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

Manolis Katevenis

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

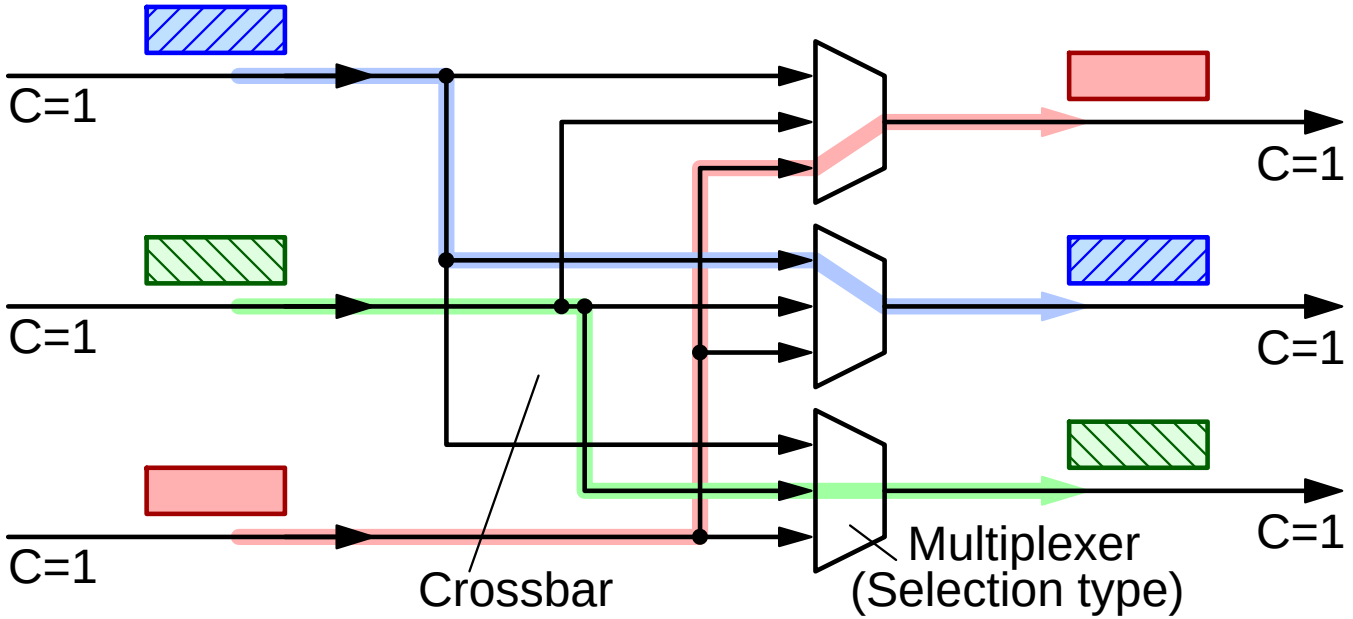
`http://archvlsi.ics.forth.gr/~kateveni/534/`

4.1 Introduction to Input Queueing

Table of Contents:

- Time-Space Switching
  - the crossbar scheduling problem
- “Naïve” Input Queueing without VOQ’s (Input Buffering)
  - independent per-output schedulers
  - head-of-line (HOL) blocking
  - performance analysis results

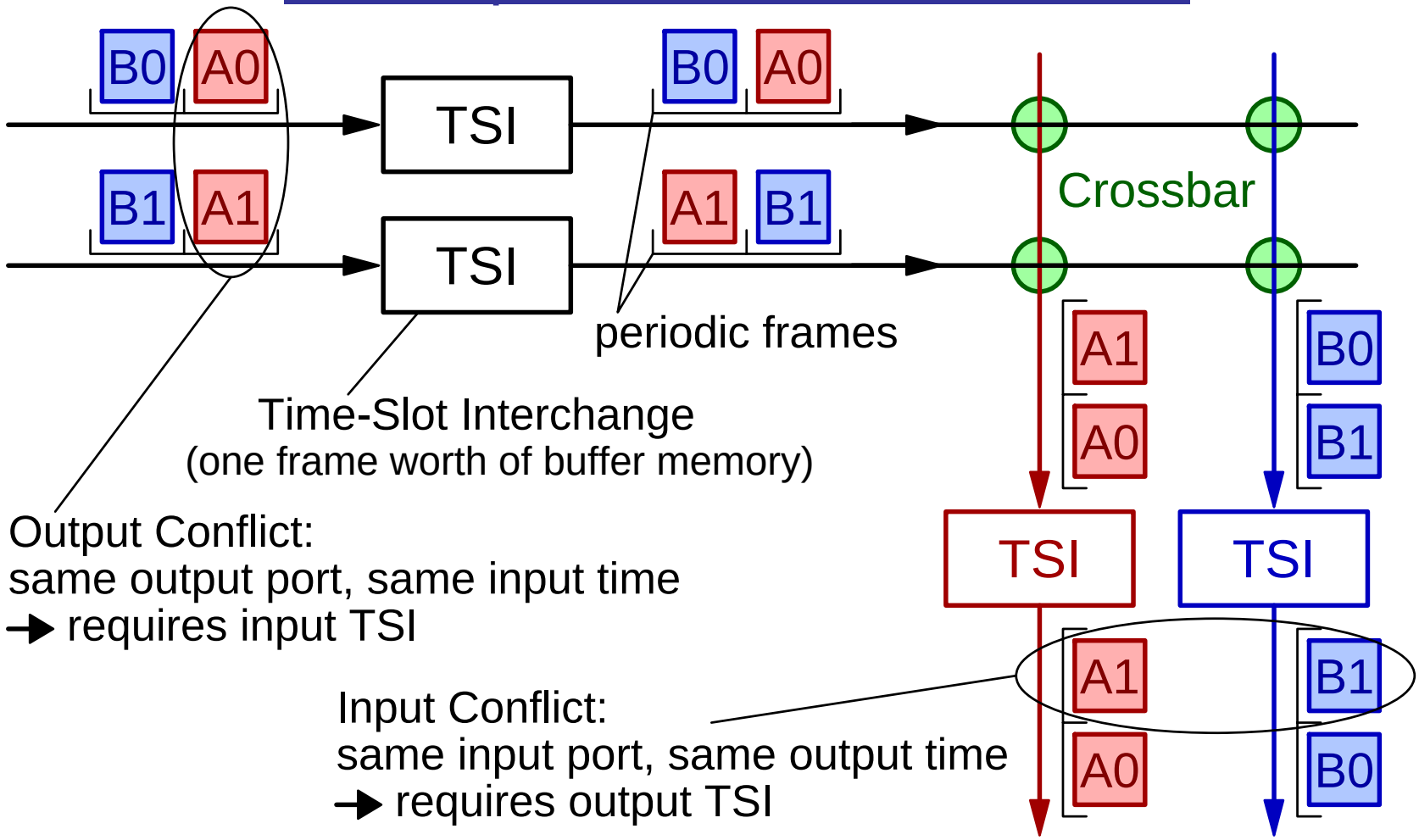
(Single-Stage) Space Switching: the Crossbar



- when a single (time-switched) path does not suffice for aggregate traffic…
- Space Switching: multiple simultaneously-active paths exist
- Crossbar: the single-stage, non-blocking space switch
- Output Contention: no two inputs allowed to simultan'sly feed same output

$\Rightarrow$ need input buffers to alter the time relationship of conflicting packets

Time-Space-Time Circuit Switch



- Time switching (TSI's) needed to resolve output and input conflicts
- In packet switching: no strict output timing req'mnts ⇒ may skip output TSI

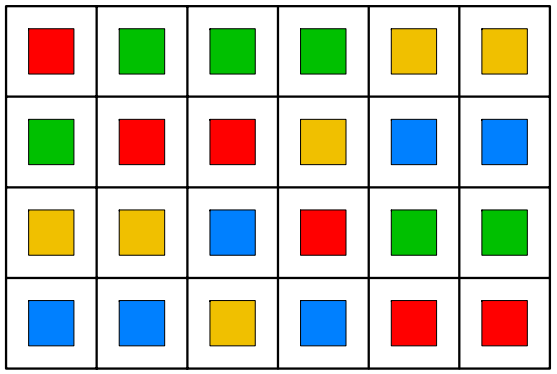
The *Static* Crossbar Scheduling Problem in Ckt Sw'ng

number of conn. to output...

number of conn. from input...

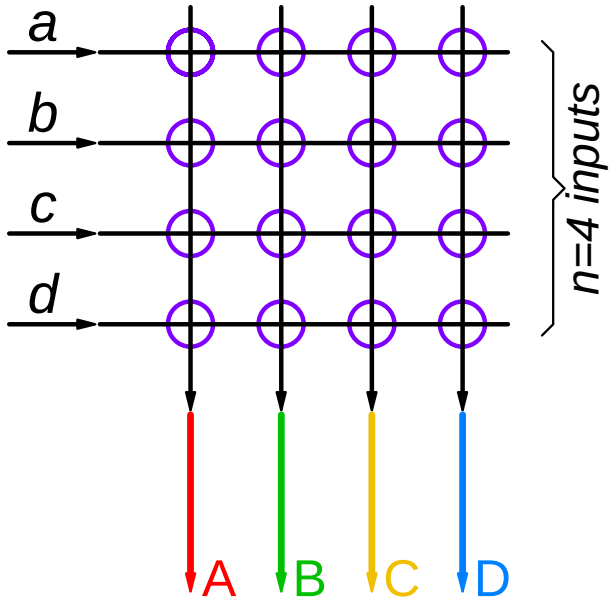
	A	B	C	D	*Tot.*
a	1	3	2	0	6
b	2	1	1	2	6
c	1	2	2	1	6
d	2	0	1	3	6
Tot.	6	6	6	6	

Crossbar Schedule:



m=6 slots per frame

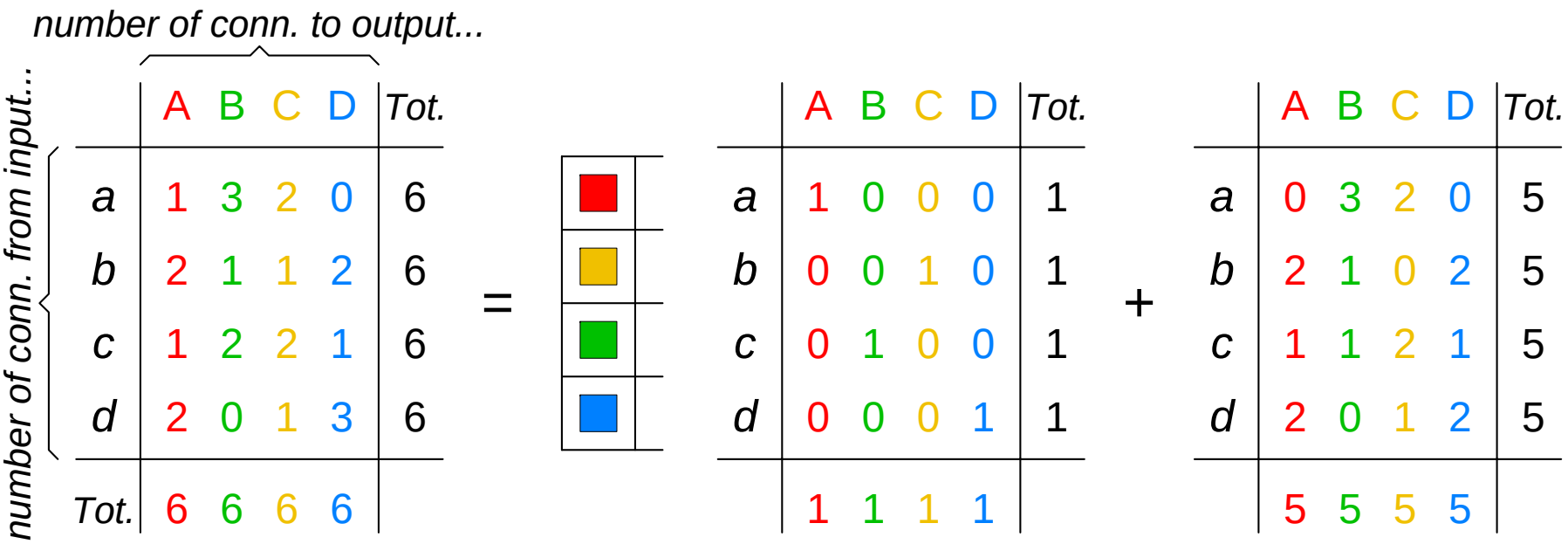
n=4 outputs



n=4 inputs

- Given the number of connections requested from each input to each output (left matrix), find a schedule (middle)

Building the Schedule one Step at a Time



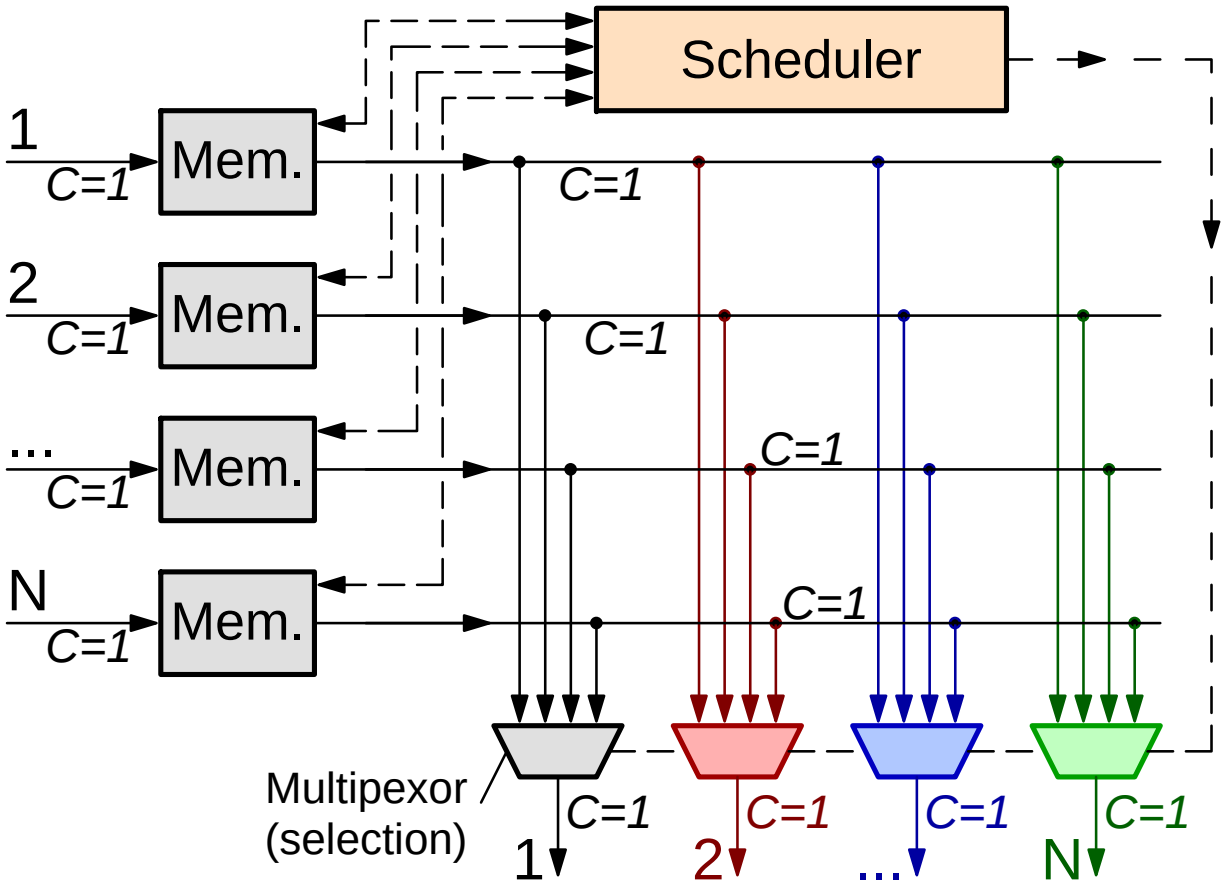
- Decompose the (full) request matrix into the sum of a permutation matrix plus a “smaller” (full) request matrix

  [not obvious how to do –e.g. what if we start b-A, c-B, d-C ?]

- What about requests arriving one-at-a-time? (exercises 9)

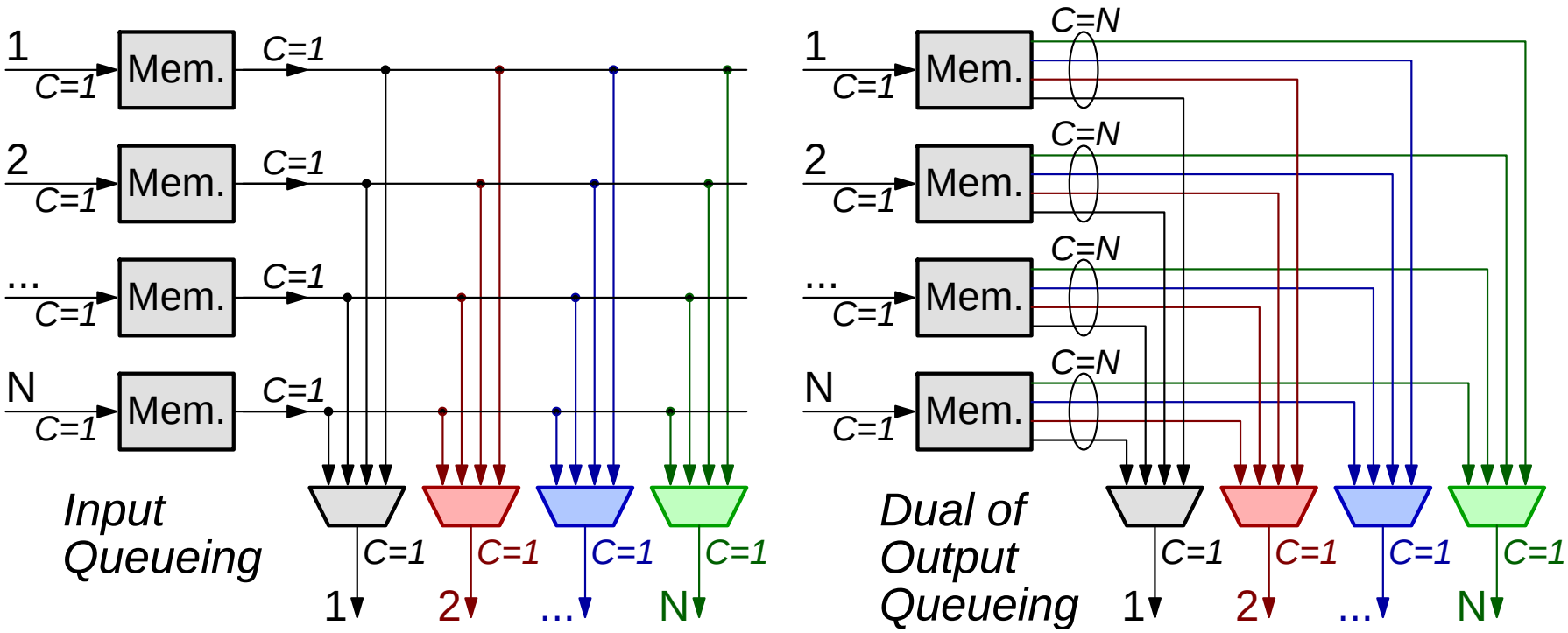
The Input Queueing Family in Packet Switching

- Per-input buffer memories
- \+ Scalable: each memory needs throughput = 2 ⇒ feasible *independ. of N*
- \+ Low cost: total memory thruput = *2·N* – same as shared buffer



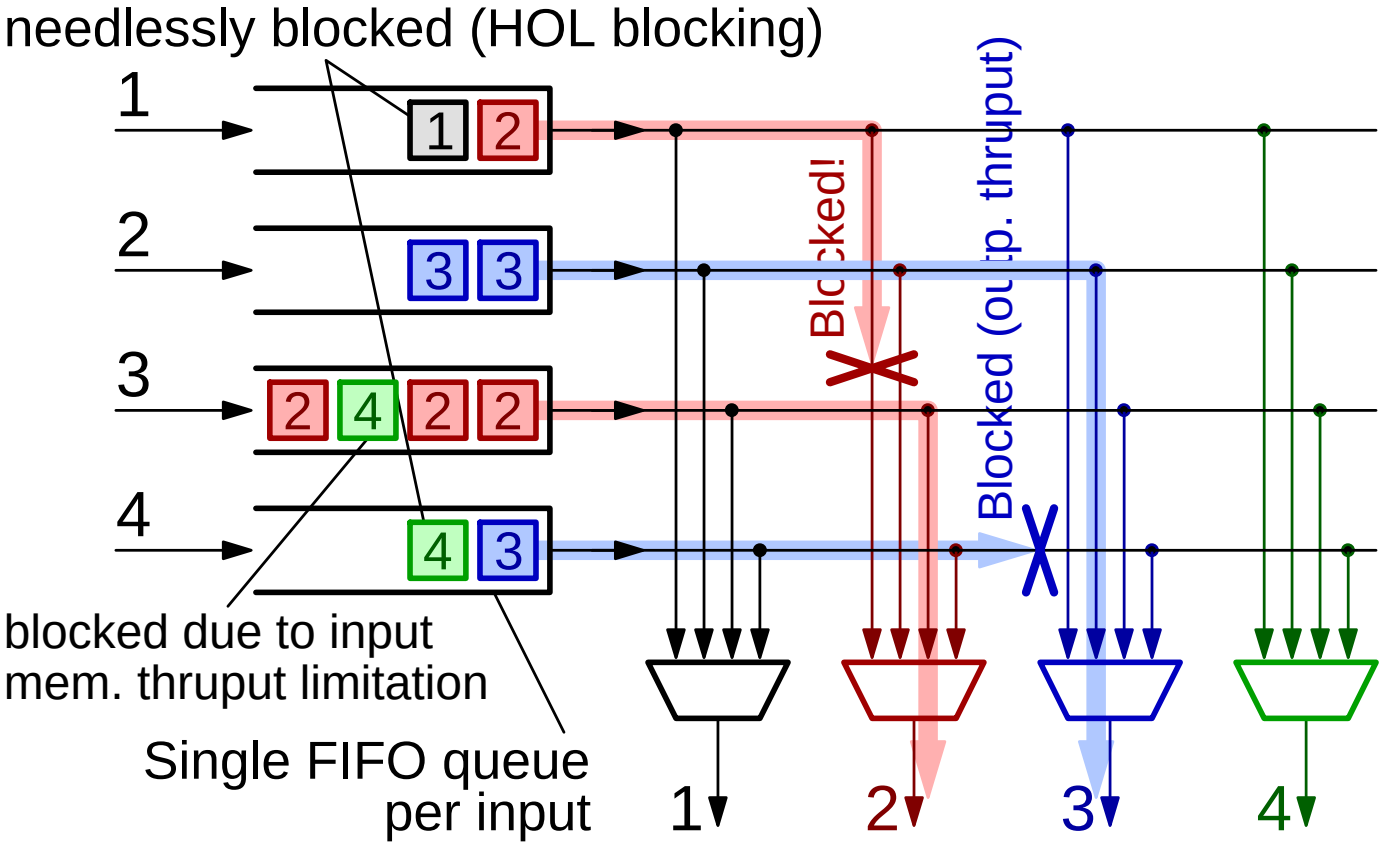
- Performance suffers a lot, unless *(i)* multiple queues per input, and *(ii)* sophisticated scheduler, and usually *(iii)* other modifications to be seen later (small crosspoint Q's, or internal speedup and OQ's)

Input Queueing is ***not*** the dual of Output Queueing



- **Asymmetry** between packet arrival and packet departure conflicts:
- Simultaneous **arrivals** may conflict with each other (packets destined to the same output), and the switch is *obliged to accept* them.
- Simultaneous **departures** are *scheduled by the switch* $\Rightarrow$ can be made to *not* originate from the same input – albeit at potential performance cost.

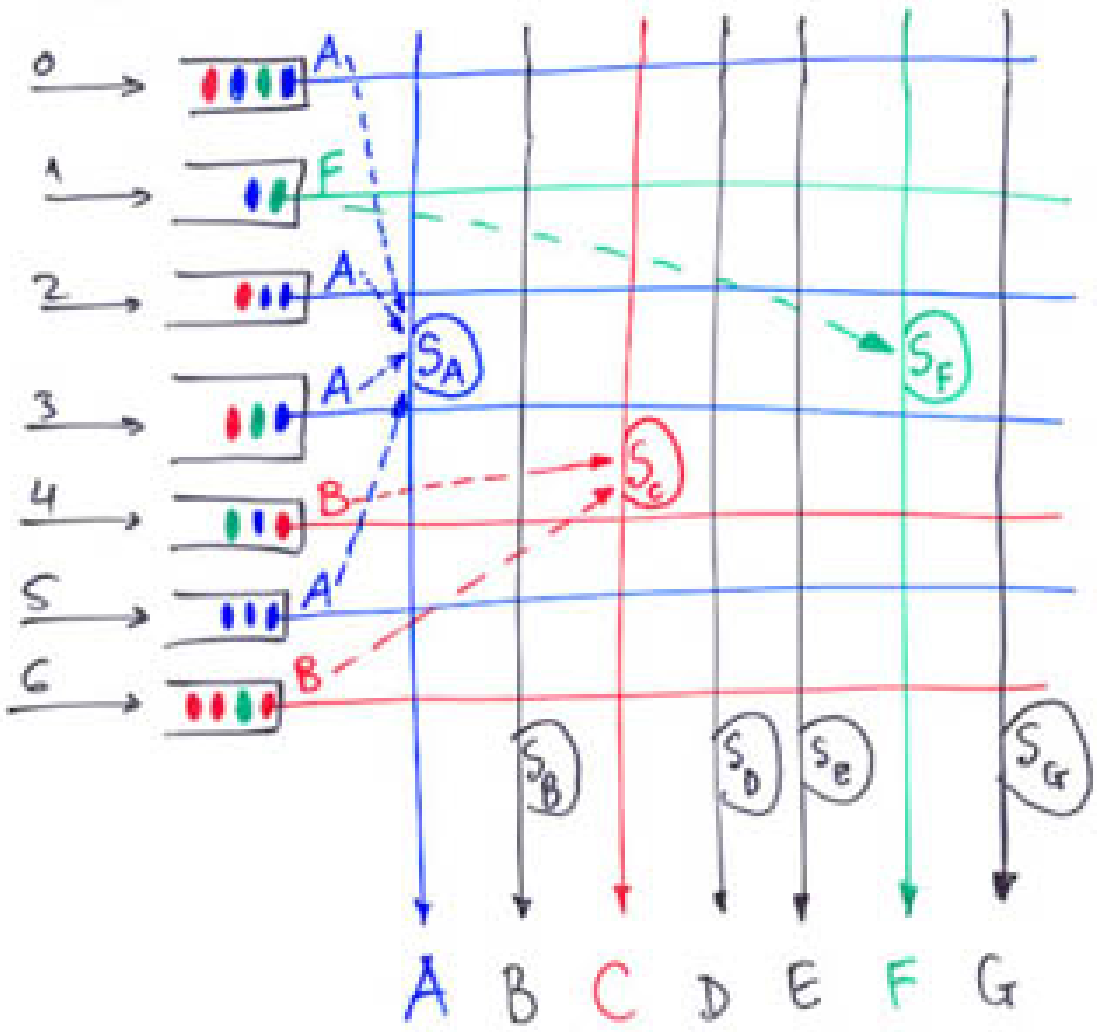
Single Queue per Input ⇒ Head-of-Line (HOL) Blocking



- Whenever one First-in-First-Out (FIFO) queue feeds multiple destinations, beware of the danger of head-of-line (HOL) blocking

  (called HOL-blocking when bottleneck is the FIFO organization – *not* when other bottleneck, e.g. memory read throughput or output port throughput)

Independent Per-Output Schedulers



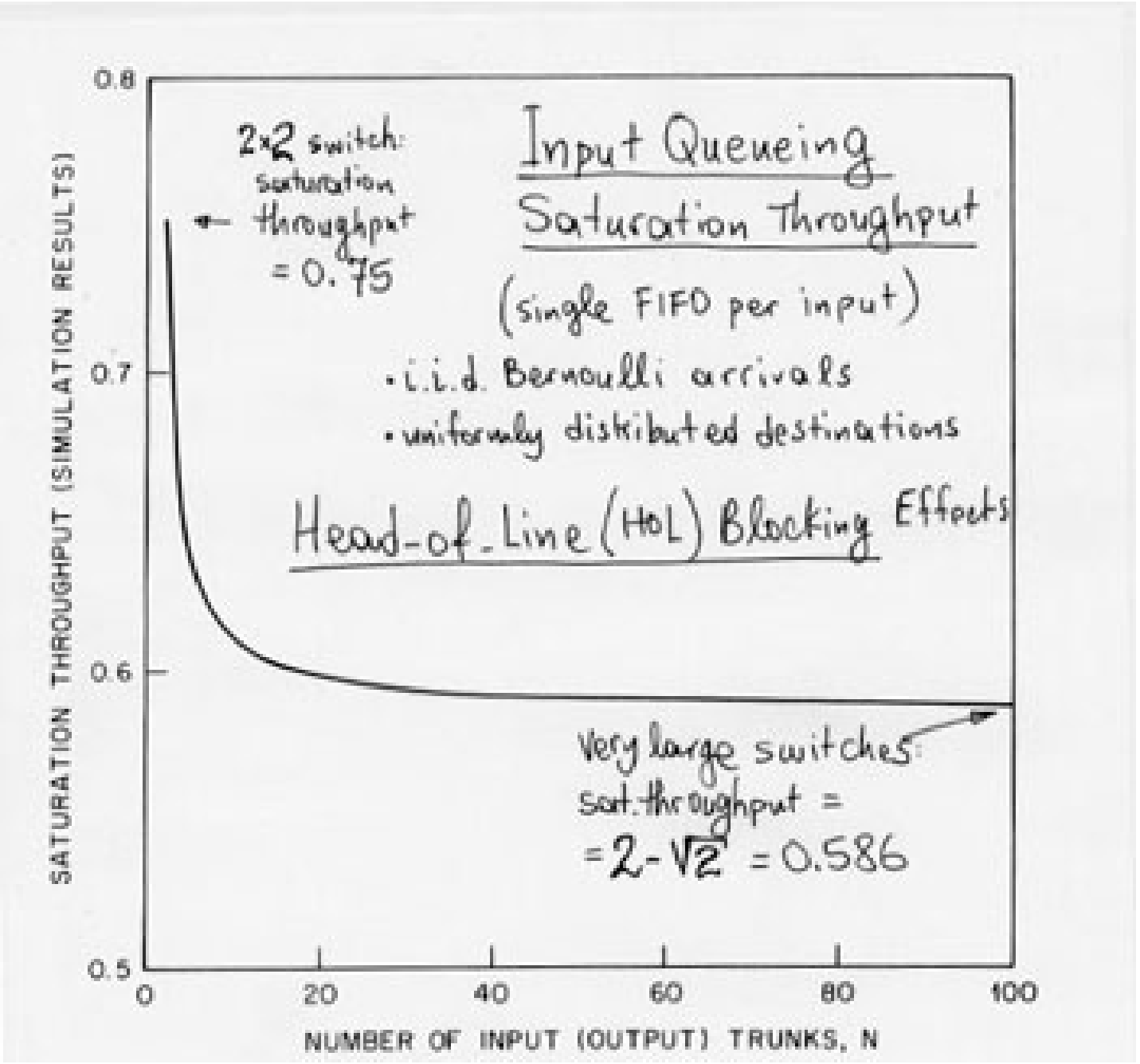
- with old-fashioned Input Q.
- (as with crosspoint queuing (similar to output Q or shared buffer)).

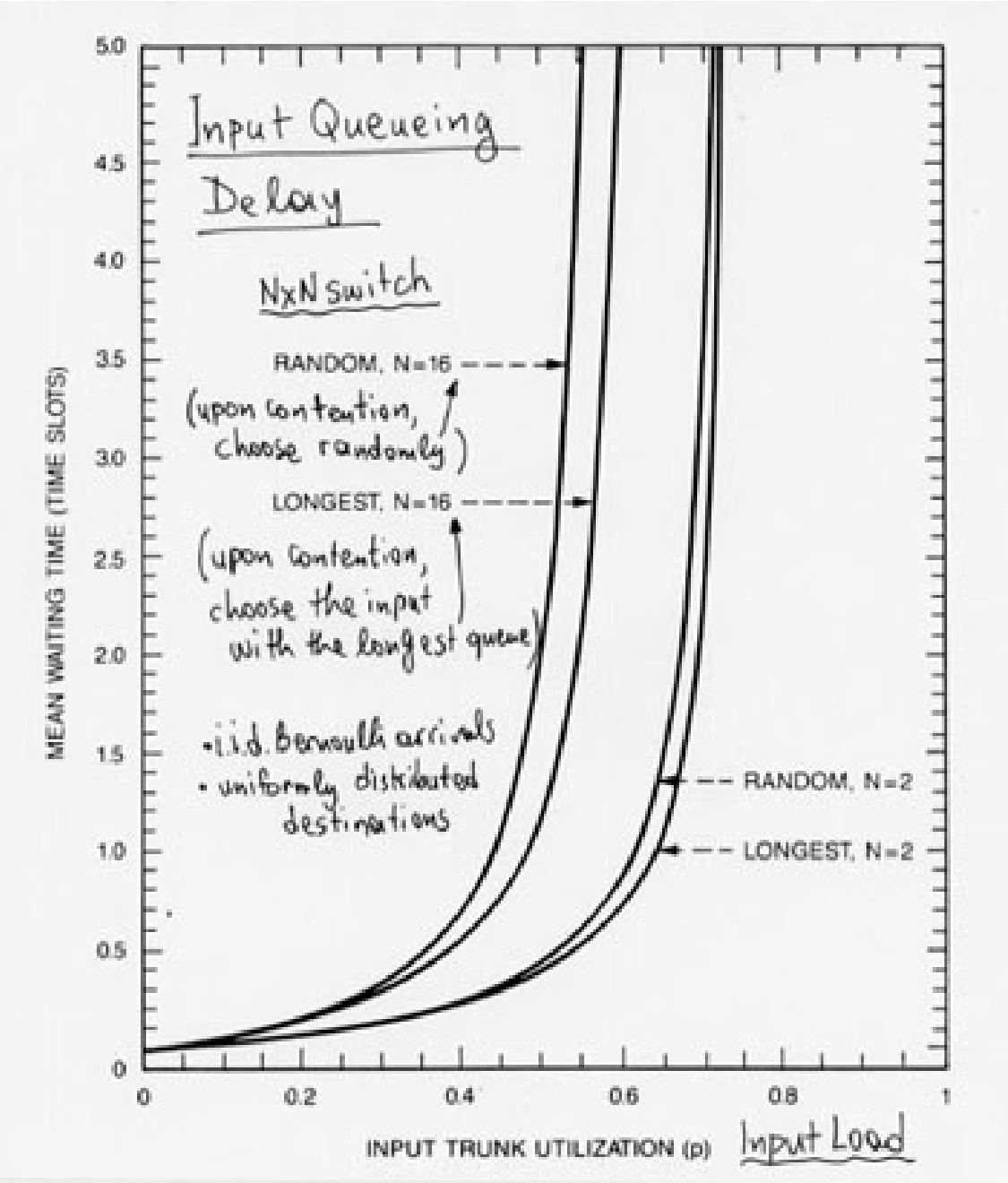
- each output has its own scheduler
- each scheduler sees a distinct set of requests
- each scheduler independently makes a decision
- no better decision is possible by considering the other schedulers' decisions.

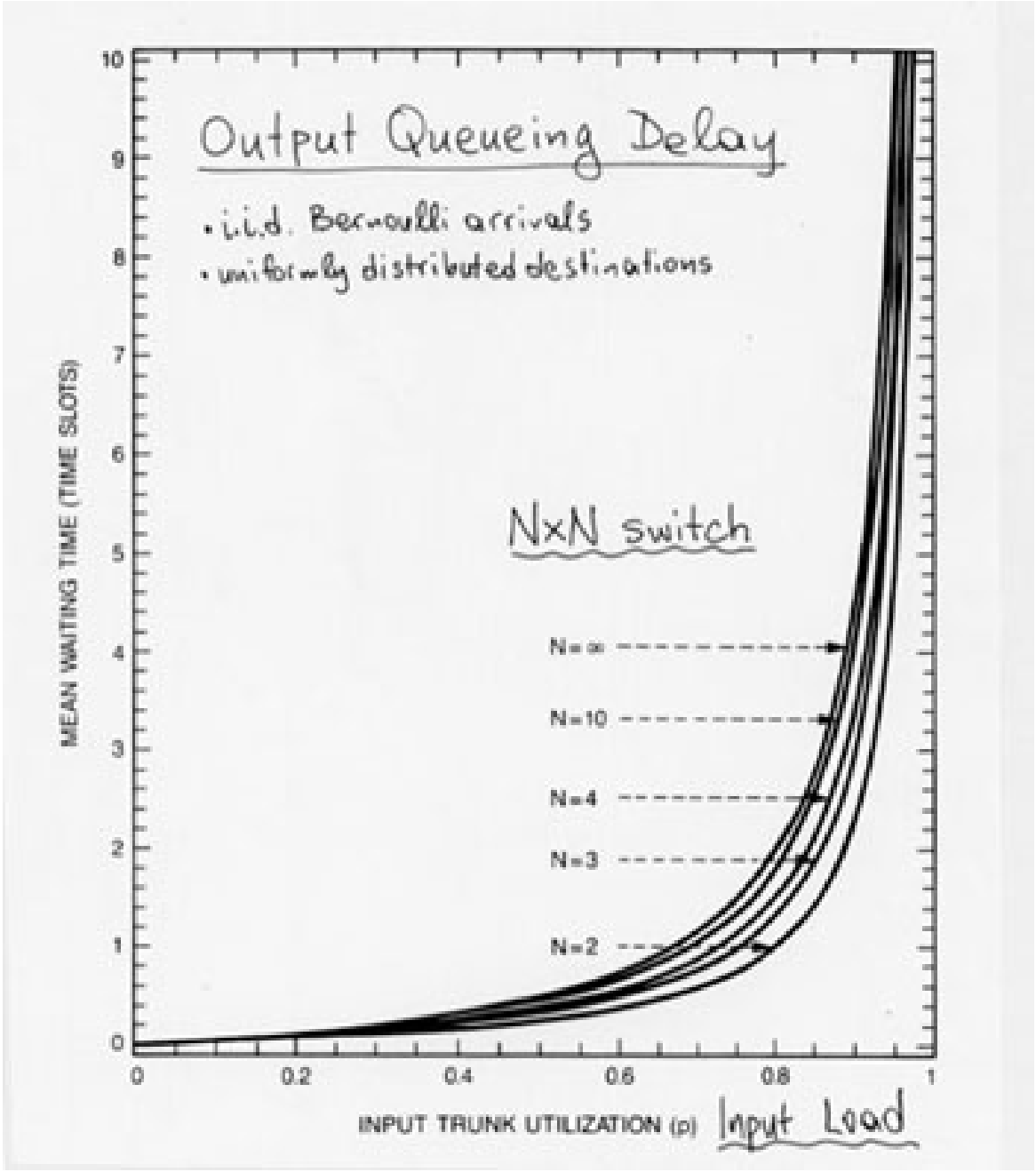
... unlike the next scheme (adv. int. buffering - VOQ's) ....

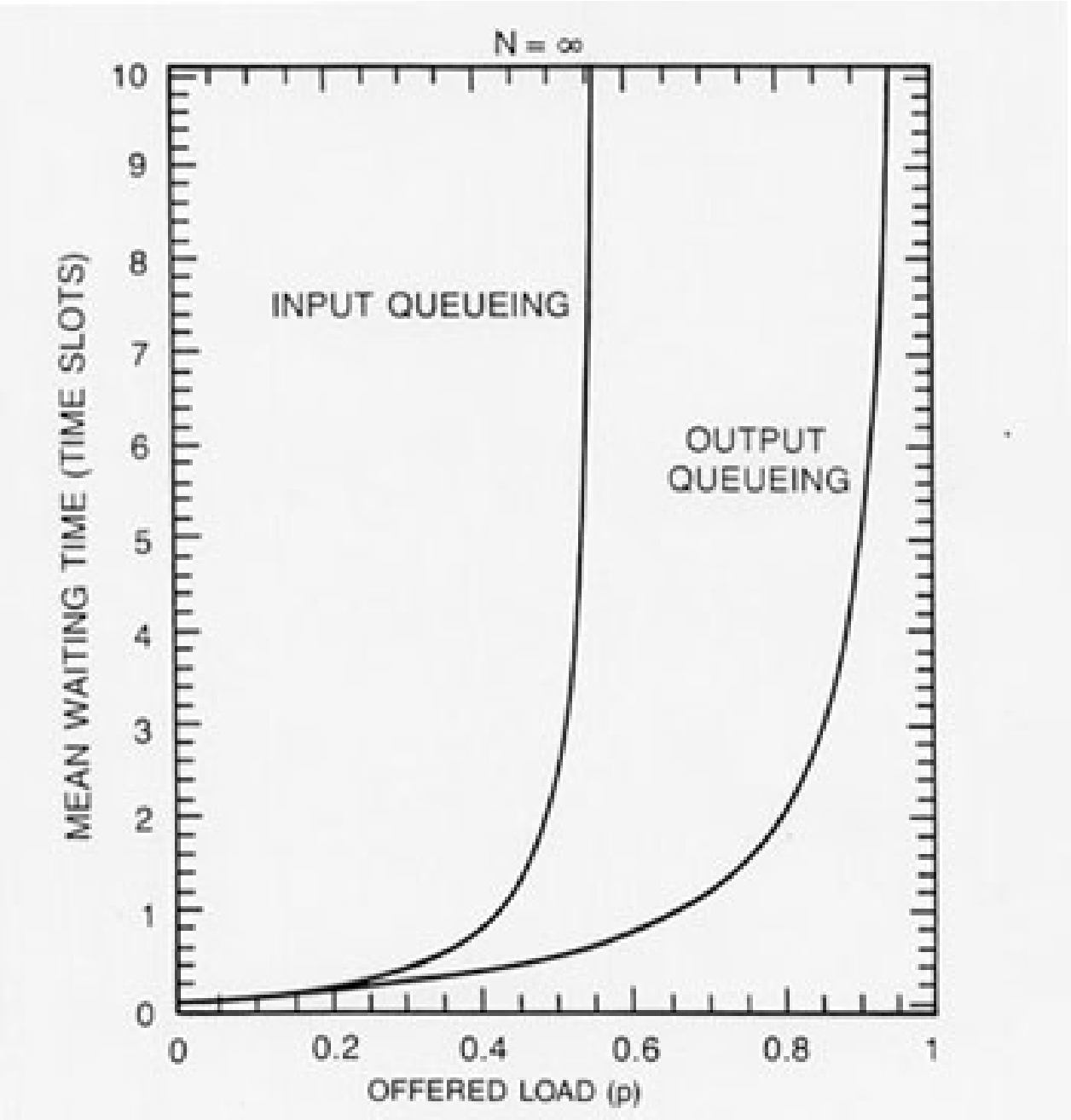
Throughput & Delay under Single-Queue Input Q'ing

- References:
  - M. Hluchyj, M. Karol: "Queueing in High-Performance Packet Switching", IEEE Journal on Sel. Areas in Commun. (JSAC), Dec. 1988, pp. 1587-1597.
  - M. Karol, M. Hluchyj, S. Morgan: "Input versus Output Queueing on a Space-Division Packet Switch", IEEE Tr. on Communic., Dec. 1987, pp. 1347-1356.
  - J. Hui, E. Arthurs: "A Broadband Packet Switch for Integrated Transport", IEEE Journal on Sel. Areas in Commun. (JSAC), Oct. 1987, pp. 1264-1273.
- *Attention:* results for i.i.d. Bernoulli (non-bursty) arrivals, with uniformlydistributed destinations (no overloaded hot-spots), are only useful for gaining a rough, first insight into the behavior of systems, but are often not representative of the real behavior of systems under real traffic!...


Input Queueing
Saturation Throughput
(single FIFO per input)
• i.i.d. Bernoulli arrivals
• uniformly distributed destinations
2×2 switch: saturation throughput = 0.75
Head-of-Line (HoL) Blocking Effects
Very large switches: sat. throughput = 2 - √2 = 0.586
SATURATION THROUGHPUT (SIMULATION RESULTS)
0.8
0.7
0.6
0.5
0
20
40
60
80
100
NUMBER OF INPUT (OUTPUT) TRUNKS, N


Input Queueing
Delay
NxN switch
RANDOM, N=16
(upon contention, choose randomly)
LONGEST, N=16
(upon contention, choose the input with the longest queue)
• i.i.d. Bernoulli arrivals
• uniformly distributed destinations
RANDOM, N=2
LONGEST, N=2
MEAN WAITING TIME (TIME SLOTS)
INPUT TRUNK UTILIZATION (p)
Input Load
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
0.2
0.4
0.6
0.8
1


Output Queueing Delay
• i.i.d. Bernoulli arrivals
• uniformly distributed destinations
NxN switch
N=∞
N=10
N=4
N=3
N=2
MEAN WAITING TIME (TIME SLOTS)
INPUT TRUNK UTILIZATION (p)
Input Load
0
1
2
3
4
5
6
7
8
9
10
0.2
0.4
0.6
0.8
1


N = ∞
MEAN WAITING TIME (TIME SLOTS)
10
9
8
7
6
5
4
3
2
1
0
INPUT QUEUEING
OUTPUT
QUEUEING
0
0.2
0.4
0.6
0.8
1
OFFERED LOAD (p)