TENCON 2001
IEEE Region 10 International Conference on Electrical and Electronic Technology

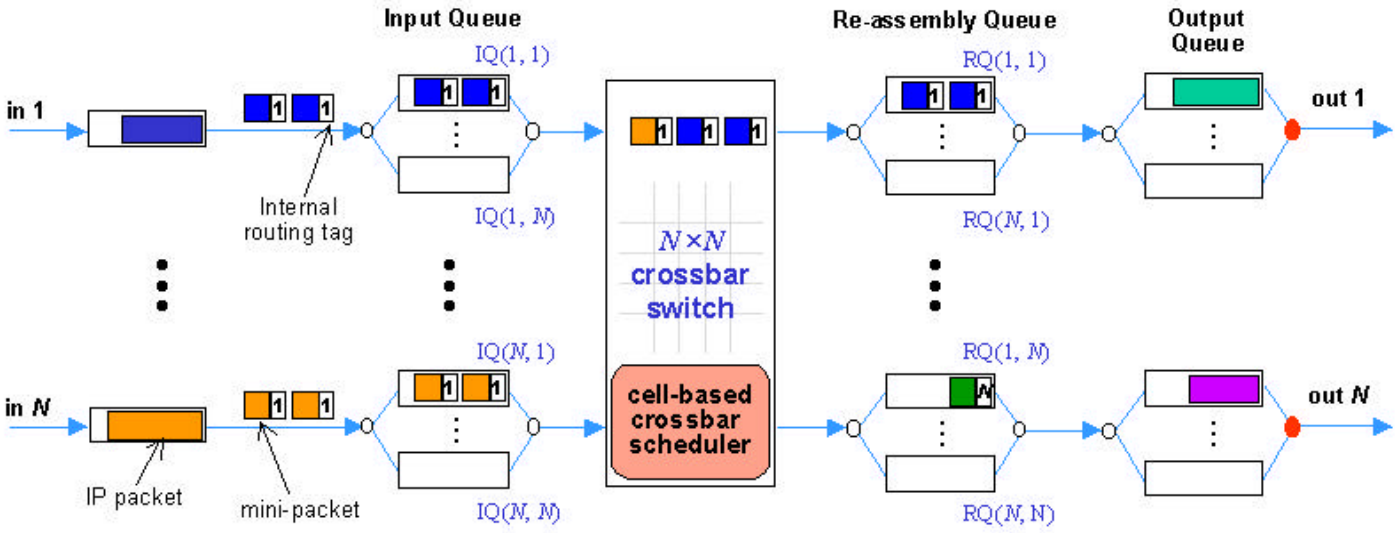
A Packet-Based Scheduling Algorithm for High-Speed Switches

August 20, 2001

Hakyong KIM

Dept. Info&Comm, K-JIST, Korea
E-mail : hykim@ieee.org

- **Contents**
  - Scheduling in High-Speed Switches
  - Packet-Based Scheduling Approach
  - Packet-Based SLIP (PBSLIP) Algorithm
  - Performance Evaluations and Remarks

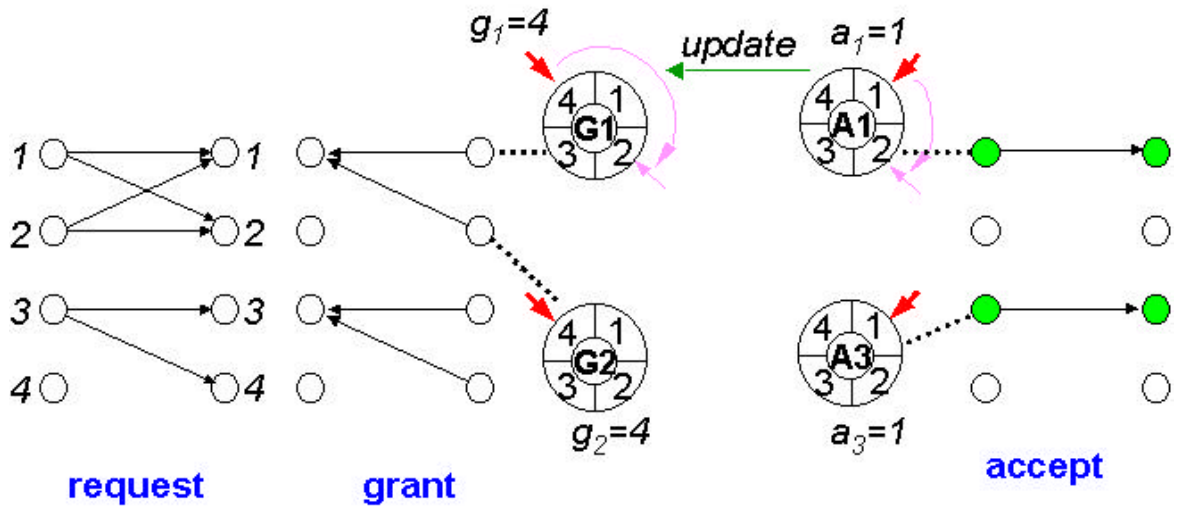


Switch architecture for packet-based scheduling

- **Scheduling in High-Speed Switches**
  - Data traffic in the real world are bursty.
  - The burst data should be segmented into a bunch of cells or mini-packets at first for high-speed switching or routing.
  - Mini-packets are switched in the switching core.
  - Mini-packets are reassembled into the original packet.

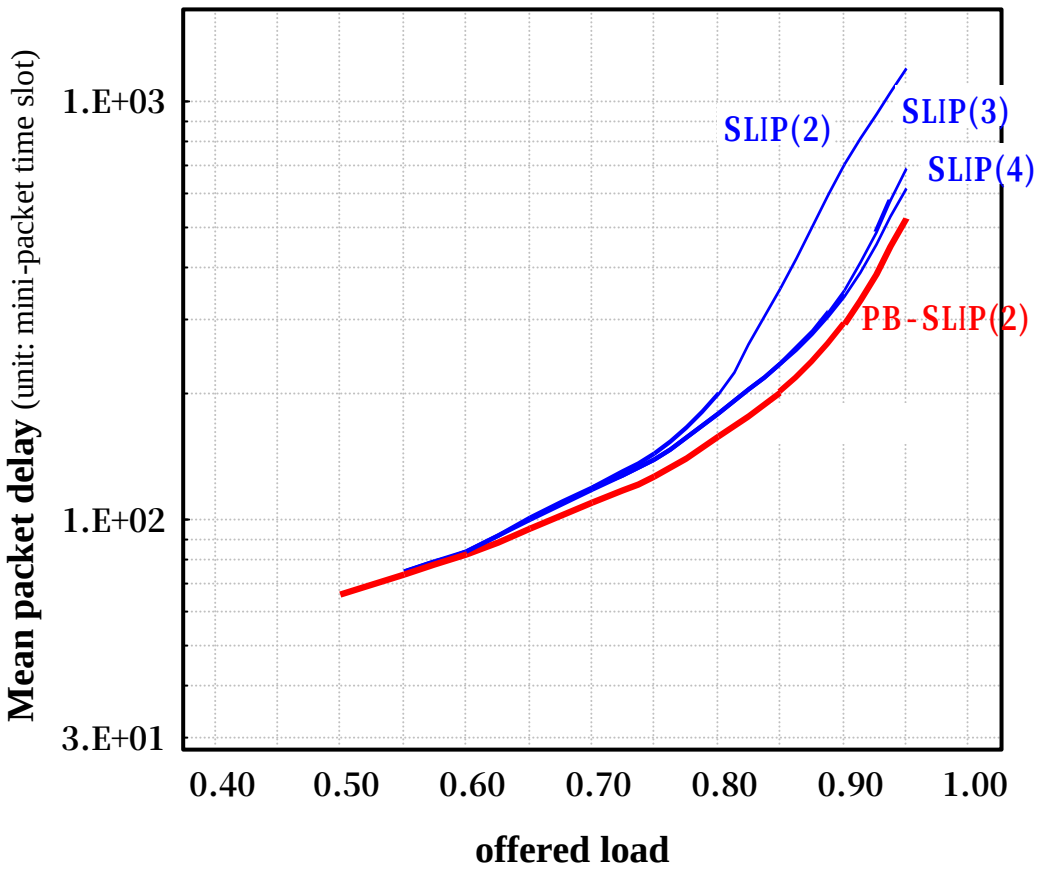
segmentation → cell switching → re-assembly

- **Packet Switching Approaches in High-Speed Switches/Routers**
  - Cell-based scheduling
  - Packet-based scheduling
- **Cell-Based Scheduling Algorithm**
  - Packets are switched at the cell level.
  - Mini-packets belonging to the same packet are switched individually, irrespective of the correlation to the packet destination.
  - not suitable for switching/routing IP packets
- **Packet-Based Scheduling Algorithm**
  - Packets are switched at the packet level.
  - If the first mini-packet is scheduled, the following mini-packets belong to the same packet are switched continuously in the following time slots.
    - Time complexity could be reduced.
  - Switching core bases its operation on cell-based or cell-by-cell switching as the cell-based switching.
  - Suitable for switching/routing IP packets
    - Performance metrics are specified by data units (packets) not by mini-packets.
  - Does not require re-assembly queue



Operation of Packet-Based SLIP Algorithm

- **Packet-Based SLIP Algorithm**
  - Basically based on the operation of SLIP
    - Three-phase algorithm: request, grant, accept
    - Parallel operation in each phase.
    - Round-robin guarantees the fairness.
  - Differences from SLIP
    - Only when the first mini-packet, destined for output j from input i, is accepted in Accept phase, the value of pointers are fixed as $g_j = i$ and $a_i = j$ as long as the last mini-packet is switched.
    - Only when the last mini-packet is accepted for transmission, the pointers are updated as $g_j = (i+1) \bmod N$ and $a_i = (j+1) \bmod N$.



Mean Packet Delay of PBSLIP

- **Performance Evaluation of PBSLIP**
  - Assumptions
    - 16 x 16 switches
    - Memory size: 1000 cells/queues
    - Packet arrival : ON-OFF (2-MMBP)
    - Mean packet length : 16 mini-packets
  - 2 iterations of PBLISP overwhelms 4 iterations of SLIP.

load	SLIP					PBSLIP				
	1	2	3	4	å	1	2	3	4	å
0.70	9.59	1.54	.037	.000	11.166	10.94	.223	.001	.000	11.171
0.75	9.81	2.05	.095	.000	11.958	11.66	.296	.002	.000	11.961
0.80	9.97	2.56	.229	.003	12.765	12.38	.387	.004	.001	12.768
0.85	10.12	2.95	.444	.014	13.529	13.03	.488	.010	.003	13.648
0.90	10.31	3.20	.747	.058	14.318	13.71	.591	.017	.007	14.326
0.98	11.12	2.98	.848	.105	15.053	14.33	.703	.027	.016	15.076

Number of Input-Output Matchings in SLIP and PBSLIP

- **Performance Evaluation of PBSLIP**
  - PBSLIP finds most input-output matchings in 2 iterations.
    - Time complexity could be reduced by 50%.
  - Appropriate for high-speed switches/ routers