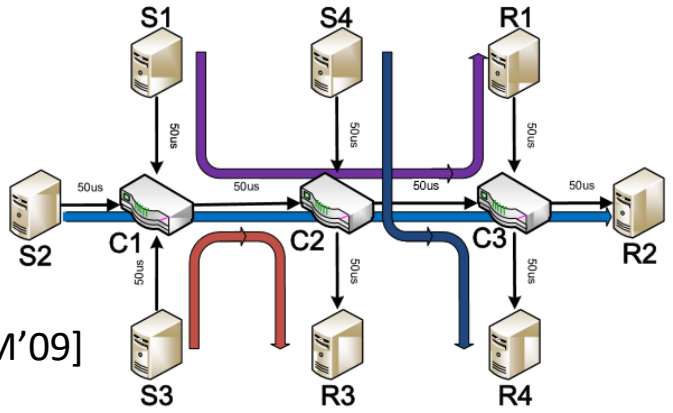
(c) SMCC

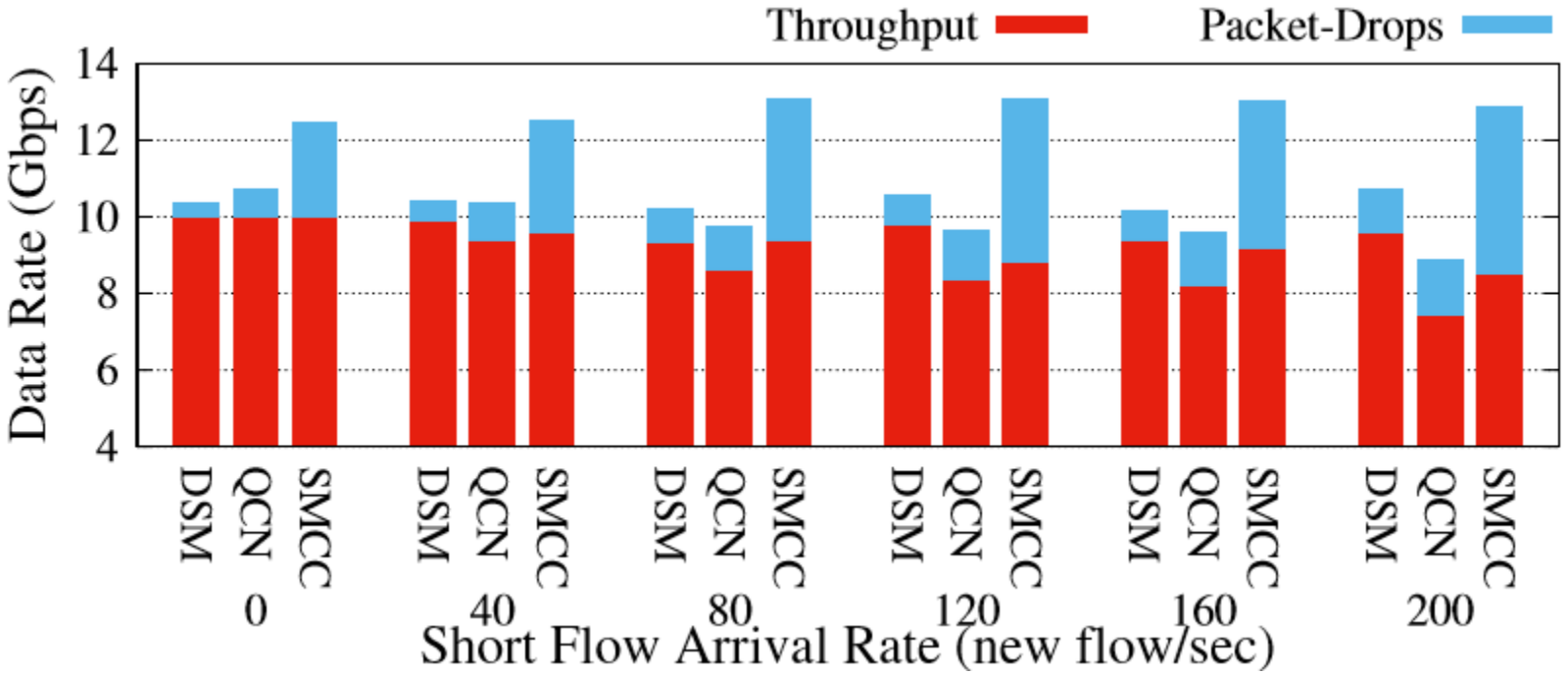
Evaluation

Multiple Bottlenecks Scenario

Mixed Flows

- 5 long-lived flows
- Short flows
  - Poisson arrival
  - Heavy-tailed flow size[SIGCOMM'09]





Conclusion

Mitigate the negative impacts of delays

- Model ➔ New congestion detector
  - Estimate the real congestion status with historical information.
  - Regard heterogeneous and time-varying feature as disturbances.

- Delay-tolerant Sliding Mode (DSM) scheme
  - New congestion detector ➔ Delay-tolerant
  - Sliding mode control method ➔ Robust to disturbances

- Properties of DSM
  - Complexity √
  - Responsiveness √
  - Stability √
  - Adaptability √
  - Heterogeneous and time-varying delays √
  - Multiple bottlenecks scenarios √

Thanks!

Q & A