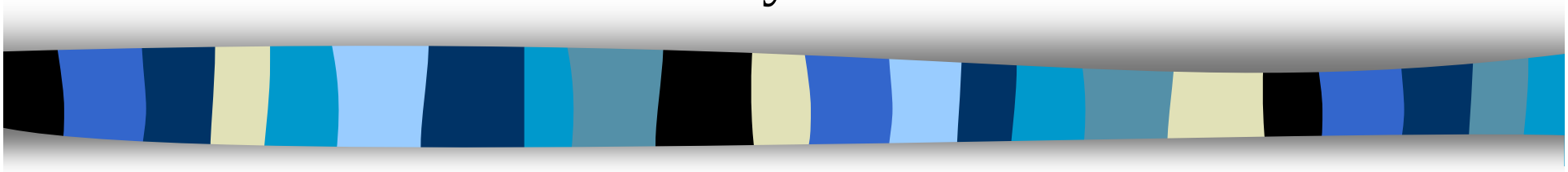
Pipelining Multicast Scheduling in All-Optical Packet Switches with Delay Guarantee



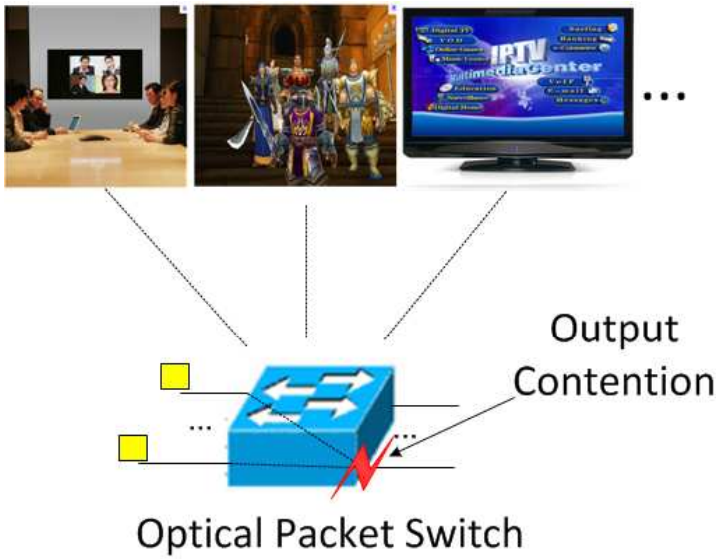
Zhiyang Guo and Yuanyuan Yang
New York State University
at Stony Brook

Outline

- Introduction
- The Switch Architecture
- Delay Guaranteed Multicast Scheduling
- Pipelining The Scheduling
- Performance Study
- Conclusion

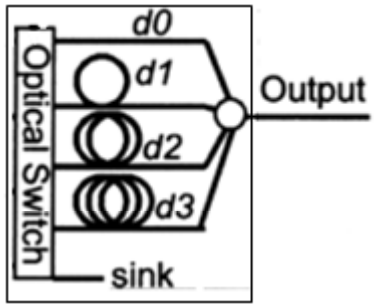
Introduction

- Emerging delay-sensitive multicast applications
  - IPTV, video conference, online gaming, etc.
- Output contention is a major challenge
  - Buffer-less approaches (e.g. deflection)
  - Use electronic buffer
  - Use fiber delay line (FDL)



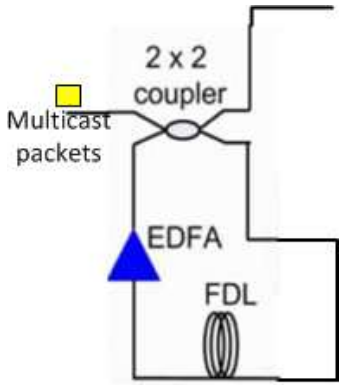
Related Work

- Existing Multicast Scheduling With FDL Buffer
  - *Output Buffering.*



4×1 output buffer

  - *Wavelength Assisted Routing*.

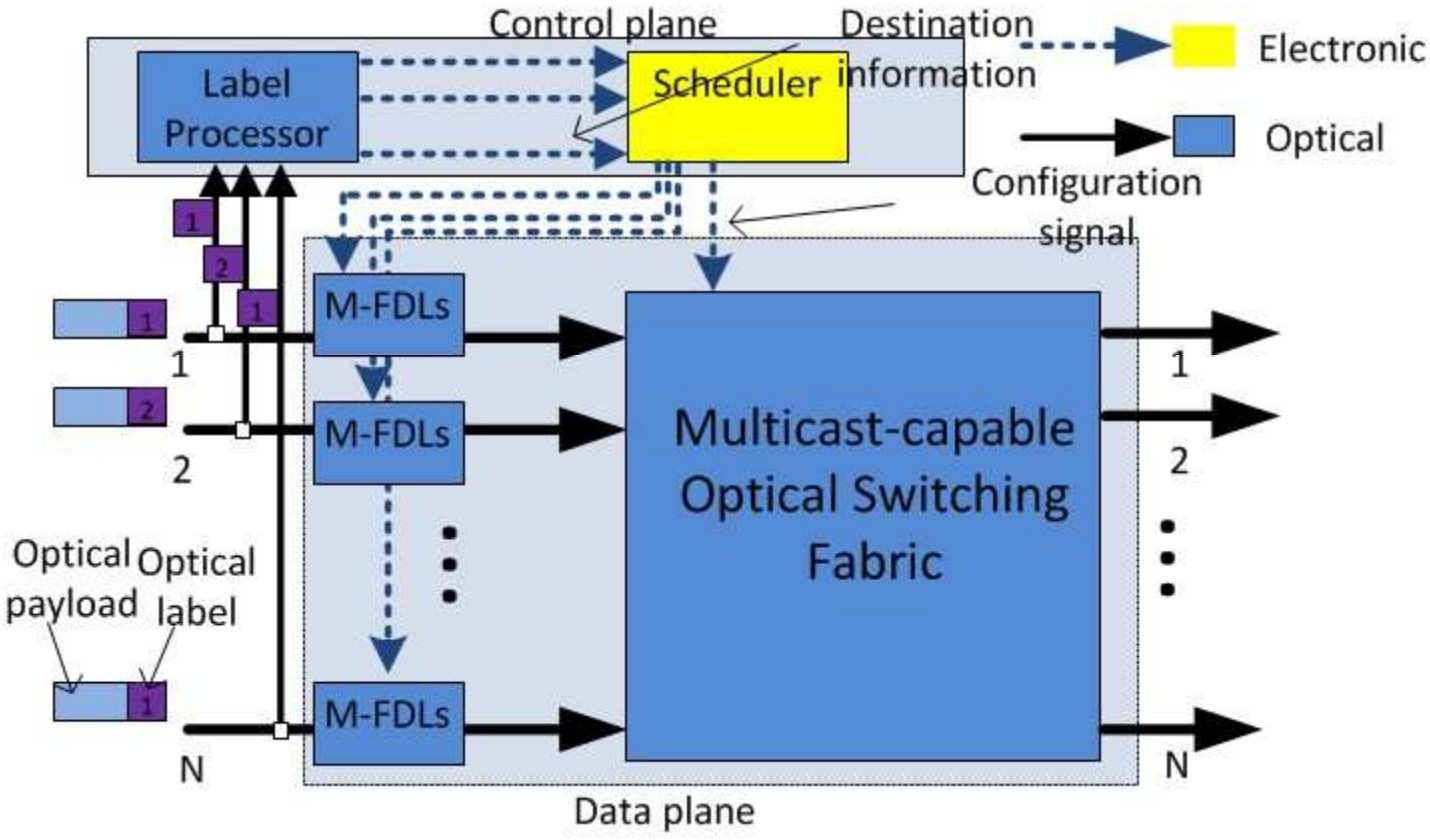


FDL loop buffer

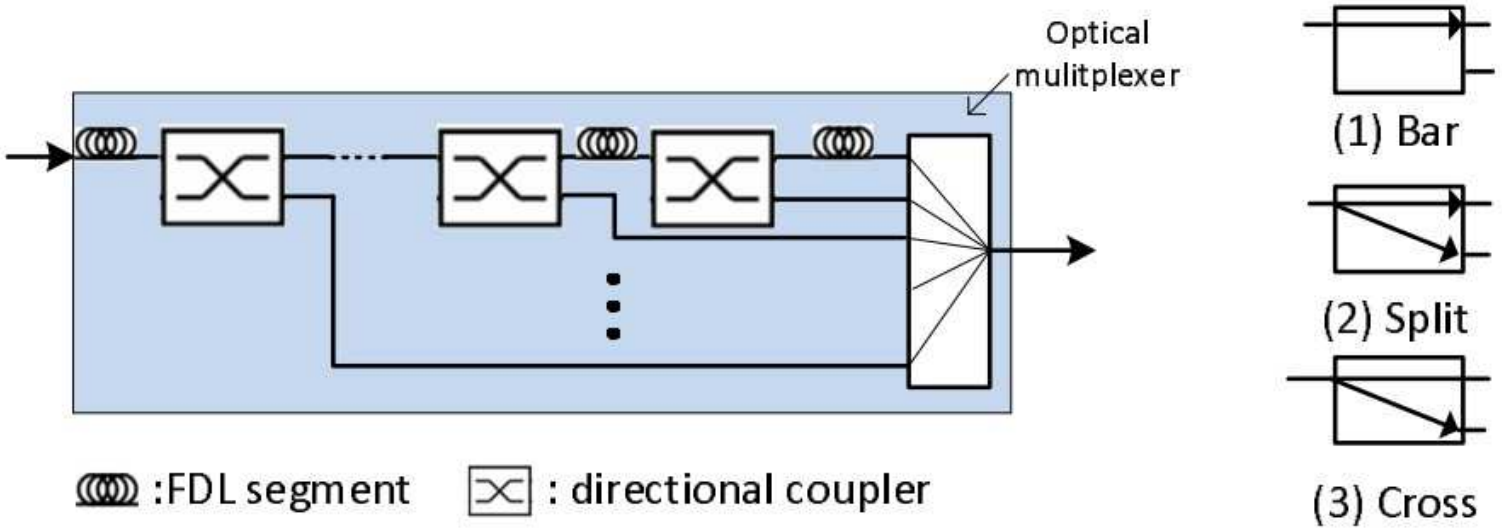
Our Contribution

- An efficient optical buffer called multicast-enabled FDLs (M-FDLs)
- A novel Delay-Guaranteed Multicast Scheduling (DGMS) algorithm
- Pipelining technique and hardware implementation.

The Switch Architecture

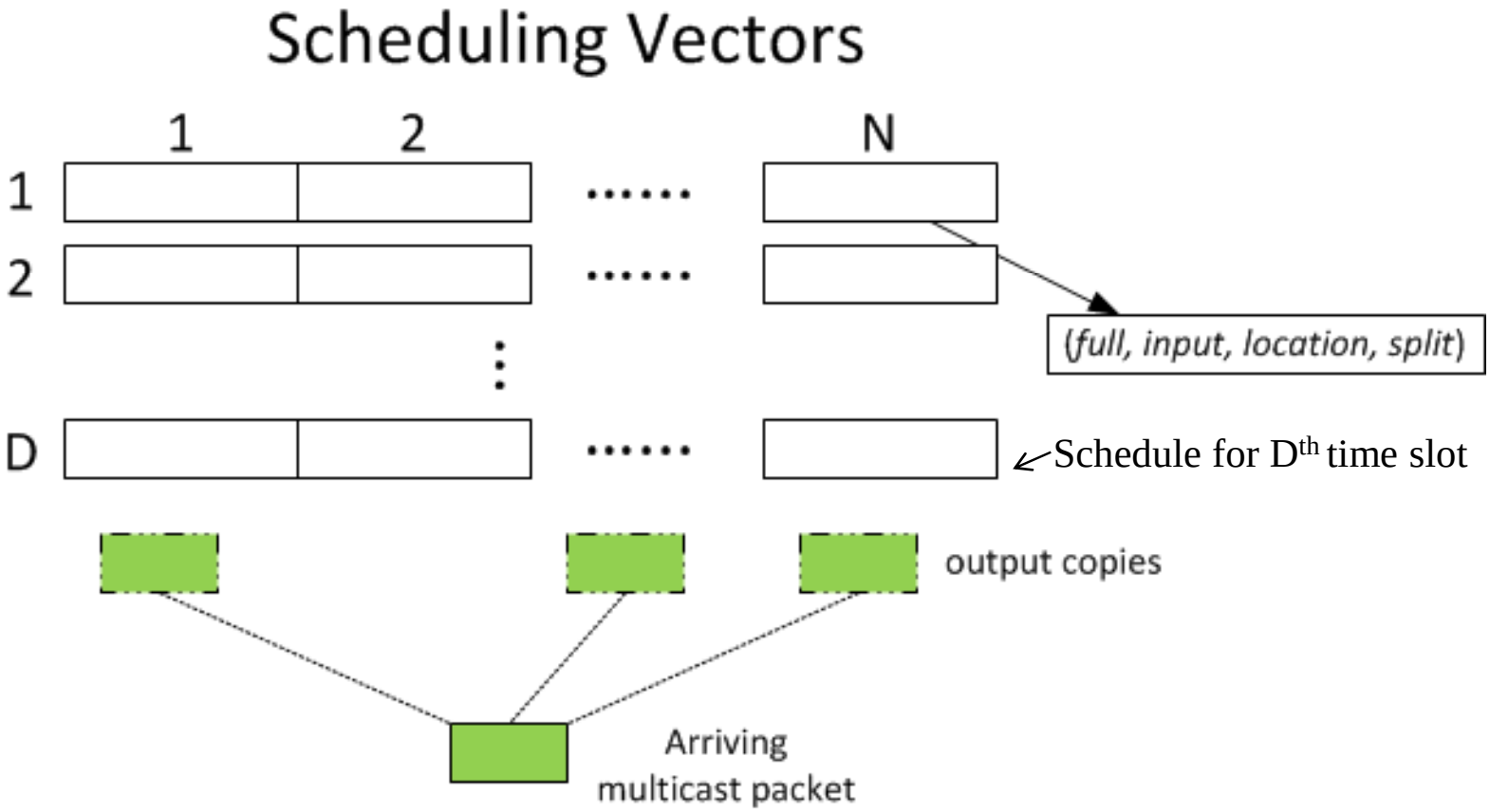


The Structure of M-FDL



(a) The structure of M-FDL (b) 3 states of the coupler

Delay Guaranteed Multicast Scheduling (DGMS)

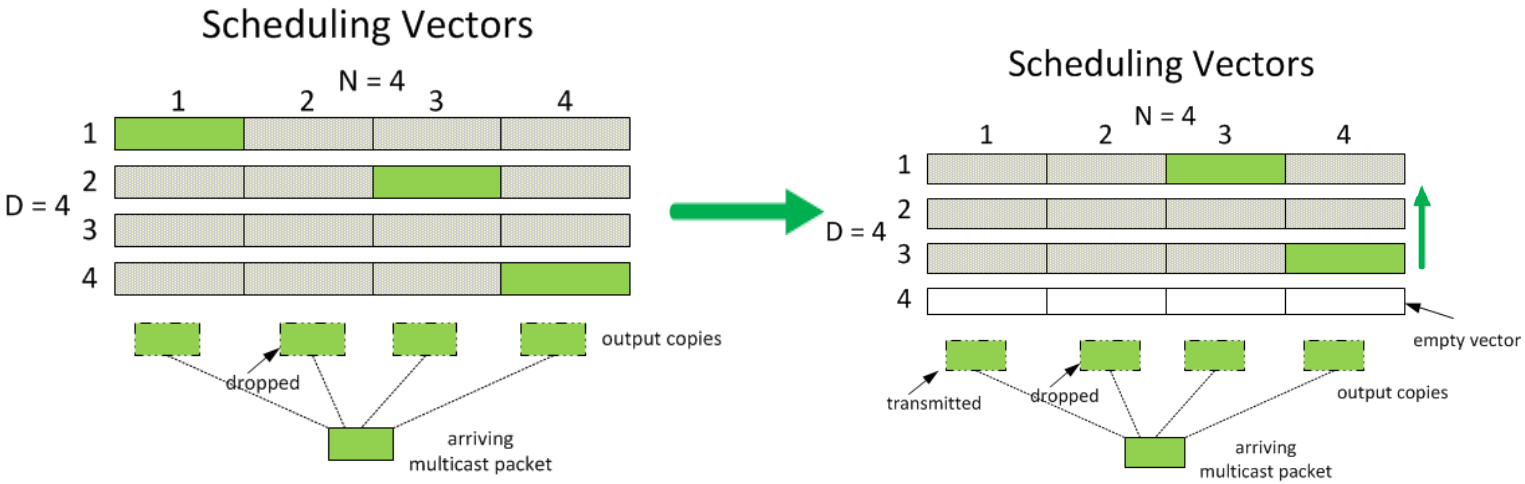


Delay Guaranteed Multicast Scheduling

- For a multicast packet, find the **earliest** possible **eligible** entry for each output copy
  - The entry is not full.
  - No packet from the same input has been previously assigned to this scheduling vector.

Delay Guaranteed Multicast Scheduling

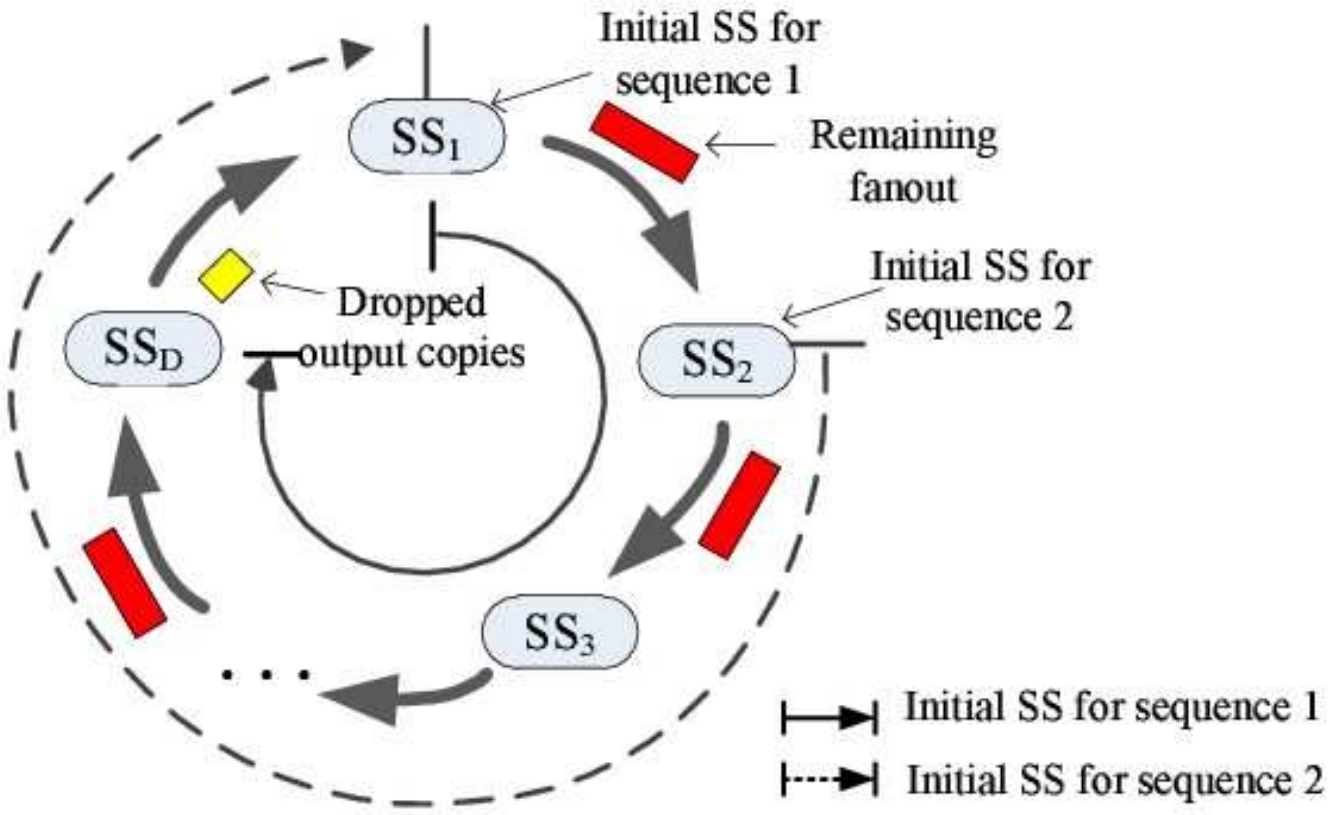
- Configuring switch
  - Send configuration signal according to the 1st scheduling vector
  - Update Scheduling vectors



Pipelining The Scheduling

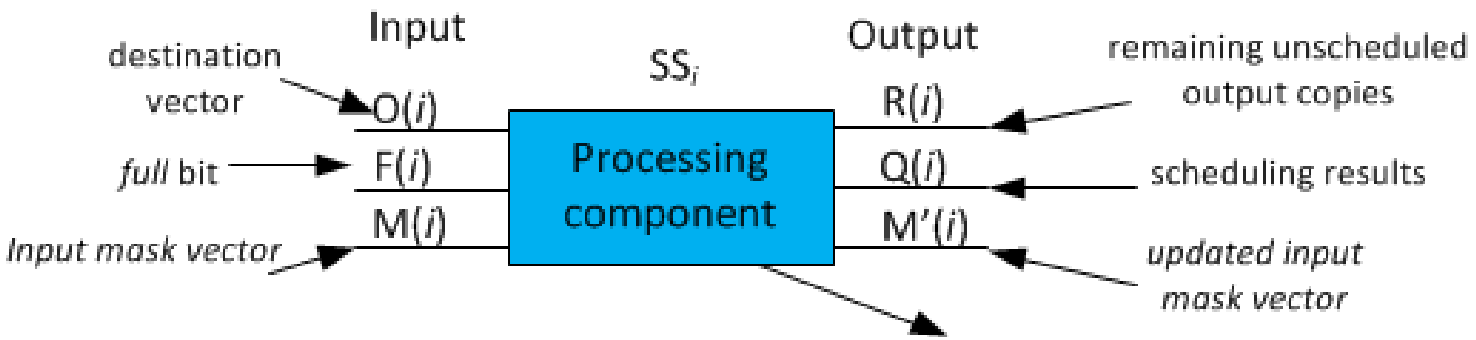
- DGMS can be pipelined to reduce time complexity
  - Time complexity of DGMS is $O(N^2D)$
  - Scheduling can be pipelined to reduce time complexity to $O(N)$.

Pipelining The Scheduling



Shifting the initial SS is equivalent to updating the scheduling vector

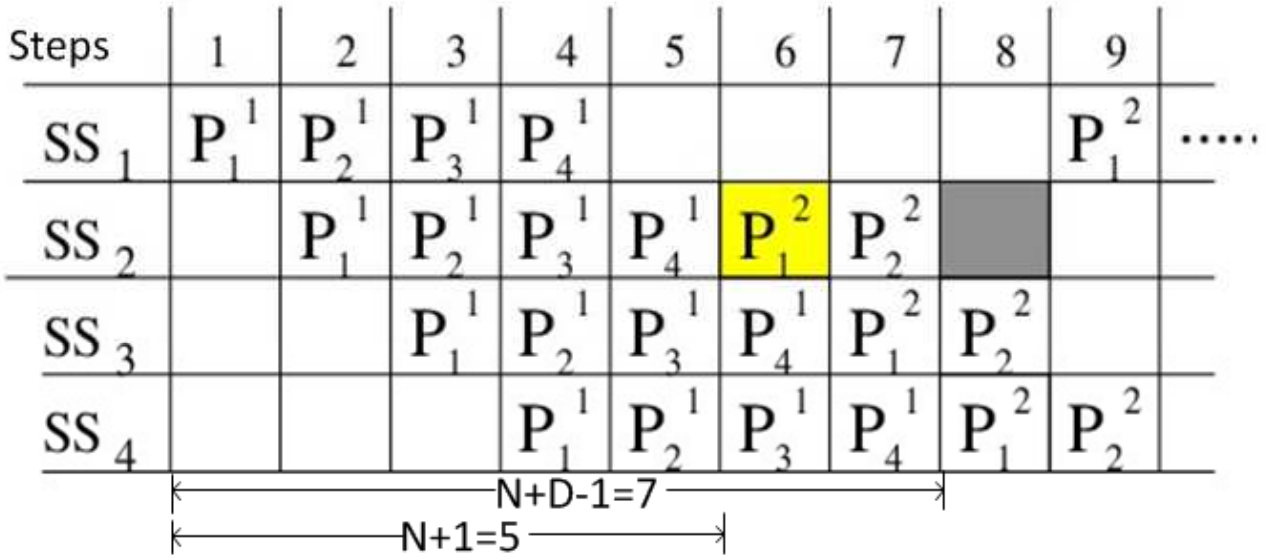
Combinational Circuit Design



$$
\begin{aligned}
R(i) &= O(i) \cap (M(i) \cup F(i)) \quad \forall i = 1, 2, \ldots, N \\
Q(i) &= \overline{M(i)} \cap \overline{F(i)} \cap O_i \quad \forall i = 1, 2, \ldots, N \\
M'(i) &= M(i) \cup Q(1) \cup Q(2) \cup \cdots \cup Q(N)
\end{aligned}
$$

Each sub-scheduler can complete the scheduling in $O(1)$ time with $O(N)$ hardware cost.

Pipelining the Scheduling



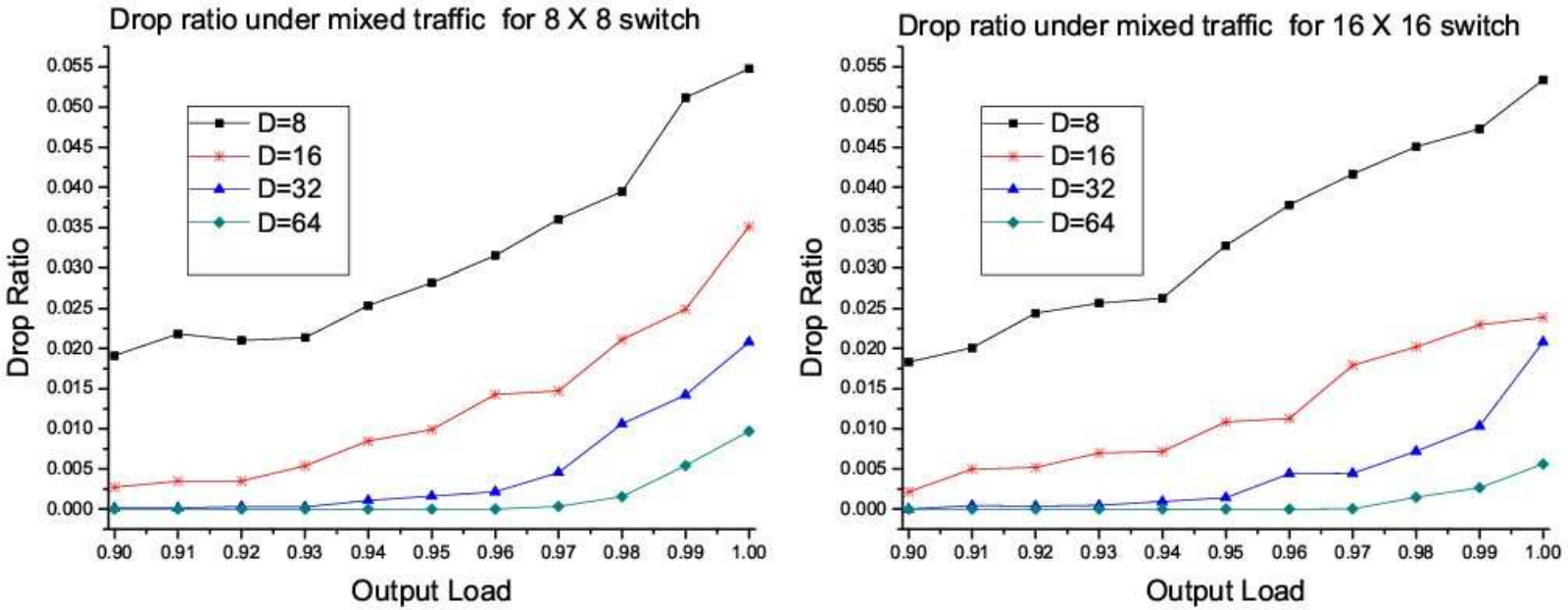
P_j^k: The packet arriving from input j in the k_{th} time slot

Pipelining consecutive time slots to further reduce time complexity

Performance Evaluations -Simulation Environment

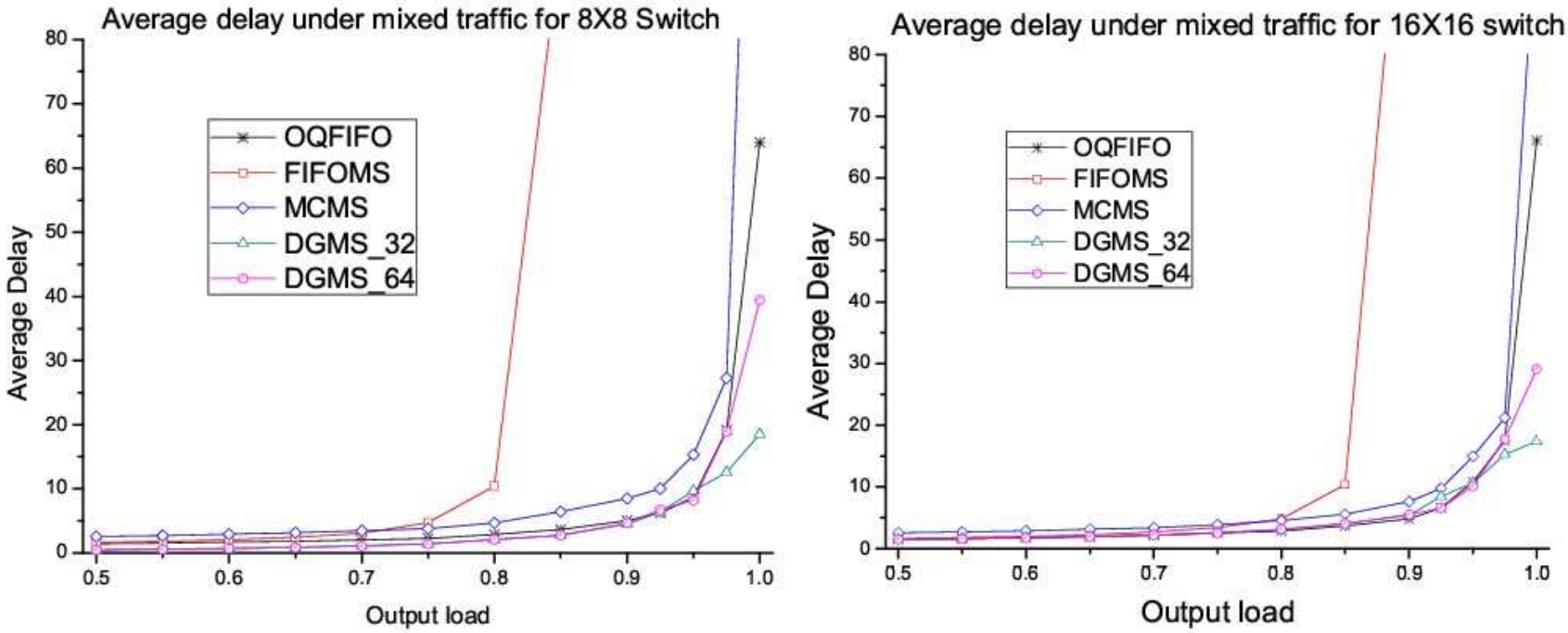
- Traffic patterns
  - Mixed traffic. (half multicast & half unicast)
  - Internet traffic trace.
- Comparison with previous algorithms
  - FIFOMS.
  - MCMS.
  - Ideal OQFIFO.

Performance Evaluations



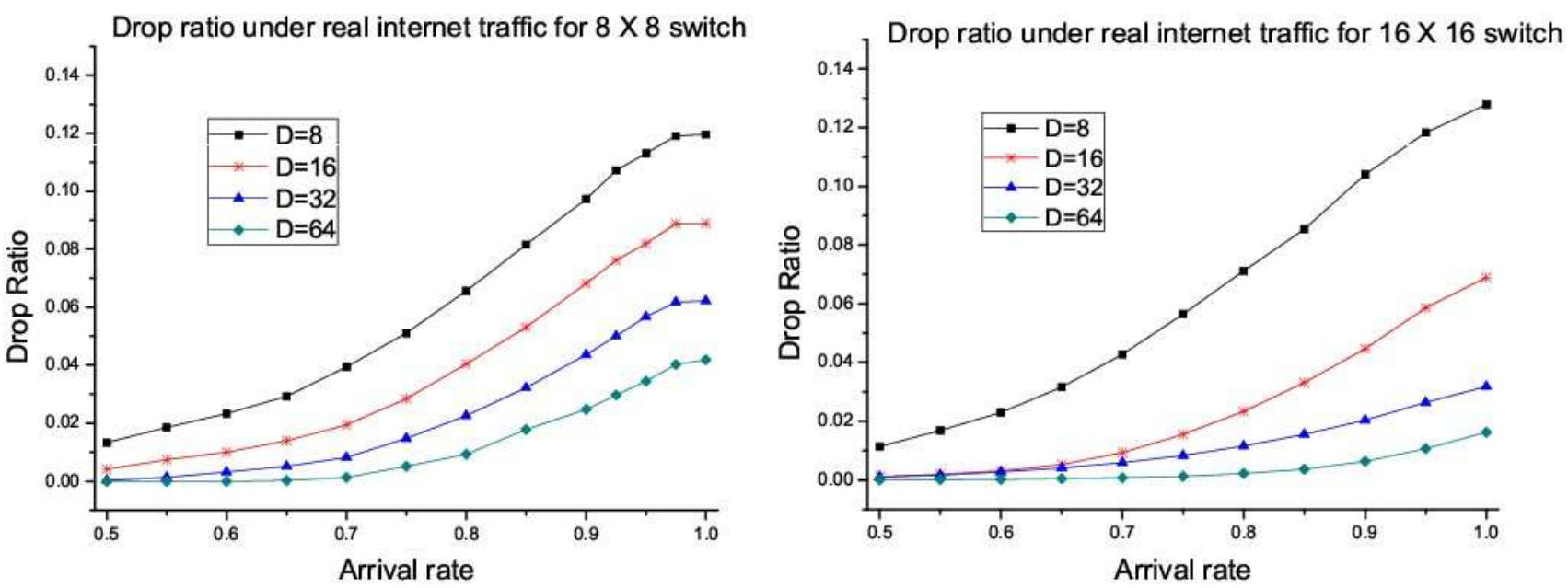
Packet drop ratio vs. *D* under mixed traffic. (a) 8 × 8 switch; (b) 16 × 16 switch.

Performance Evaluations



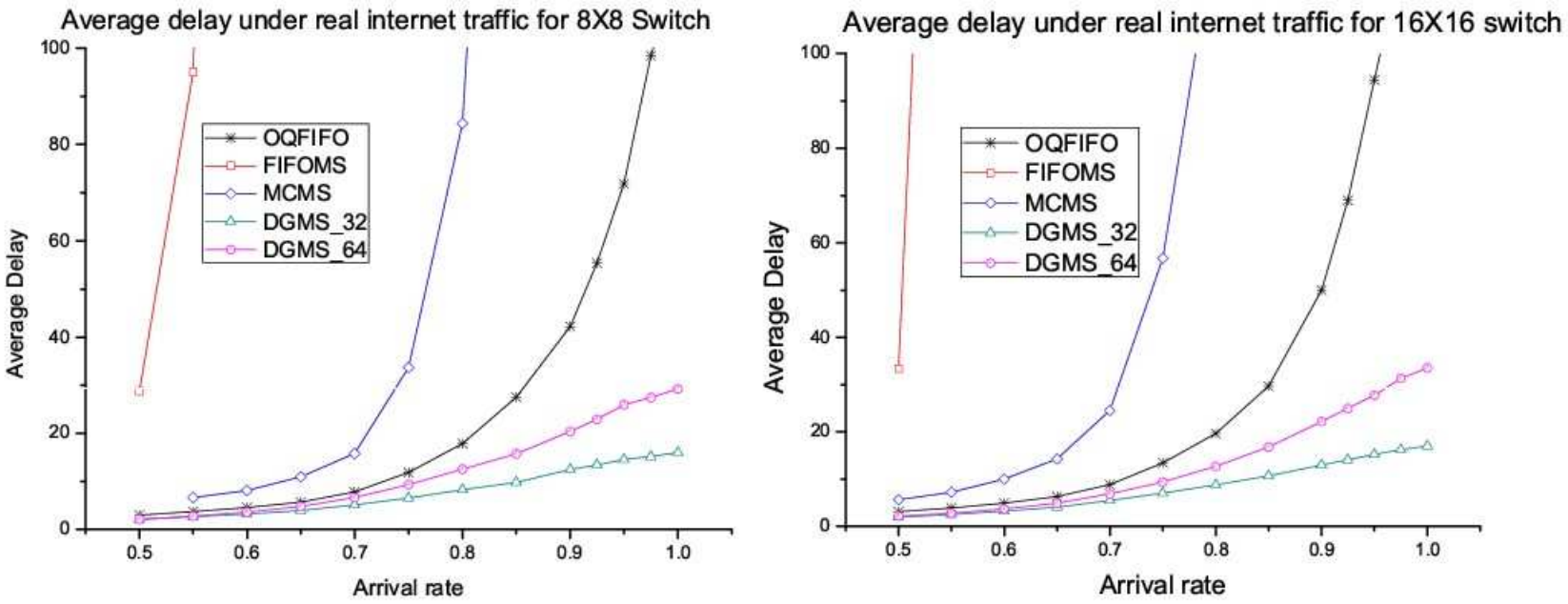
Average delay vs. output load under mixed traffic. (a) 8 × 8 switch; (b) 16 × 16 switch.

Performance Evaluations



Packet drop ratio vs. *D* under Internet traffic. (a) 8 × 8 switch; (b) 16 × 16 switch.

Performance Evaluations



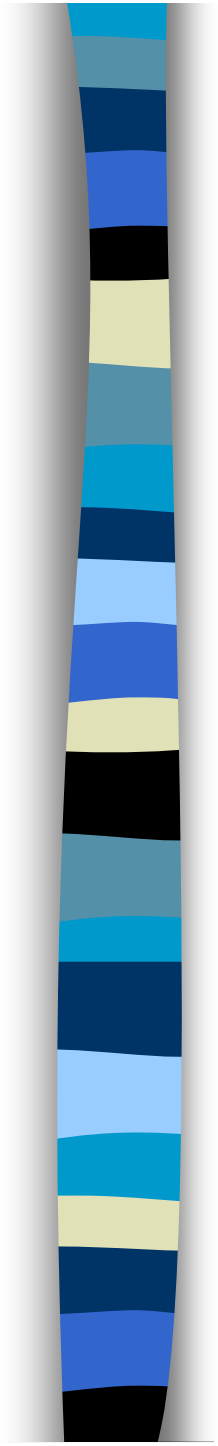
The average delay vs. arrival rate under Internet traffic. (a) 8 × 8 switch; (b) 16 × 16 switch.

Observation

- Bounded maximum delay for all transmitted packets
- Minimum packet drop ratio
- Ultra-low average packet delay
- Adaptive to traffic condition

Conclusion

- An efficient flexible FDL buffer called M-FDLs.
- A Delay-Guaranteed Multicast Scheduling (DGMS) algorithm, which achieves ultra-low average packet delay while keeps packet drops at a minimum level.
  - Guarantees bounded maximum delay for all transmitted packets.
  - Adaptive to varying transmission requirements.
- Pipelining technique and hardware implementation.



Thank You!