4. Time-Space Switching and the family of Input Queueing Architectures

4.1 Intro: Time-Space Sw.; Input Q'ing w/o VOQ's

4.2 Input Queueing with VOQ's: Crossbar Scheduling

4.3 Adv.Topics: Pipelined, Packet, Enveloppe Scheduling

4.4 Comb. Input-Output Q'ing (CIOQ) – Int. Speedup

4.5 Comb. Input-Crossp. Q'ing – Buffered Crossbars

4.6 Partitioned Crossbars

Manolis Katevenis

CS-534 – Univ. of Crete and FORTH, Greece

http://archvlsi.ics.forth.gr/~kateveni/534/

4.4 CIOQ 4.5 CICQ

Table of Contents:

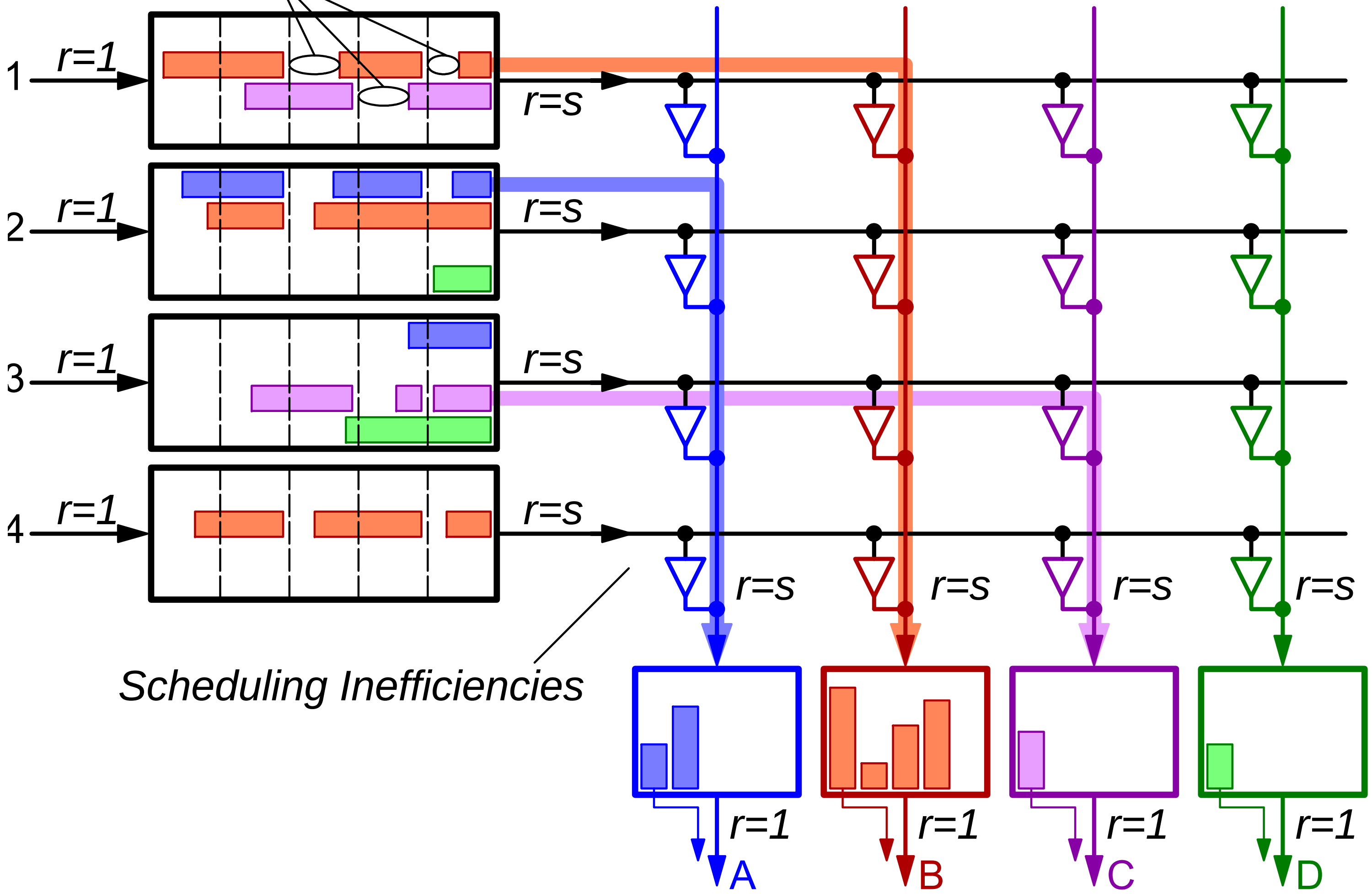
- Combined Input-Output Queueing (CIOQ) – Internal Speedup
  - Input Queued Crossbar under non-uniform traffic saturates well below peak capacity – the “Unbalanced” traffic pattern example
  - Speed up the internal crossbar by a factor of s ⇔ ensure that the input load stays always below 1/s of peak crossbar capacity
  - Theoretical results: Output Q’ing Emulation with speedup of 2
- Combined Input-Crosspoint Queueing (CICQ) – Buffered Xbar
  - Loosely-coupled, independent, single-resource schedulers
  - Approximate “matchings” yield better scheduling efficiency

4.4 Combined Input-Output Queueing (CIOQ)

Internal Speedup

(typically, the crossbar is 2 to 3 times faster than external lines)

Variable-Size Packets:
Segmentation Overhead



Scheduling Inefficiencies

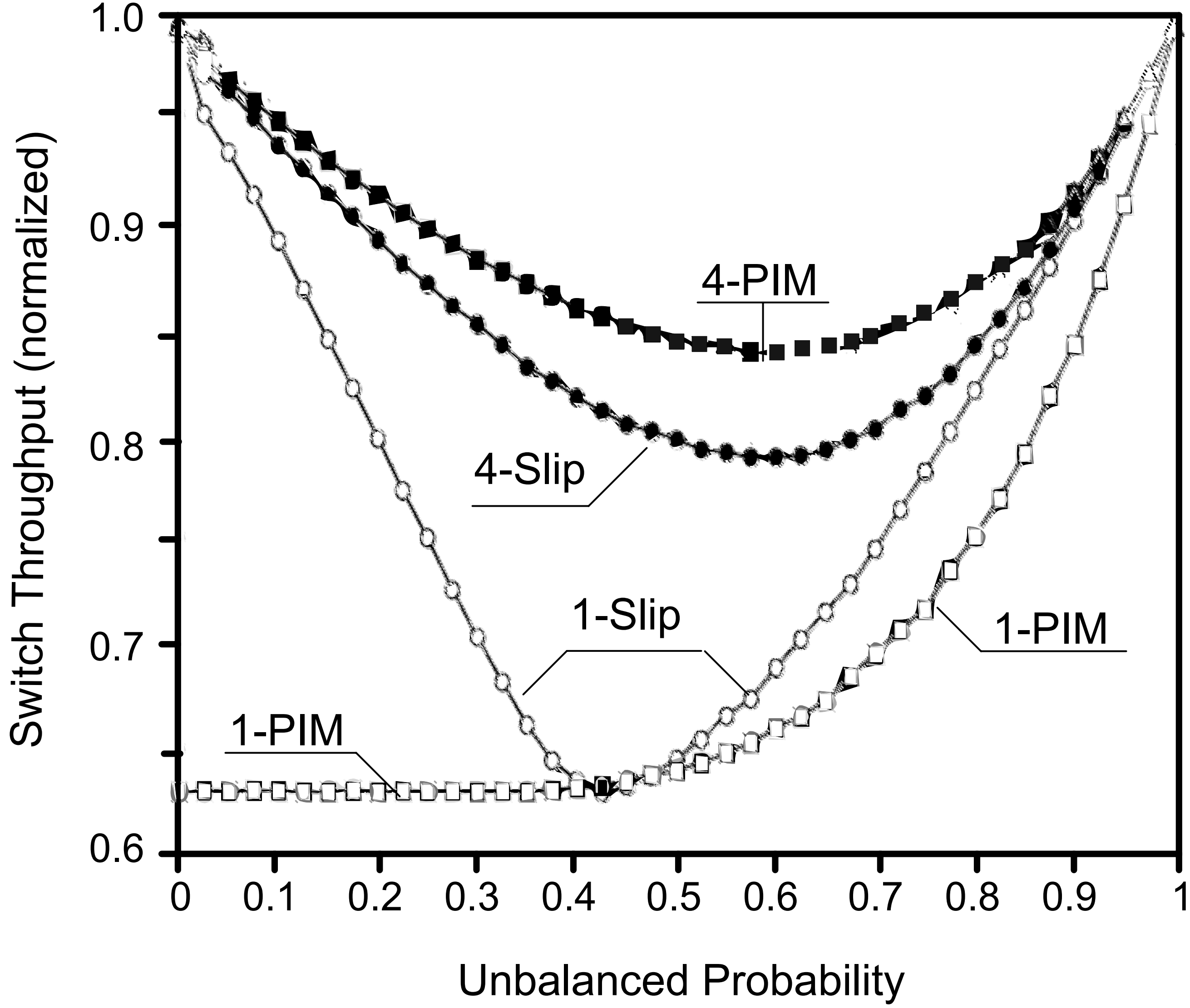
Internal Speedup – Combined Input-Output Queueing (CIOQ)

- Most widely used architecture in high-end internet switches
- Make the crossbar faster than the external lines, in order to:
  - compensate for the inefficiencies of the scheduler (e.g. unbal. traffic)
  - compensate for the segmentation overhead of variable-size packets
  - allow for separate (output) queues per QoS class
- Typical Speedup Factor values are between 2 and 3:
  - speedup of about 2 needed for variable-size packets (see § 2.2)
  - theoretical results: speedup of 2 suffices to emulate output queueing (using complex schedulers though – hard to totally unrealistic)
- The cost of Internal Speedup:
  - buffers at outputs too, increased throughput for crossbar & buffers
  - nowadays, increased throughput is too expensive (power consumpt’n) for off-chip communication ⇒ only use speedup *inside* switch chips, placing at least portion of input and output queues on-chip, with the rest of these queues on the line cards

Unbalanced Traffic: *simple example of hard traffic pattern*

- Each input has a “favored” output
  - “favored” input-output pairs are disjoint (they form a permutation)
- Each input sends traffic @ total rate = *load* as follows:
  - *(u × load)* to its favored output (u is the “unbalance factor), plus
  - *((1-u) × load)* to all outputs, uniformly distributed
  - ⇒ each output receives traffic @ total rate = *load*
- *“u”* is the “Unbalance Factor”:
  - u = 0 % ⇒ totally uniform traffic (usually easy)
  - u = 100 % ⇒ totally directional traffic (permutation) (often easy)
  - u = intermediate ⇒ … usually hard traffic …

Crossbar Sched. Perf. under Unbalanced Traffic



- 32×32 switch
- Saturation Throughput simulations: load = 100%
- Source: Rojas-Cessa e.a: “CIXOB-k: combined input-crosspoint-output buffered packet switch”, IEEE Globecom 2001

Can a CIOQ Sw. Emulate an Output Queued Switch?

- ***Full Emulation:***

  consider a CIOQ switch (combined input-output queueing, with internal speedup), and an OQ switch, both as “black boxes”. Consider precisely the same cells entering into both switches at precisely the same times. Full emulation is when the CIOQ switch will always forward to its outputs precisely the same cells as the OQ switch does, and at precisely the same times, for any arbitrary traffic pattern; i.e., an external observer is unable to tell which switch is which, no matter what traffic sequence (s)he injects.

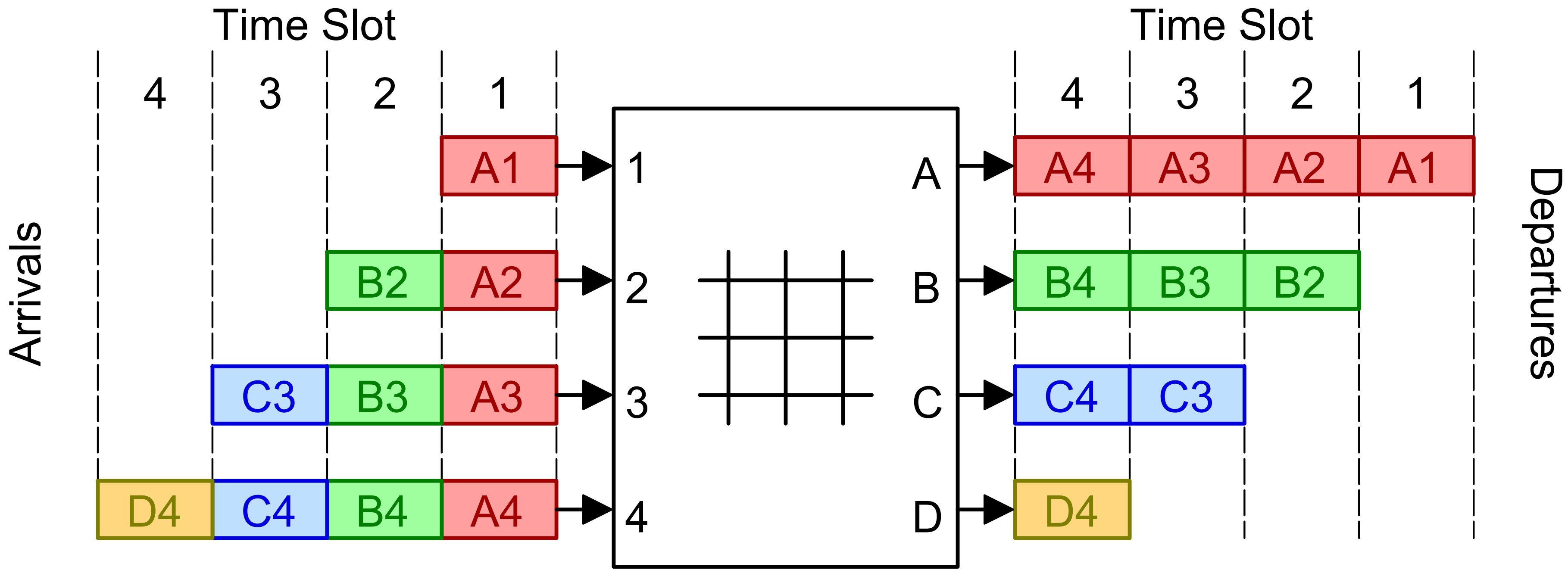
- ***Work-Conserving Operation:***

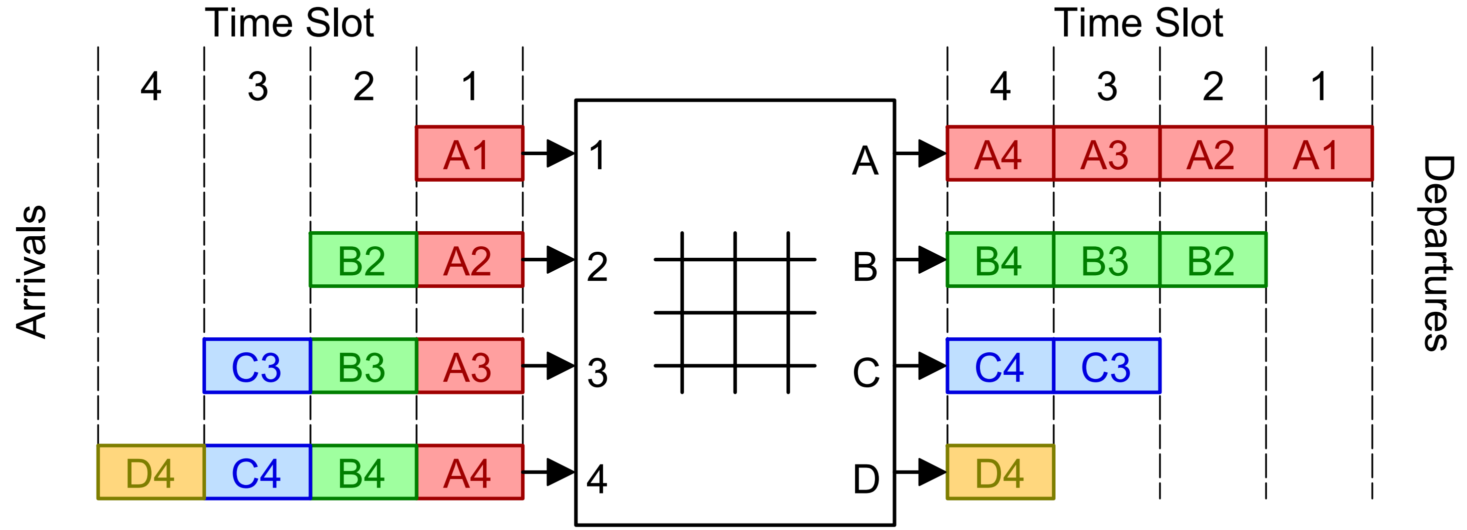
  no output port is ever left idle, except when there are no cells destined to it anywhere inside the switch. Hence, the outputs of the CIOQ switch will be busy (or idle) at precisely the same times as the corresponding OQ outputs, but *not necessarily* forwarding the exact same cell – may be forwarding another one of the cells destined to the same output (implies same *average* cell delay, due to “delay conservation” theorem for work-conserving switches).

Emulation of Output Q'ng by CIOQ with Speedup ≈ 2

- Results in IEEE JSAC, June 1999 (paper 1 by Chuang, Goel, McKeown, Prabhakar; paper 2 by Krishna, Patel, Charny, Simcoe):
- *Speedup = 2 - 1/N* is *necessary and sufficient* for a *N×N* CIOQ switch to *fully* emulate an OQ with FIFO output service
  - – necessary: see next two slides
  - – sufficient: need complex xbar scheduler – theoretical value only
- *Speedup = 2* is *sufficient* for CIOQ to *fully* emulate OQ with quite general service policies (PIFO – push-in first-out)
  - – need complex crossbar scheduler – of theoretical value only
- *Speedup = 2* is *sufficient* for a CIOQ switch that uses *LOOFA* scheduler to be *Work-Conserving*
  - – *LOOFA:* Lowest-Occupancy Output First Algorithm – maximal match where the shallowest output queues are connected first

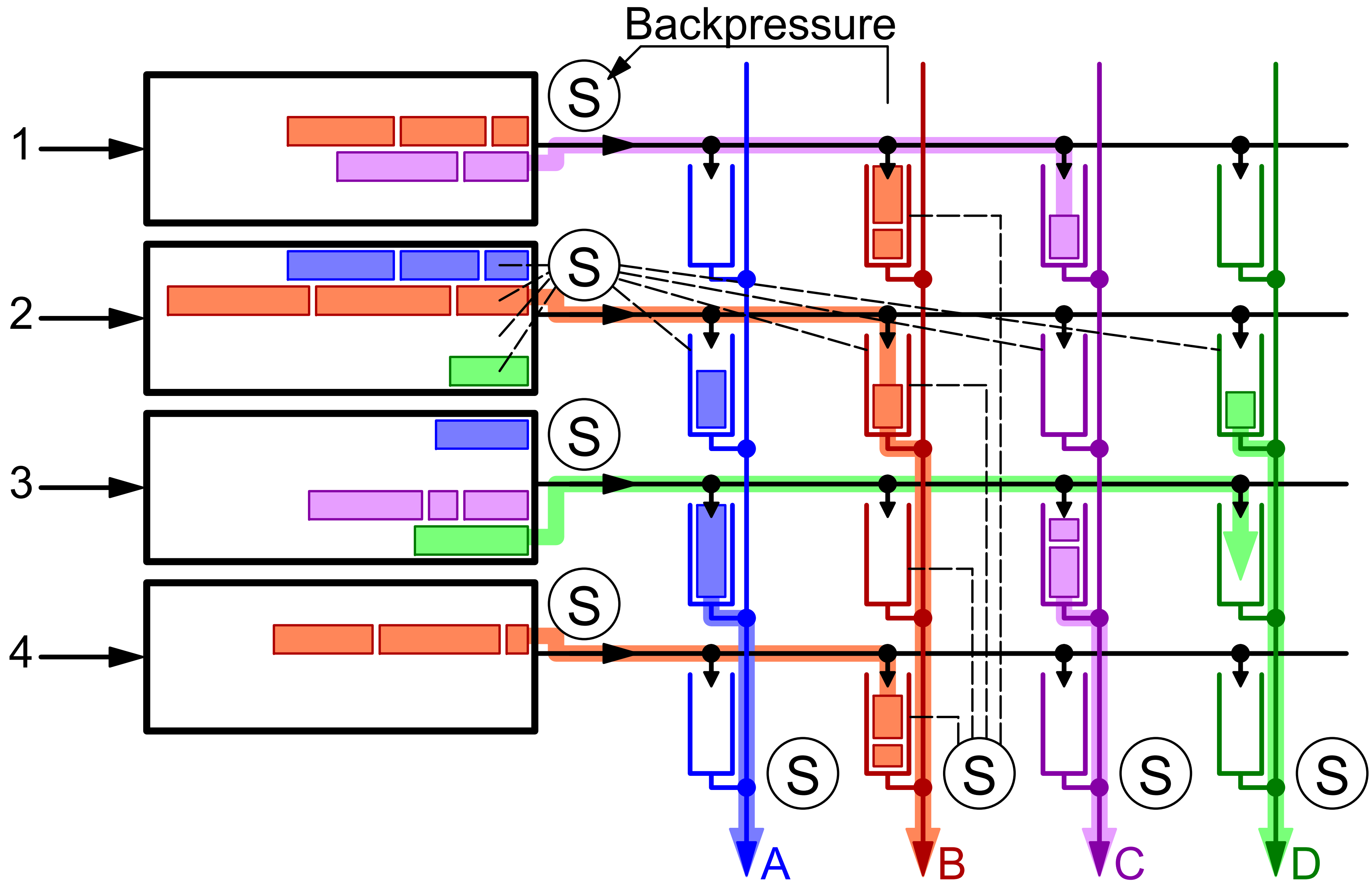
Traffic pattern to prove the lower bound of s = 2 – 1/N of speedup that is necessary for CIOQ to emulate OQ





Phase	PA	PB	PC	PD	PA	PB	PC	PD	PA	PB	PC	PD	PA	PB	PC	PD
1	**A1**				**A1**				**A1**				**A1**			
2					**A2**				**A2**				**A2**			
3		**B2**				**B2**			**A3**	**B2**			**A3**	**B2**		
4						**B3**				**B3**			**A4**	**B3**		
5			**C3**				**C3**			**B4**	**C3**			**B4**	**C3**	
6							**C4**				**C4**				**C4**	
7				**D4**				**D4**				**D4**				**D4**

4.5 Buffered Crossbars (CICQ)



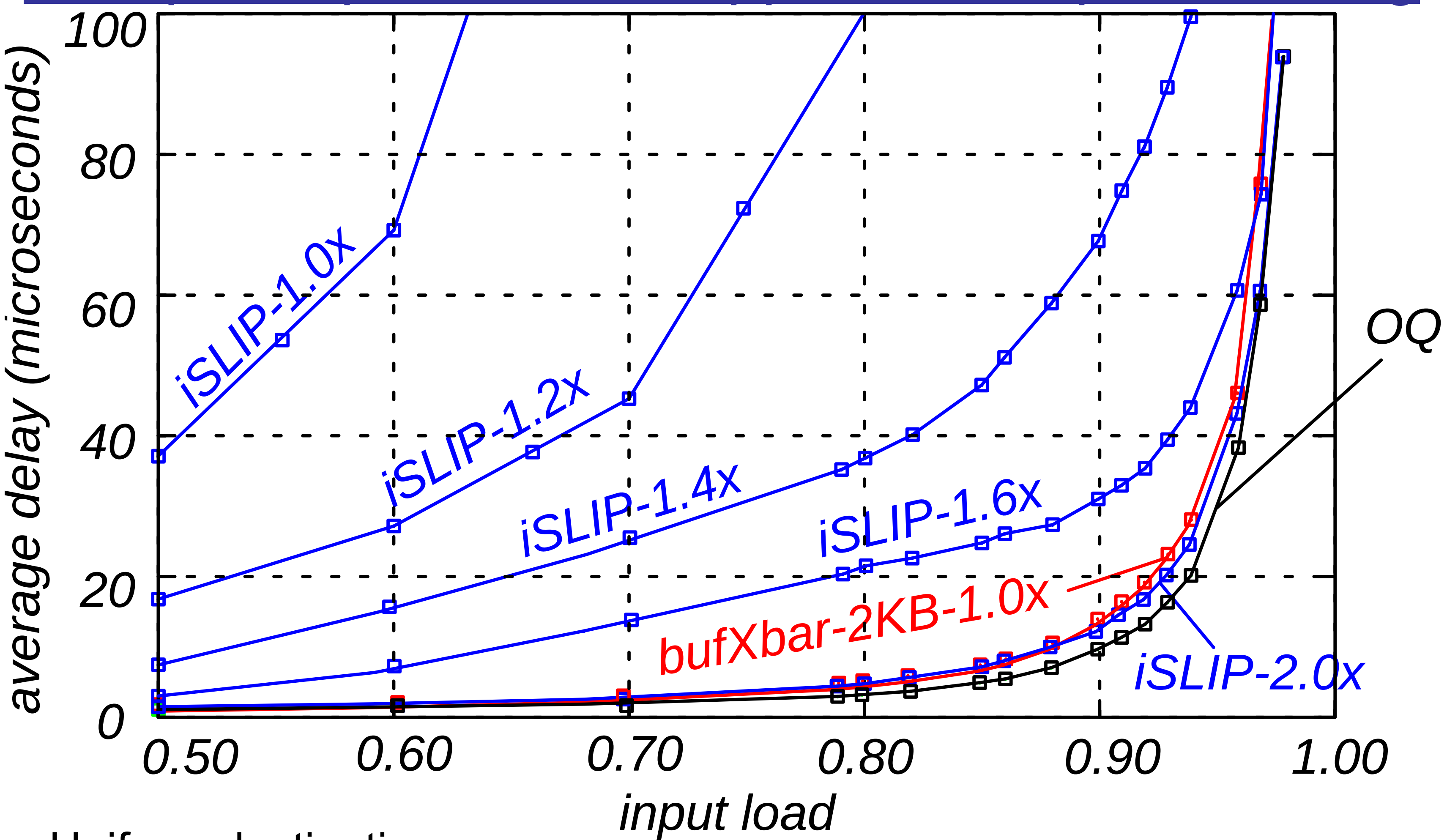
Buffered Crossbars, or Comb. Input-Crosspoint Q’ng

- *Small* buffers per crosspoint, large buffers per input
- Backpressure from crosspoint buffers to VOQ’s at inputs
- Loosely-coupled, independent, single-resource schedulers
  - per-output schedulers decide which flow (crosspoint queue) to serve among the non-empty ones in each output’s column
  - per-input schedulers decide which flow to serve among the ones with non-empty VOQ and with credits available in each input’s row

⇒ Approximate “matchings” yield better scheduling efficiency

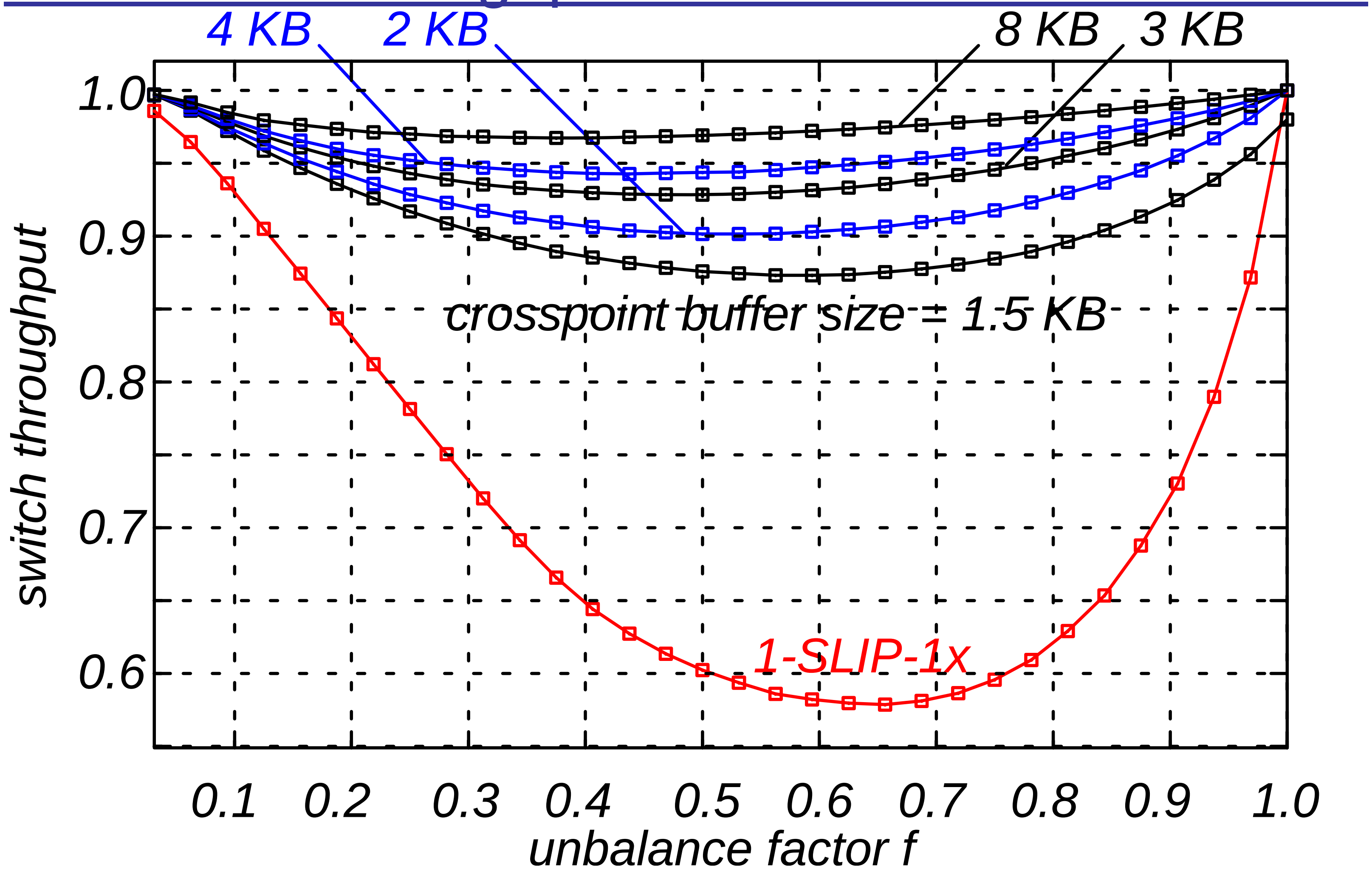
- in the short term, *(i)* multiple inputs may feed the same column (e.g. 2 and 4); *(ii)* multiple outputs may be fed by the same row (e.g. A and C)
- in the long run, these cannot persist, because *(i)* buffers in that column are filled faster than they get emptied, so they will fill-up; *(ii)* buffers in that row are being emptied faster than they get filled, so they will drain.

No Speedup needed to approach Output Queuing



- Uniform destinations
- Internet-style synthetic workload; 40-1500 byte packet sizes
- Unbuffered crossbar w. SAR: one-iteration iSLIP, 64-byte segments

Saturation Throughput under Unbalanced Traffic



- Poisson arrivals, Pareto sizes (40-1500)
- For iSLIP, packet sizes are multiples of 64 B (⇒ no SAR overhead)

Buffered Crossbars (CICQ) – References:

- D. Stephens, H. Zhang: “Implementing Distributed Packet Fair Queueing in a Scalable Switch Architecture”, INFOCOM 1998
- T. Javidi, R. Magill, and T. Hrabik: “A High-Throughput Scheduling Algorithm for a Buffered Crossbar Switch Fabric”, ICC 2001
- R. Rojas-Cessa, E. Oki, and H. Jonathan Chao: “CIXOB-k: Combined Input-Crosspoint-Output Buffered Switch”, GLOBECOM 2001
- Abel, Minkenberg, Luijten, Gusat, Iliadis: “A Four-Terabit Packet Switch Supporting Long Round-Trip Times”, IEEE Micro, Jan. 2003
- N. Chrysos, M. Katevenis: “Weighted Fairness in Buffered Crossbar Scheduling”, IEEE Wrksh. High Perf. Switching & Routing (HPSR) 2003

⇒ M. Katevenis, G. Passas, D. Simos, I. Papaefstathiou, N. Chrysos: “Variable Packet Size Buffered Crossbar (CICQ) Switches”, ICC 2004

- G. Passas, M. Katevenis: “Packet Mode Scheduling in Buffered Crossbar (CICQ) Switches”, IEEE W.High Perf.Sw.Rtng (HPSR) 2006