Performance of Switching Networks

(A general view based on a simple model)

J-P Dufey, CERN

Outline:

- Overview and Definitions
- Non Blocking vs Blocking Switches
- Input vs Output Queueing
- Simulation Model
- Performance of the Various Architectures
- Review of some standard technologies
- Conclusions

Factors that determine the Performance of a Switching Network

1) Performance of point to point links

a) Bandwidth: <= network link bandwidth
(may be limited by internal bandwidth (e.g. PCI) in source and destination modules)

b) Overheads in sources and destinations

Analysis of point to point links does not require a network => direct measurements.

This is not the object of this presentation.

2) Performance of the switching network

Interaction between channels simultaneously active (blocking, contention)
Depends on:

- technology
- switch architecture
- type of traffic: random vs coherent (i.e. event building)

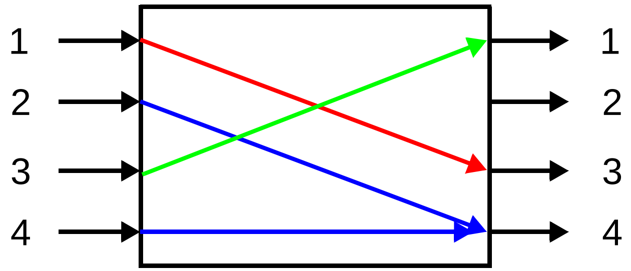
Analysis requires simulation, analytical calculations (and small demonstrators):

This is the subject of this presentation

Definitions: *Blocking, Contention*

Switching Pattern:

a particular set of connections between input and output ports.



We denote this switching pattern by: 3 4 1 4

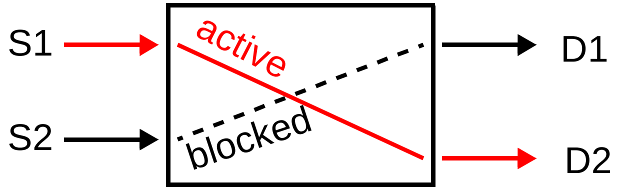
Output Contention:

when more than 1 input attempt to send data to the same output

In previous pattern 2 and 4 contend for output 4

Blocking Pattern:

a switching pattern, with no output contention, is blocking if the data cannot flow on all connections simultaneously

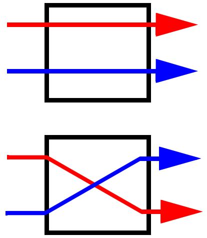


Connection S1 to D2 inhibits data transfer on S2 to D1

Definitions: *Non-Blocking and Blocking Switches*

Non-Blocking switch:

a switch is non-blocking if all output-contention free switching patterns are non-blocking.



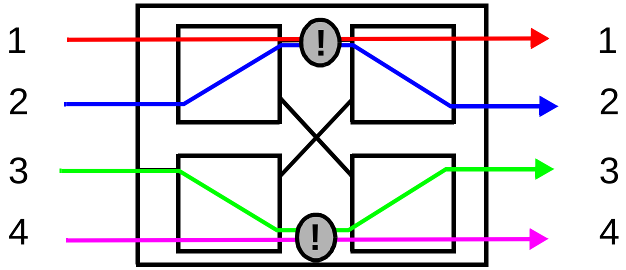
this 2 x 2 switch is non blocking if both traffics in each pattern can take place simultaneously

Blocking switch:

a switch with blocking patterns.

Blocking appears when non-blocking switches are interconnected.

It is caused by output contention within the switching fabric.



“1 2 3 4” is blocking

Number of switching patterns: N^N

Number of contention free patterns: $N!$ $\quad (\sim\ N^N \bullet e^{-N} \bullet \sqrt{2 \cdot \pi \cdot N}\ )$

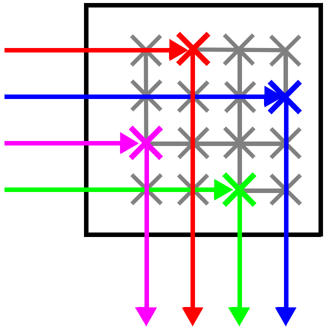
==> # contention free patterns << # of switching patterns

(e.g. if N = 100, $e^{-N} = 10^{-44}$)

Resolving Contention

a) by input queueing

Example: Crossbar switch:



N^2 cross points

~ N internal links

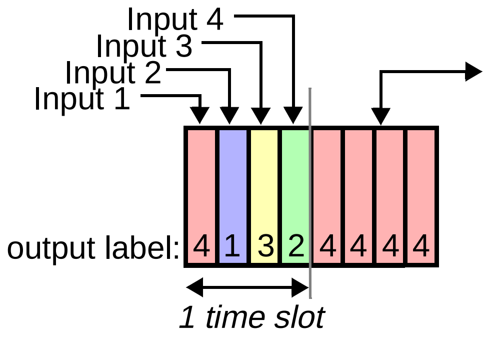
max 1 cross point enabled / column

- Aggregate internal bandwidth is N times I/O bandwidth, but each source has a reserved bandwidth, even if not used.
- In case of contention, the sources waiting for the link must store the data *==> buffer space must be provided at input (FIFO)*
- The 1st packet in line blocks the next packets even if their path is free. ==> “head of line blocking” ==> lower link bandwidth utilization
- For data frames with variable size

Resolving Contention

b) by output queueing

Example: Time division switch (shared bus):



Shared bus

Packets transmitted to output even when output contention occurs

=> output queueing

- Internal bus bandwidth: N times I/O bandwidth, shared between all inputs.
- An output port can recieve up to N packets during a time slot
  ==> *buffer space must be provided at output*
- Requires fast memory (N times faster than for equivalent crossbar switch)
- Fixed size packets only.
- No Head of Line Blocking ==> full throughput is possible
- Output buffer overflow occurs if load is not properly balanced.

Non-blocking switches are not scalable:

N^2 crossing points

or shared bus with N * link bandwidth

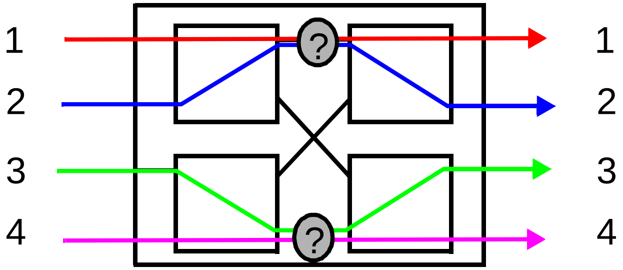
+ memory access time ÷ 1/N)

Switching Fabrics

Large switching networks can be implemented
by interconnecting non-blocking switches

But single path networks are blocking:

Example: 4X4 network based on 2X2 non-blocking switches



The 4! switching patterns that are output-contention free can be divided in:

16 non-blocking patterns:

1 3 2 4
1 3 4 2
1 4 2 3
2 3 1 4
2 3 4 1
2 4 1 3
3 1 2 4
3 1 4 2
4 1 2 3
3 2 1 4
3 2 4 1
4 2 1 3

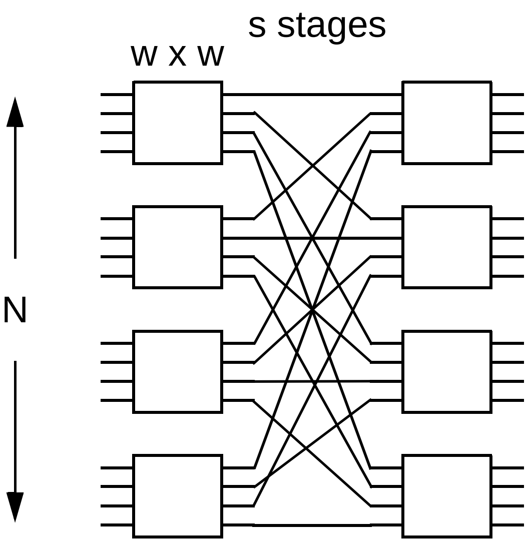
8 blocking patterns:

1 2 3 4
3 4 1 2
1 2 4 3
4 3 1 2
2 1 3 4
3 4 2 1
2 1 4 3
4 3 2 1

Switching Fabrics: *General case*

N X N switching fabric (Banyan) built from

w x w non-blocking switching elements:



# of stages (integer):	$s = \log_w N$
# of switching elements:	$s \times N / w = N (\log_n N) / w$
# of switching patterns:	N^N
# non-blocking patterns:	$(w!)^{s \cdot N / w}$

==> # blocking **>>** # non-blocking

However # non-blocking >> N

==> it is always possible to find a set of N non-blocking configurations that interconnect each input to each output exactly once

(will be used for building a barrel shifter)

Example:

w = 4,

N = 16, ==> s = 2

# elements = 8

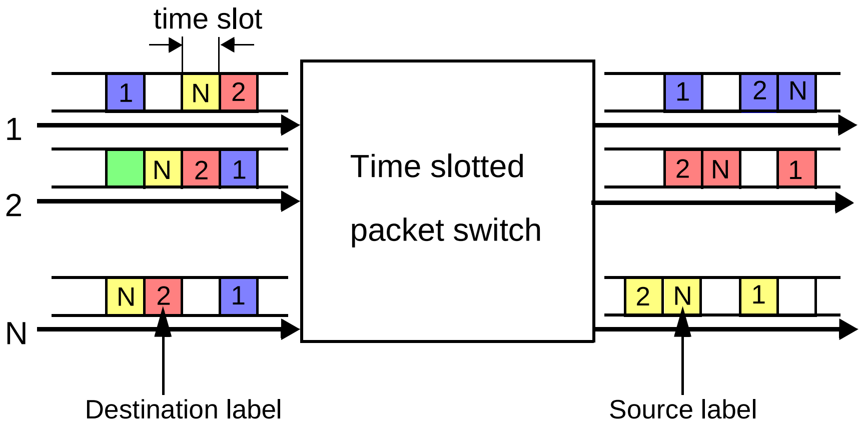
total # patterns $= 16^{16} = 1.8 \times 10^{19}$

# non-blocking patterns $= 2^{48} = 3.0 \times 10^{10}$

Simulation Model

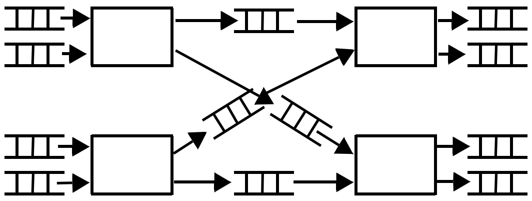
Implements:

- Non-blocking switches of any size
- Input queueing / Output queueing
- Switching fabrics ($N = w^k$) with Banyan interconnection
- Optional inter-stage buffers with limited or unlimited capacity
- Fixed / variable length packets,
- Sequential / random access of sources to the network
- Random traffic:
  - *equal probability of destinations*
  - *no correlation between consecutive destinations*
- Event building traffic
  - *sequential destination assignment*
  - *non-blocking destination assignment (barrel shifter)*



- time unit = transfer time of 1 cell
- variable size fragments = several consecutive cells to the same destination + variable inter-trigger delay)

optional “inter-stage” buffers:



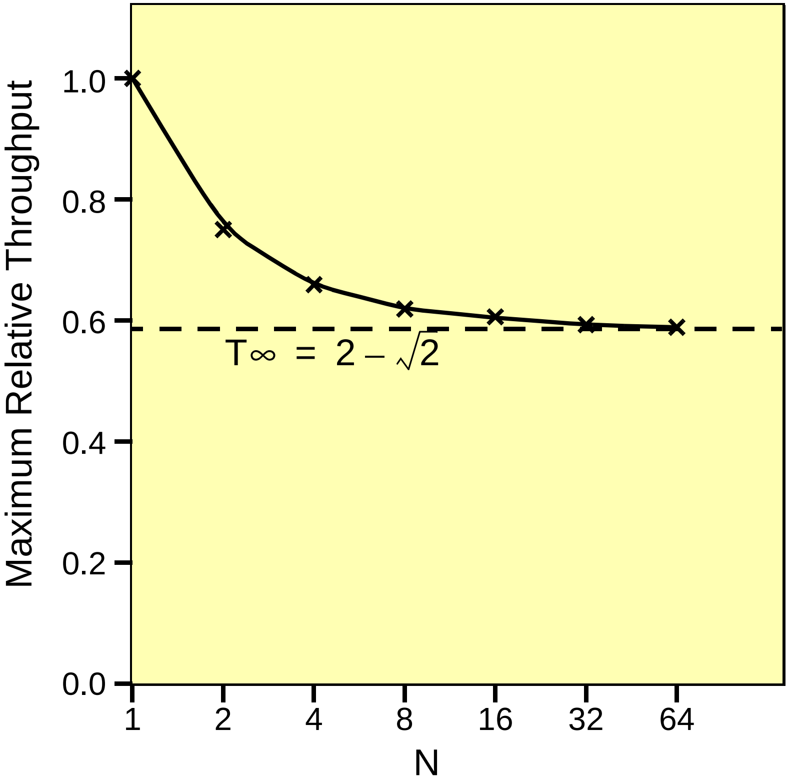
Performance of non-blocking switches

Input queueing, Random traffic

Saturation of input traffic to determine maximum possible throughput

N	[Ref 1]	Model
1	1.00	--
2	0.7500	0.7516
3	0.6825	
4	0.6553	0.659
5	0.6399	
6	0.6302	
7	0.6234	
8	0.6184	0.619
∞	0.5858	0.5887 (64x64)

Aymptotic: $T_\infty = 2 - \sqrt{2}$



Ref [1]: M.J. Karol et al., “Input versus Output Queueing on a Space-Division Packet Switch”, *IEEE Trans. on Communications*, vol. Com-35, No 12, Dec. 1987.

Performance of non-blocking switches

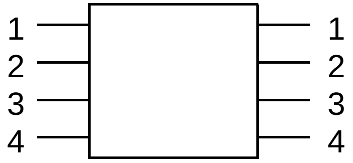
Event Building traffic: Ideal case

Assumptions:

- The sources access the network in the same order (1->N):
- All event fragments have the same size
- The input traffic is saturated
- The input buffer is not limited (no data loss at input)
- Non-blocking switch

The result is that the traffic organizes itself automatically as a “barrel shifter”

Example: 4 X 4, non-blocking switch:



Time slot:

	1	2	3	4	5
throughput:	***0.25***	***0.50***	***0.75***	***1.0***	***1.0***
input 1 ->	**1**	**2**	**3**	**4**	**1**
input 2 ->	**①**	**2 1**	**3 2**	**4 3**	**1 4**
input 3 ->	**①**	**2 ①**	**3 2 1**	**1 4 3 2**	**1 4 3**
input 4 ->	**①**	**2 ①**	**3 2 ①**	**4 3 2 1**	**1 4 3 2**

From time slot 4 (N) the throughput is maximum

Performance of non-blocking switches
Event Building traffic: Real case

Removing some of the “ideal” assumptions:

• Random order of the sources	==> still 100%
• Lower input load	==> 100% of input load
• variable size of fragments	~ random traffic throughput (eg 58% for 32 x 32)
• Introduce a perturbation (1 source at random sends to a random destination)	==> ~ 80 % (on 32 x 32)

Output queueing:

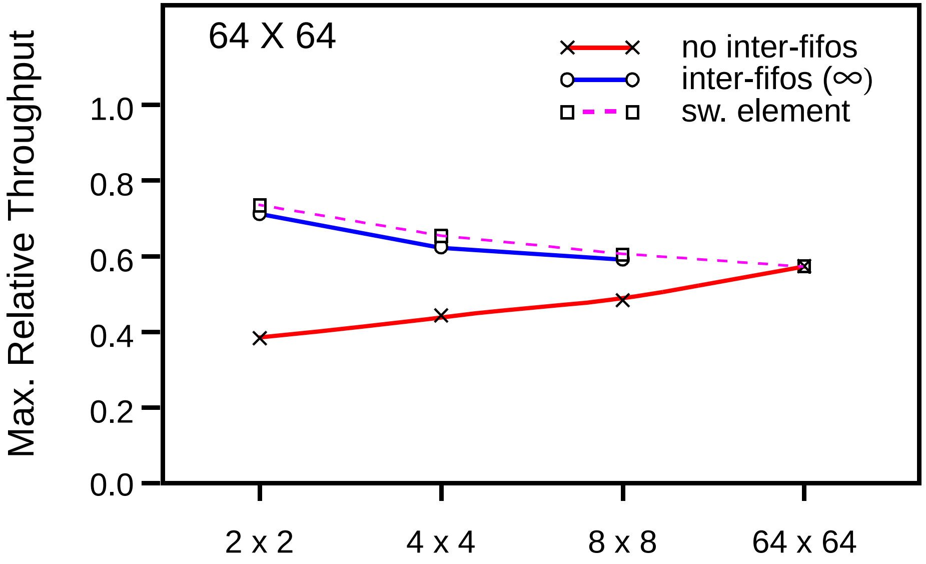
- Throughput = 100%

Performance of Switching Fabrics

A) dependence on the switching element size

Random Traffic, Input Queueing:

- For fixed size (N x N) switching fabric, analyze the throughput as a function of the switching element size (w x w)
- Influence of inter-stage buffers



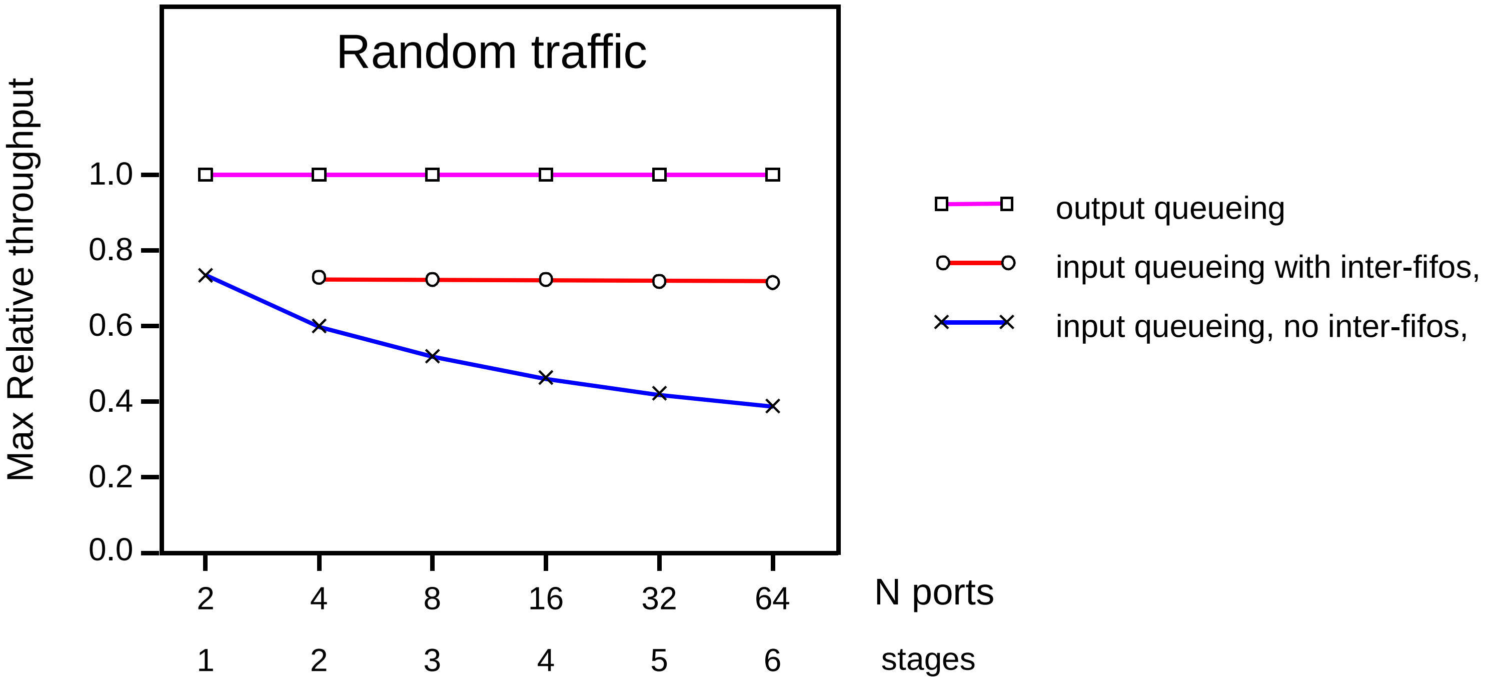
- No inter-stage fifos
  => choose largest elements
- with inter-stage fifos
  => choose smallest elements

Inter-stage fifos restore the throughput of individual switching elements

Performance of Switching Fabrics

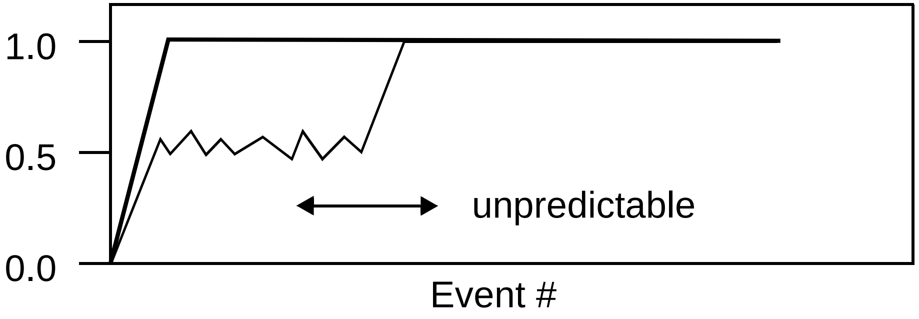
B) Scalability

2 x 2 switching elements



Event Building: *Fixed size event fragments*

- event building of fixed size event fragment on *non-blocking* switches
  ==> self-organization and 100% throughput
- still true on switching fabrics with internal blocking if
  the sources gain access to the network in fixed sequential order
- If random access: sudden jump to 100% after a large amount of events
  e.g. for a 16 x 16, 2 x 2 switching elements
  after ~ 10’000 events in one case
  after ~ 45’000 events in another run (different random number sequence)



- Very large input buffers are required
- Traffic perturbations lower the max. throughput to ~ 60% (random traffic)

==> self-organization is not safe in a real system

Event Building: *Fixed size event fragments (cntd)*

- Can one gain with *intermediate buffers* ?

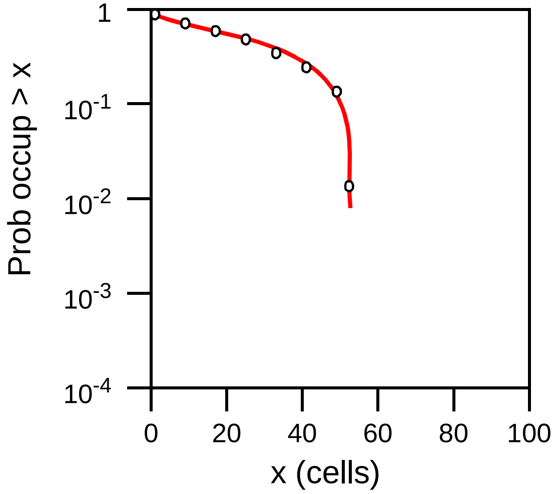
  example: 64 x 64, 2 stages 8 x 8:

no inter-buffers:	55 %
with inter-buffers:	61 %

- *Output queueing:*

  throughput can be very close to 100%

Output buffer occupancy

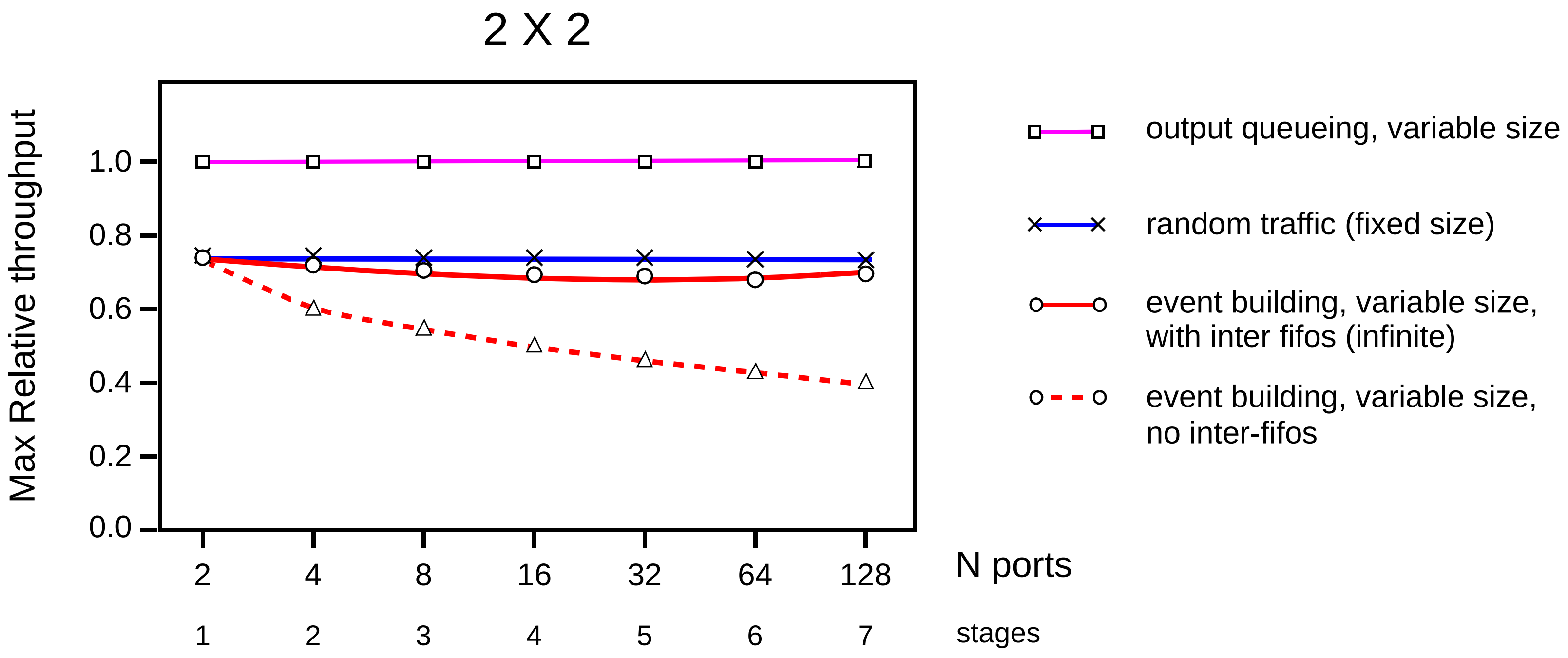


64 x 64, 2 stages 8 x 8

98 % input load

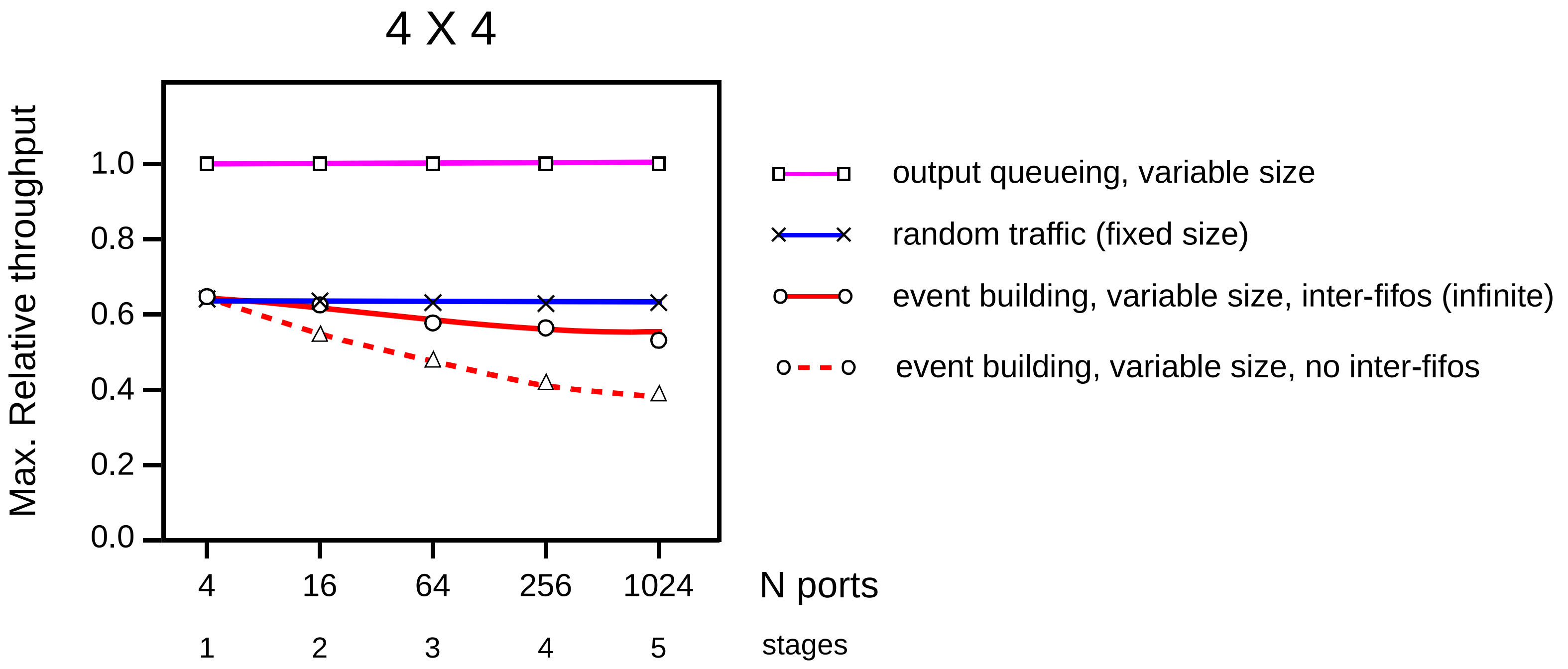
Variable size fragments:
avrg: 4 cells
max: 12 cells

Event Building: *Variable size event fragments*
Scaling with 2 X 2 sw. elements



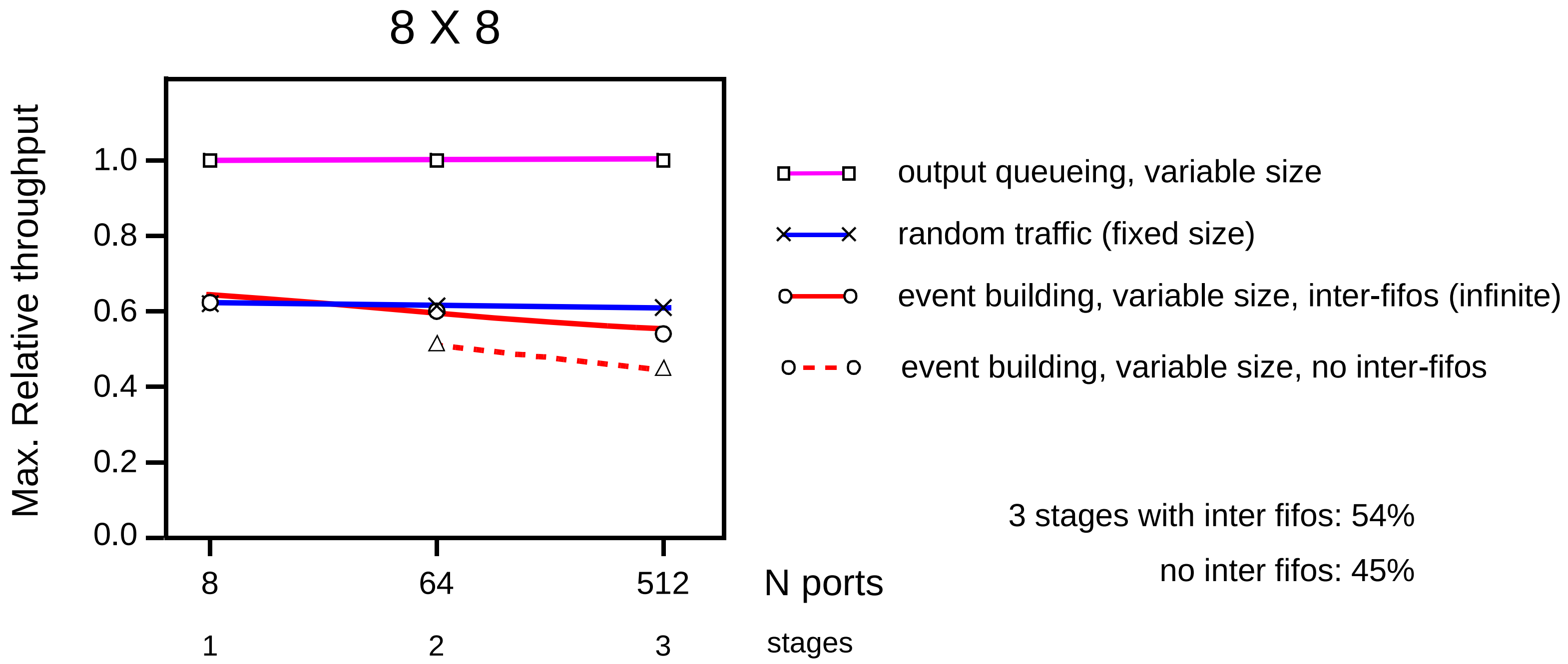
Event Building *Variable size event fragments*

Scaling with 4 X 4 sw. elements



Event Building *Variable size event fragments*

Scaling with 8 X 8 sw. elements



3 stages with inter fifos: 54%

no inter fifos: 45%

With input queueing and variable event fragment size,

the event building traffic is ~ equivalent to a random traffic

Some Standard Technologies

- **ATM**

  Output queueing (for QoS)
  Semi-permanant virtual connections -> no connection overhead
  Automatic segmentation and reassembly on top of fixed cells
  Efficient low-level transport protocol (AAL5)

- **Gigabit Ethernet**

  Can use switches with output queueing
  Connection-less
  Variable size packets, max 1.7 kB
  Complication of running without high level TP (TCP/IP)

- **Fibre Channel, class 1**

  Input queueing
  Quite long connection protocol for each transfer

- **Myrinet**

  Input queueing
  Variable packet length, no limit
  Possibility of inter-stage buffers
  Fast connection protocol

Some Standard Technologies *(Cntd)*

- **SCI**

  SCI ringlets are not equivalent to a switching network

  Max. aggregate throughput on a ringlet ~ 1.5 - 2 times the ringlet throughput (best assumption).

  To scale to higher aggregate throughput a switching network is required to interconnect the ringlets.

  Presently switches to interconnect 4 ringlets are available.

- **Others**

  Many simple crossbar switches with input queueing are available.

  Cheap but require the implementation of the I/O links.

  Require barrel shifter organization for high and predictable throughput

Conclusion

- Input queuing limits the throughput to ~ 40% - 60%
- Switching fabrics scale linearly provided that inter-stage buffers are implemented.
- Event building traffic with fragments of variable size is roughly equivalent to random traffic.
- Output queueing offers the best characteristics in terms of throughput that can approach 100% without congestion.