Performance Analysis of Storage-Based Routing for Circuit-Switched Networks[1]

Presenter: Yongcheng (Jeremy) Li

PhD student, School of Electronic and Information Engineering, Soochow University, China

Email: liyongcheng621@163.com

Group Meeting, Friday, August 5, 2016

[1] C. Sun C, W. Guo, Z. Liu, *et al*, "Performance Analysis of Storage-Based Routing for Circuit-Switched Networks," *Journal of Optical Communications and Networking*, vol. 8, no. 5, pp. 282-289, May 2016.

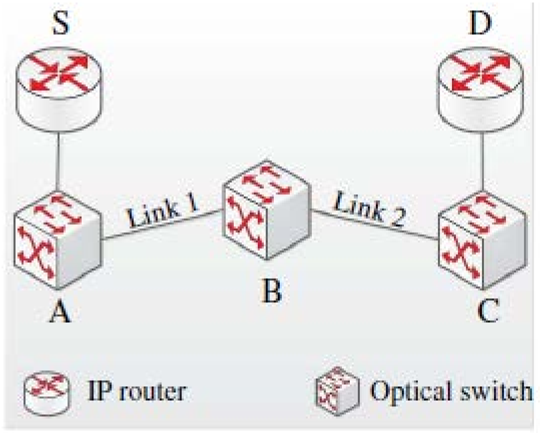
Outline

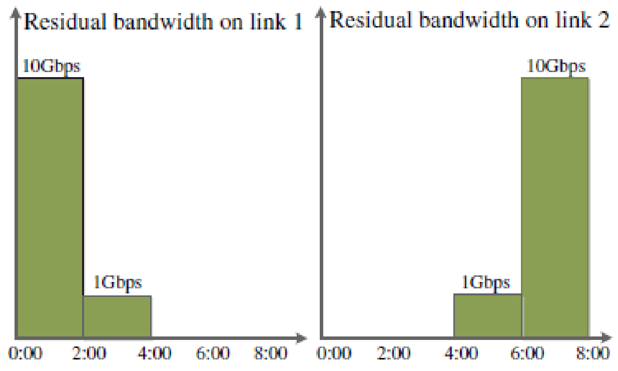
Background

- Circuit switching provides deterministic delay, guaranteed bandwidth, and low jitter, and it plays a significant role in many networking applications.
- Apart from real-time streaming applications, circuit switching has also been applied to support bulk data transfers which may not be as sensitive to delay as real-time applications.
- Traditional circuit switching uses an end-to-end provisioning mechanism, which reserves same amount of bandwidth on every link along a path.
- But available network resources may not be uniformly distributed.

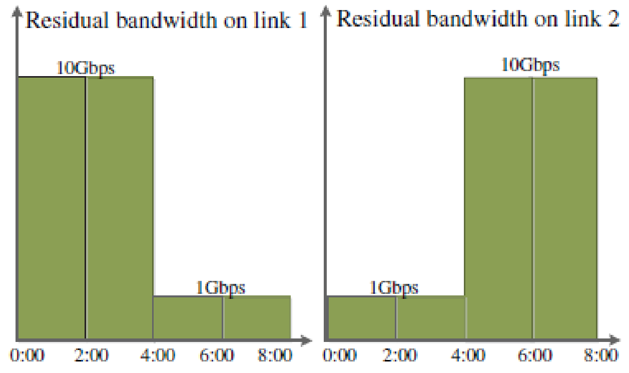
Background (contd.)

- Disadvantage
  - Different bandwidth usage and holding times of different connections may lead to dynamically varying residual bandwidths from link to link.
  - End-to-end circuit provisioning may not fully utilize available bandwidth on each link along the path.





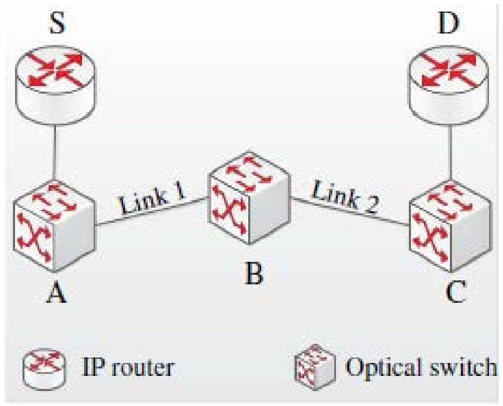
(b)



(c)

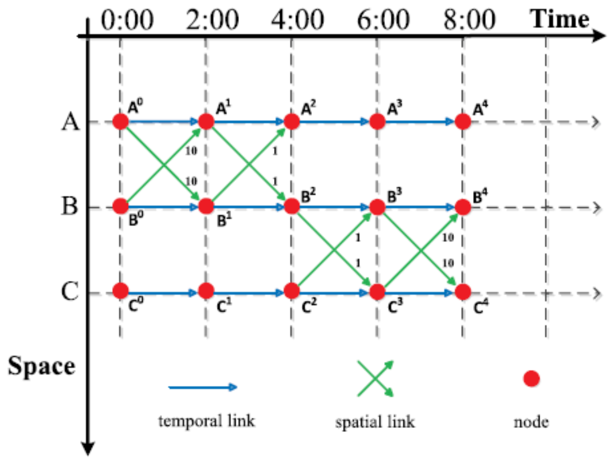
Background (contd.)

- Solution
  - Deploy storage capacity at intermediate nodes of a circuit-switched path to address non-uniform data rates on different links of the path.
  - Data could be stored electronically, exploiting long-term storage, large capacity, and random access at arbitrary times.
  - For an intermediate node, if incoming traffic rate is larger than outgoing rate, buffer local data by using storage and transfer part of data at a later time.



Time-Varying Graphs (TVGs)

- Presents node connection information (spatial information) of a graph during different time intervals (temporal information).
- Data is temporarily stored at some node when its *temporal links* are used.
- Directed links with weight value of bandwidth (unit: Gbps) are called *spatial links*.
- Spatial links reflect interconnections between different nodes.



Linear Program (LP) Model

- Notations
  - Topology: $G = (V, E)$, where $V = \{i | 1 \leq i \leq n\}$ represents n nodes and E represents link set.
  - $B = \{b_{ij} | 1 \leq i, j \leq n\}$ represents residual bandwidths of corresponding link.
  - $S = \{s_i | 1 \leq i \leq n\}$ is available storage for each node.
  - By dividing time into discrete time slots (of duration τ), get an extended directed graph $G' = (V', E')$, where $V' = \{v_i^k | 1 \leq i \leq n\}$, and E' is a extended link set under TVG model.
  - v_i^k represents node i at time $k\tau$.

LP model

- Notations
  - $B' = \{b_{ij}^k | 1 \le i, j \le n\}$ is a set of residual bandwidth, where b_{ij}^k represents residual bandwidth of link (i, j) during time slot $[k\tau, (k+1)\tau)$
  - $S' = \{s_i^k | 1 \le i \le n\}$ represents residual storage set, where s_i^k denotes residual storage of node i at time $k\tau$
  - Request: $r = (s, d, k_s, m, TTL)$ which arrives at source node s at time $k_s\tau$ and transfers a file with a pre-specified size m to destination d within a deadline TTL
  - Fixed path: $P = \{s, i_1, i_2 \ldots i_l, d\}$
  - Reserved bandwidth: $w_{i_p i_{p+1}}^k$
  - Reserved storage: $r_{i_p}^k$

LP model

Minimize total transmission time T_{total}

Objective:

$$\text{Min}(k'_{l+1} - k_s) \cdot \tau$$

Object to:

Ensure that every node completes entire file transfer

$$\begin{cases} \sum_{k=k_p}^{k'_p-1} w^k_{i_p i_{p+1}} \cdot \tau < m \ (0 \le p \le l), & (1) \\ \sum_{k=k_p}^{k'_p} w^k_{i_p i_{p+1}} \cdot \tau \ge m \ (0 \le p \le l), & (2) \end{cases}$$

Flow-conservation equation

$$w^k_{i_{p-1} i_p} \cdot \tau + r^k_{i_p} = w^k_{i_p i_{p+1}} \cdot \tau + r^{k+1}_{i_p} \ (1 \le p \le l, k_p \le k \le k'_p), \quad (3)$$

Constrains maximum bandwidth allocated on corresponding links

$$0 \le w^k_{i_p i_{p+1}} \le b^k_{i_p i_{p+1}} \ (1 \le p \le l, k_p \le k \le k'_p), \quad (4)$$

Constrains maximum storage allocated at corresponding nodes

$$0 \le r^k_{i_p} \le s^k_{i_p} \ (1 \le p \le l, k_p \le k \le k'_p), \quad (5)$$

Indicates that before and after data transmission, we do not allocate storage capacity for intermediate nodes

$$r^{k_p}_{i_p} = r^{k'_p+1}_{i_p} = 0 \ (1 \le p \le l), \quad (6)$$

Limit time range of data transmission

$$\begin{cases} k_s \le k_p \le k_{p+1} \ (0 \le p \le l), & (7) \\ k'_p \le k'_{p+1} \ (0 \le p \le l), & (8) \end{cases}$$

Ensure that data transmission can be accomplished within TTL

$$(k'_{l+1} - k_s) \cdot \tau \le \text{TTL}. \quad (9)$$

Store-Wait-Forward Algorithm

- Process
  - Step 1:
    - Generate ***a cost matrix*** based on reciprocal value of residual bandwidth on every link (Line 3).
    - If residual bandwidth is zero, assign a cost value of infinity.
  - Step 2:
    - Perform Dijkstra’s shortest-path algorithm on graph G to obtain a candidate path $P = \{s, i_1, i_2 ... i_l, d\}$; (Line 4) .

    (Shortest path is defined as one with minimum cost sum of all links along a path).
  - Step 3:
    - Apply ***Forward Reservation Backward Feedback*** (FRBF) scheme to reserve bandwidth and storage resources on extended graph G' (Line 5).
    - If FRBF fails to allocate network resources, delay the request for a time slot and re-run SWF until a valid path is found or the request is considered to be blocked.

UCDAVIS
UNIVERSITY OF CALIFORNIA

Forward Reservation Backward Feedback

- Maximize bandwidth utilization of each link in an end-to-end path.
- Two phases: *forward reservation phase* and backward feedback phase.
- Forward reservation phase
  - Make bandwidth and storage reservation link by link and hop by hop.
  - For source node, fully utilize the residual bandwidth.
  - Reserved bandwidth value $w_{i_0 i_1}^{k}$ and reserved storage value r_{i0}^{k+1} are shown as $w_{i_0 i_1}^{k} = b_{i_0 i_1}^{k}, r_{i0}^{k+1} = 0$.
  - Eq. (11) defines feedback to assist us with network resource reservation.
- Backward feedback phase
  - Reduce upstream data transmission rate.

Forward Reservation Backward Feedback

$$\text{feedback} = (b^k_{i_p i_{p+1}} \cdot \tau + s^{k+1}_{i_p}) - (w^k_{i_{p-1} i_p} \cdot \tau + r^k_{i_p}), \qquad (11)$$

$$\begin{cases} w^k_{i_p i_{p+1}} = w^k_{i_{p-1} i_p} + r^k_{i_p}/\tau, & \boxed{\text{feedback} \geq s^{k+1}_{i_p}} \\ w^k_{i_p i_{p+1}} = b^k_{i_p i_{p+1}}, & \boxed{0 \leq \text{feedback} < s^{k+1}_{i_p}} \\ w^k_{i_p i_{p+1}} = b^k_{i_p i_{p+1}}, & \boxed{\text{feedback} < 0} \end{cases} \qquad (12)$$

$$\begin{cases} r^{k+1}_{i_p} = 0, & \text{feedback} \geq s^{k+1}_{i_p} \\ r^{k+1}_{i_p} = s^{k+1}_{i_p} - \text{feedback}, & 0 \leq \text{feedback} < s^{k+1}_{i_p} \\ r^{k+1}_{i_p} = s^{k+1}_{i_p}, & \text{feedback} < 0 \end{cases}. \qquad (13)$$

Transmission capacity of outgoing link is capable of transferring all the received data and storage is not needed

Make use of full transmission capacity of outgoing link, and storage should also be applied to buffer exceeded data

Data transmission will overflow even if we transfer data at full speed and apply entire residual storage at node i_p

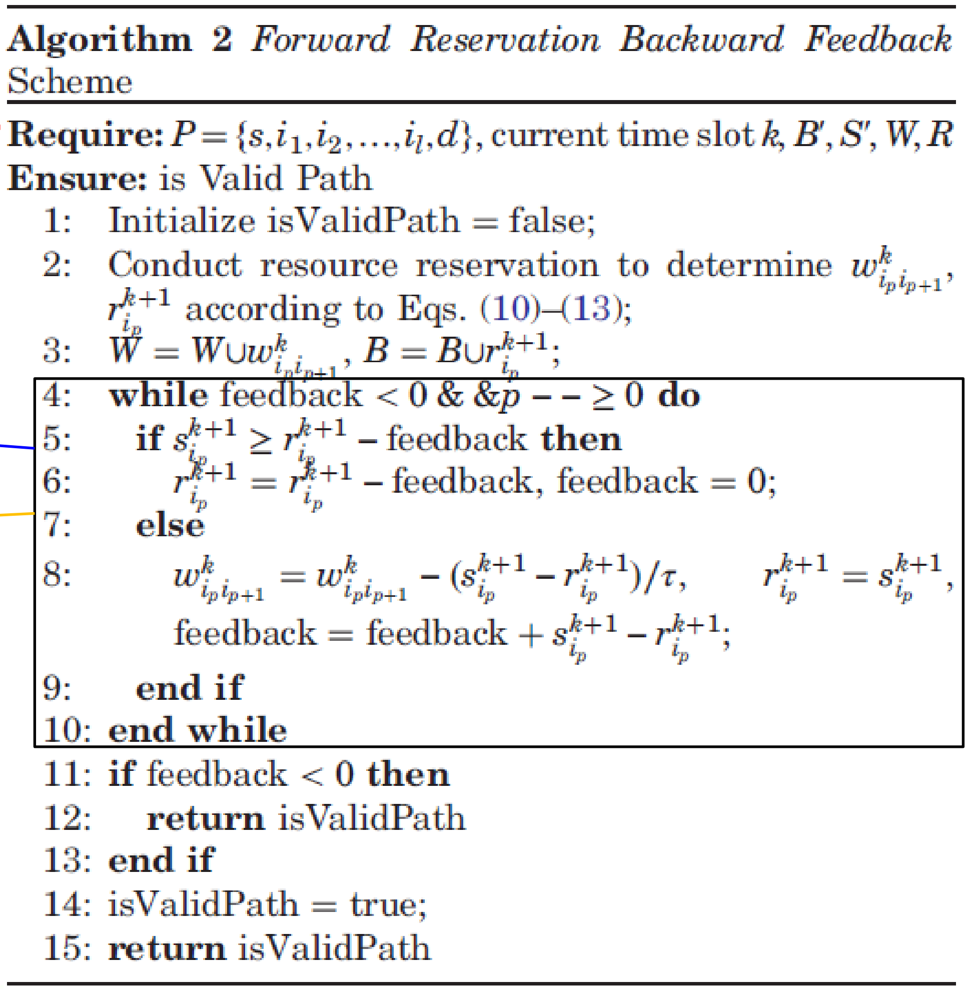
Forward Reservation Backward Feedback

- FRBF

Reduce incoming link's transmission rate and buffer exceeded data at previous node until feedback value becomes zero.

Exceeded data could be stored at previous node.

Reduce upstream data transmission rate and check upstream nodes's storage capacity to avoid data overflow.



Algorithm 2 *Forward Reservation Backward Feedback* Scheme

```
Require: $P = \{s, i_1, i_2, \ldots, i_l, d\}$, current time slot $k$, $B'$, $S'$, $W$, $R$
Ensure: is Valid Path
 1: Initialize isValidPath = false;
 2: Conduct resource reservation to determine $w^k_{i_p i_{p+1}}$, $r^{k+1}_{i_p}$ according to Eqs. (10)–(13);
 3: $W = W \cup w^k_{i_p i_{p+1}}$, $B = B \cup r^{k+1}_{i_p}$;
 4: while feedback $< 0$ & &$p--\geq 0$ do
 5:   if $s^{k+1}_{i_p} \geq r^{k+1}_{i_p}$ − feedback then
 6:     $r^{k+1}_{i_p} = r^{k+1}_{i_p}$ − feedback, feedback = 0;
 7:   else
 8:     $w^k_{i_p i_{p+1}} = w^k_{i_p i_{p+1}} - (s^{k+1}_{i_p} - r^{k+1}_{i_p})/\tau$,   $r^{k+1}_{i_p} = s^{k+1}_{i_p}$,
        feedback = feedback + $s^{k+1}_{i_p} - r^{k+1}_{i_p}$;
 9:   end if
10: end while
11: if feedback $< 0$ then
12:   return isValidPath
13: end if
14: isValidPath = true;
15: return isValidPath
```

Performance Evaluation

- Simulation conditions
  - Network topology: NSF
  - Approaches: LP, SWF, and VBR[1]
  - One wavelength per link
  - Maximum bandwidth capacity: 10 Gbps
  - Residual bandwidth during each time slot is independently and uniformly distributed in [6, 10] Gbps
  - Source and destination nodes are randomly generated

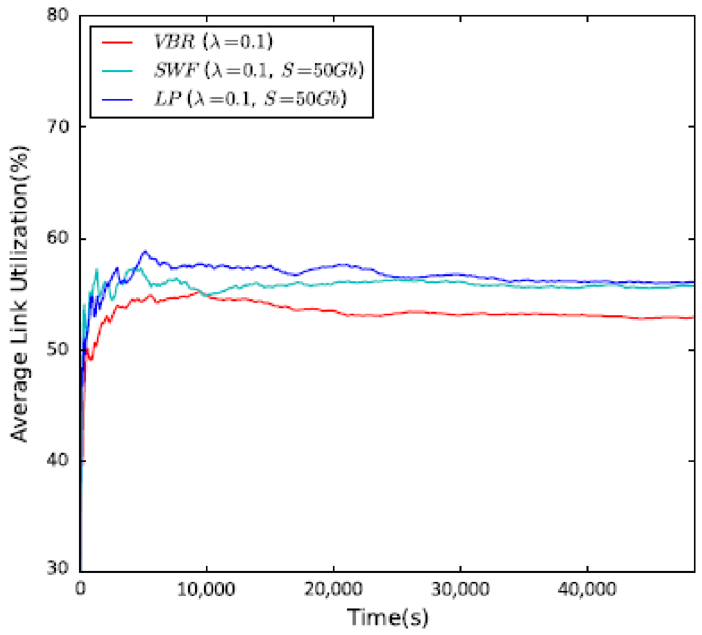
[1] VBR applies same routing and resource allocation strategy as SWF except that storage capacity of each node is 0

Performance Evaluation (contd.)

- File size ***m*** is uniformly distributed in [200, 800] Gb
- Time deadline TTL : 400 s
- Time granularity: 1 s
- Requests arrive following a Poisson process with arrival rate λ (requests/s)
- *10,000* requests were performed for each run
- *10* runs were conducted to achieve mean value as data points in figures

UCDAVIS
UNIVERSITY OF CALIFORNIA

Average link utilization

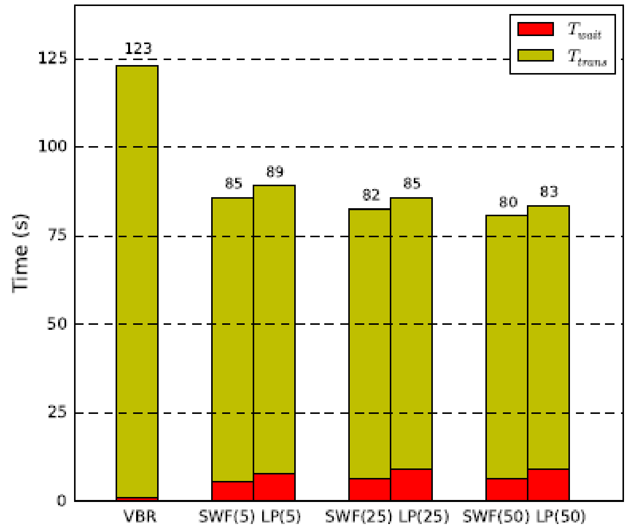


- Compares average link utilization of LP, SWF, and VBR under one run of simulated experiment.
- Both LP and SWF achieve up to 56% link utilization vs. VBR, which achieves only 52% utilization.
- Performances of LP and SWF are quite close, which indicates that SWF scheme is efficient.

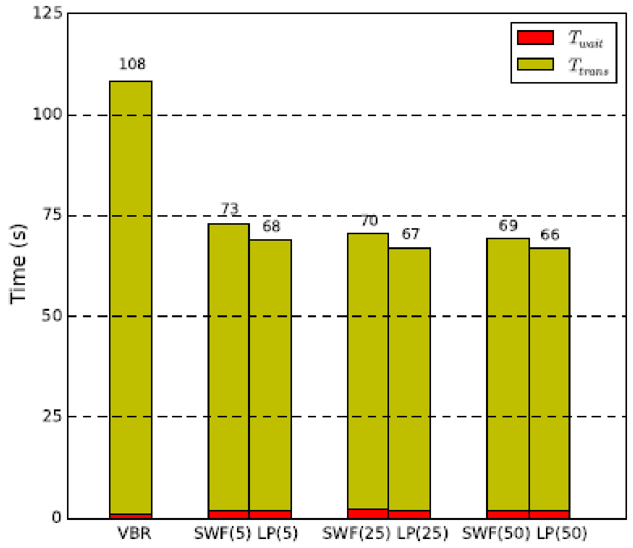
UCDAVIS UNIVERSITY OF CALIFORNIA

Total transmission time and wait time

λ is set as 0.05



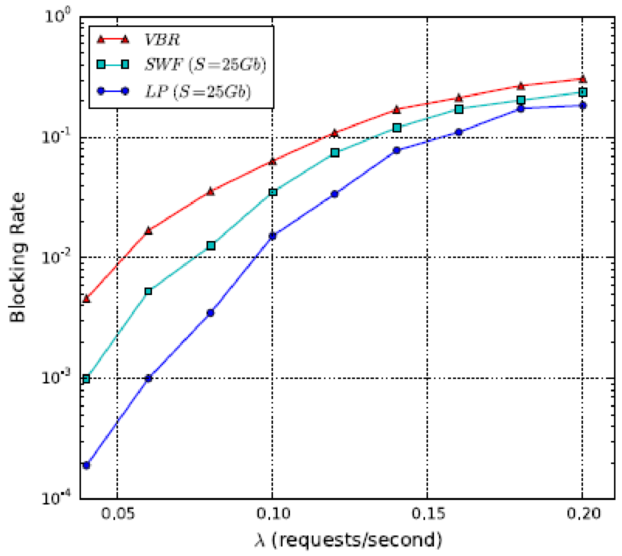
λ is set as 0.1

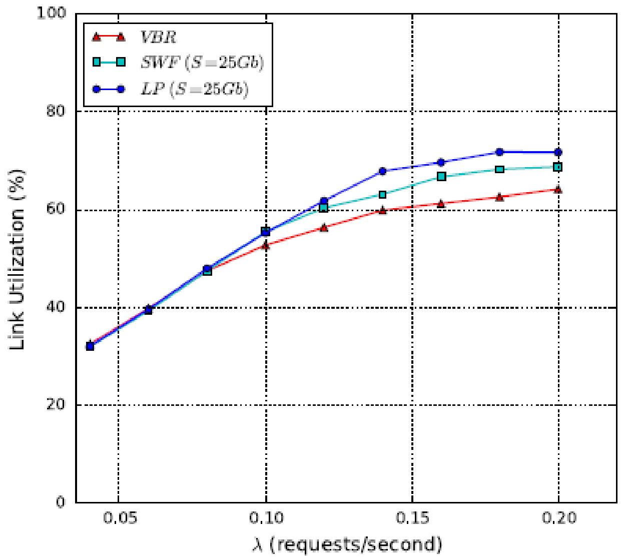


- **Storage can efficiently reduce data transmission time**
- **T_{wait} of VBR is much smaller than LP and SWF because bandwidth is fragmented for VBR and requests can be served quickly, although sometimes at a low transmission rate**
- **T_{total} decreases when storage size increases, while T_{wait} experiences an increase since more data may be buffered at an intermediate node**

UC DAVIS
UNIVERSITY OF CALIFORNIA

Blocking rate and link utilization

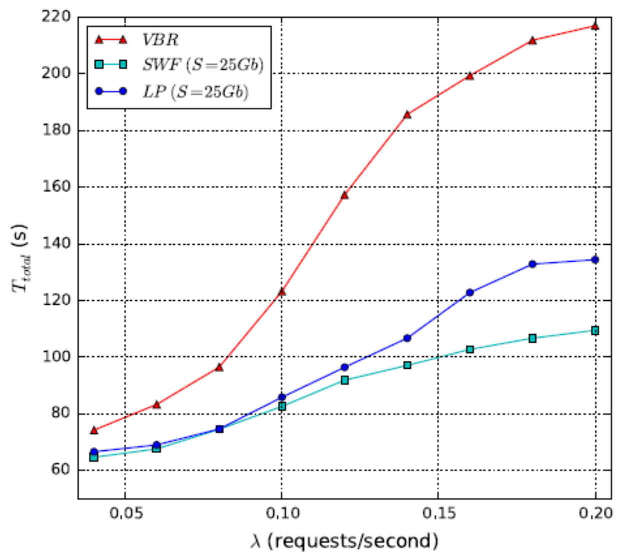


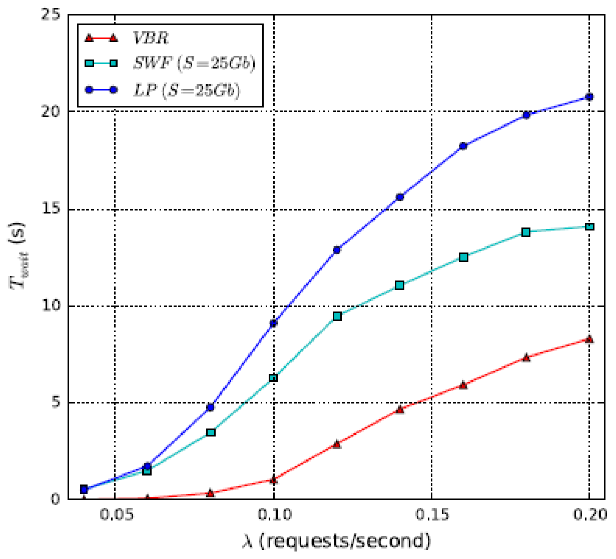


- **LP and SWF outperform in terms of blocking rate and link utilization.**
- **LP and SWF can make use of a small bandwidth to transfer data, while VBR cannot establish an end-to-end path due to bandwidth shortage.**

Performance evaluation

- Data transmission time and wait time





- **Introduction of storage reduces data transmission time significantly.**
- **As a trade-off, storage-based schemes introduce a longer delay since data transmission may experience a wait time at switching nodes.**

Conclusion

- Proposed to deploy storage capacity at intermediate nodes for circuit switching to address the bottleneck link along a path.
- Proposed a storage based routing and resource allocation algorithm (SWF).
- Simulation results demonstrate that SWF consumes less data transfer time but introduces some delay as a trade-off.

Thank you for your attention!

Presenter: Yongcheng (Jeremy) Li

PhD student, School of Electronic and Information Engineering,

Soochow University, China

Email: liyongcheng621@163.com

Group Meeting, Friday, August 5, 2016