Switch Performance

A general view based on a simple model

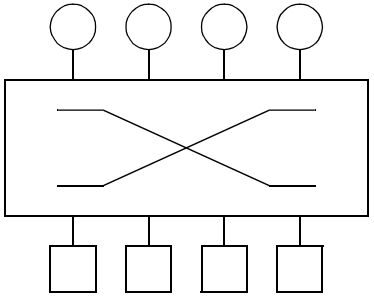
J-P Dufey, CERN

Outline:

- Overview and definitions
- (Non) Blocking switches
- Input vs Output Queueing
- Simulation model
- Performance of the various architectures
- Conclusions

Why Switching Networks?

Switching network

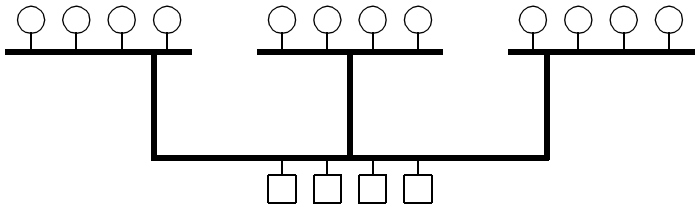


- There is at least 1 data path between any pair source -> destination
- Data transmission can occur between upto N pairs simultaneously
- Aggregate throughput at any time is

$$T \leq N \times t$$

(t = link bandwidth)

Tree structured Bus based network



- Only 1 transmission at a time
- Aggregate throughput

$$T \leq B$$

(B = bus bandwidth)

Factors that determine the Performance of a Switching Network

1) Performance of point to point links

a) Nominal 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. (But Main points recalled on next slide.

2) Performance of the switching network

Interaction between channels simultaneously active (blocking, contention)

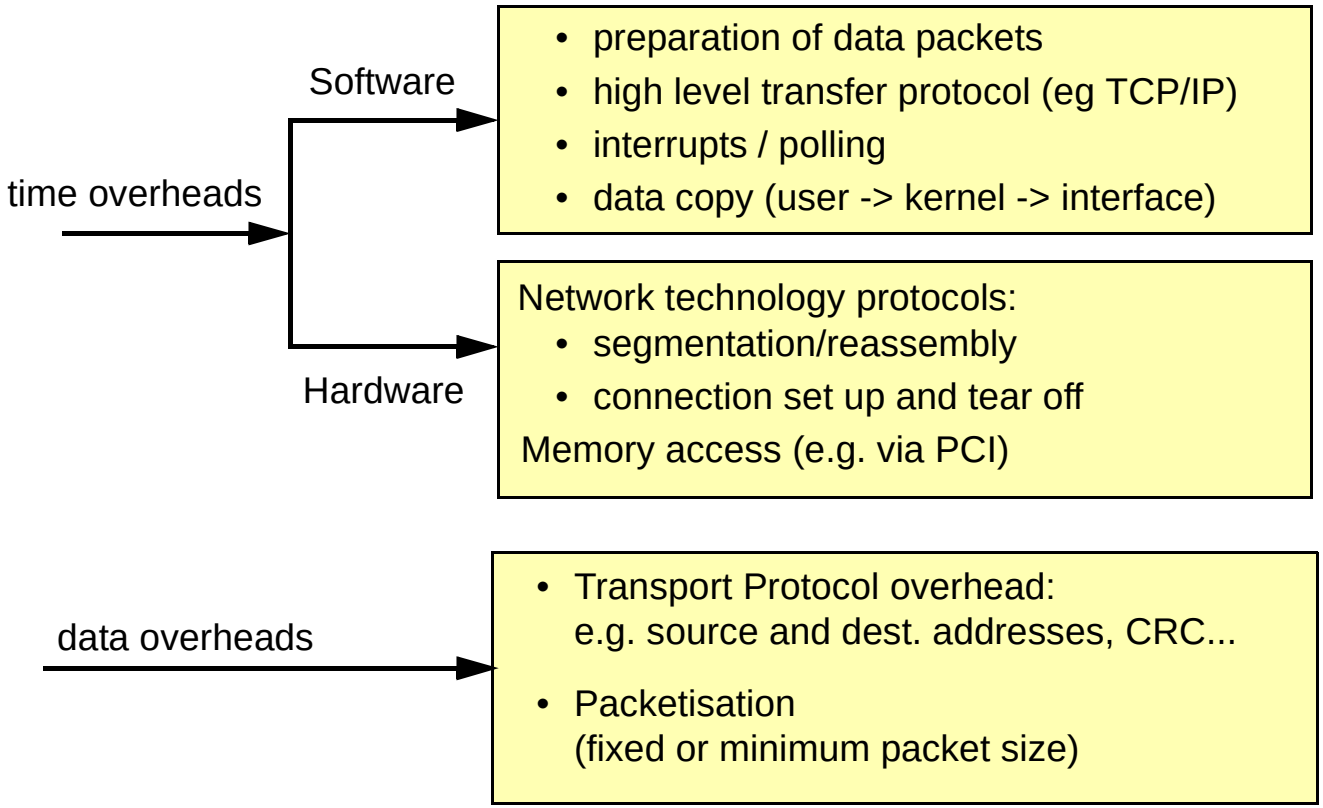
Depends on:

- 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

A Reminder on Overheads in sources and destinations

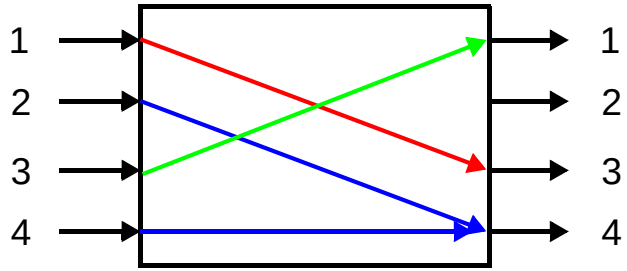


Overheads can severely limit the effective bandwidth

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

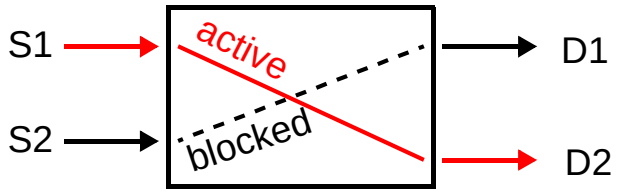
Output Contention:

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

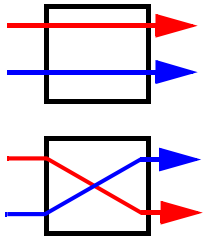


Connection S1 to D2 inhibits data transfer on S2 to D1

Definitions: *Non-Blocking and Blocking Switches*

Non-Blocking switch:

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

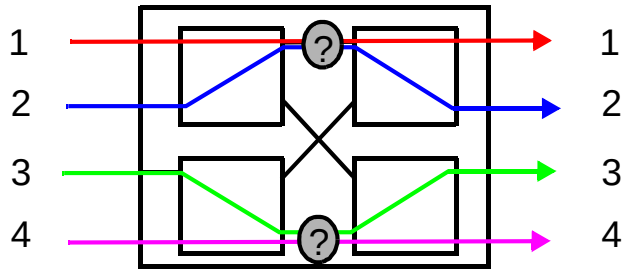


this 2 x 2 switch is non blocking if both traffics in each pattern can take place simultaneously

Blocking switch

Blocking appears when non-blocking switches are interconnected.

It is caused by output contention within the switching fabric.



"1 2 3 4" is blocking (Internal blocking)

Number of switching patterns: N^N

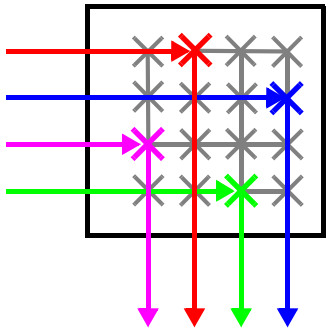
Number of contention free patterns: $N!$ $\left(\sim N^N \bullet e^{-N} \bullet \sqrt{2 \cdot \pi \cdot N}\right)$

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

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

Contention Control
1st technology: input queueing

:



N^2 cross points

~ N internal links

max 1 cross point enabled / column

Crossbar switch:

Can handle data frames of any length, independ-entely of the traffic on the other connections

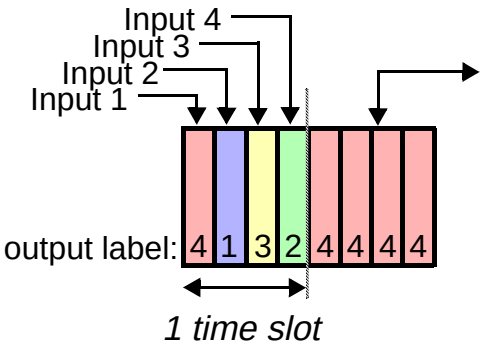
Contention: wait for vertical line free

Data must wait at input

- Aggregate internal bandwidth is N times I/O bandwidth, but each source has a reserved bandwidth, even if not used.
- Only 1 source can access an output port at a time.
- 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

Contention Control

2nd technology: output queueing



Shared bus

Packets transmitted to output even when output contention occurs

=> output queueing

Time division switch (shared bus):

Only for fixed size data packets.

Requires fast memory (N times faster than for an equivalent crossbar switch)

Data to be stored at output

- Internal link 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*
- Sources can only submit packets of fixed size.
- No Head of Line Blocking ==> full throughput is possible
- Output buffer overflow occurs if load not properly balanced.

Non-blocking switches are not scalable:

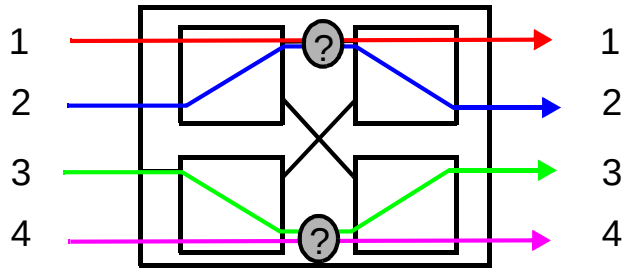
N^2 crossing points or shared bus + memory access time ÷ 1/N)

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

8 blocking patterns:

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

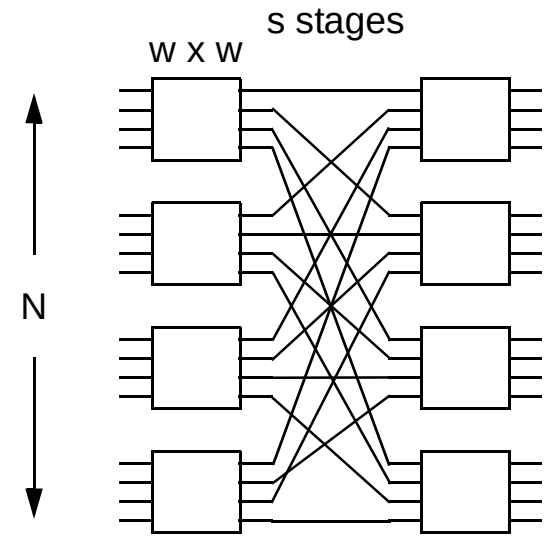
non-blocking patterns: $(w!)^{s \cdot N/w}$

==> # 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

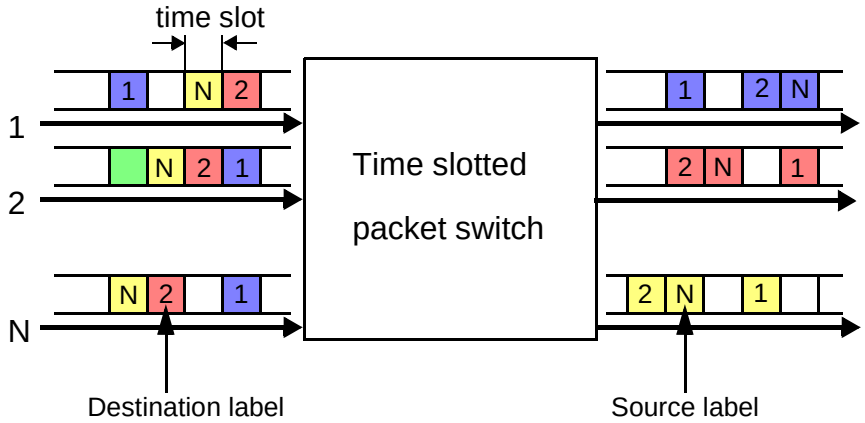
total # patterns = 16^{16} = 1.8×10^{19}

non-blocking patterns = 2^{48} = 3.0×10^{10}

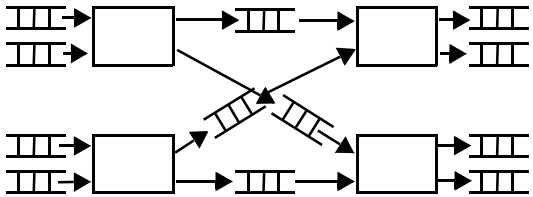
Simulation Model

Implements:

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



- 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:.



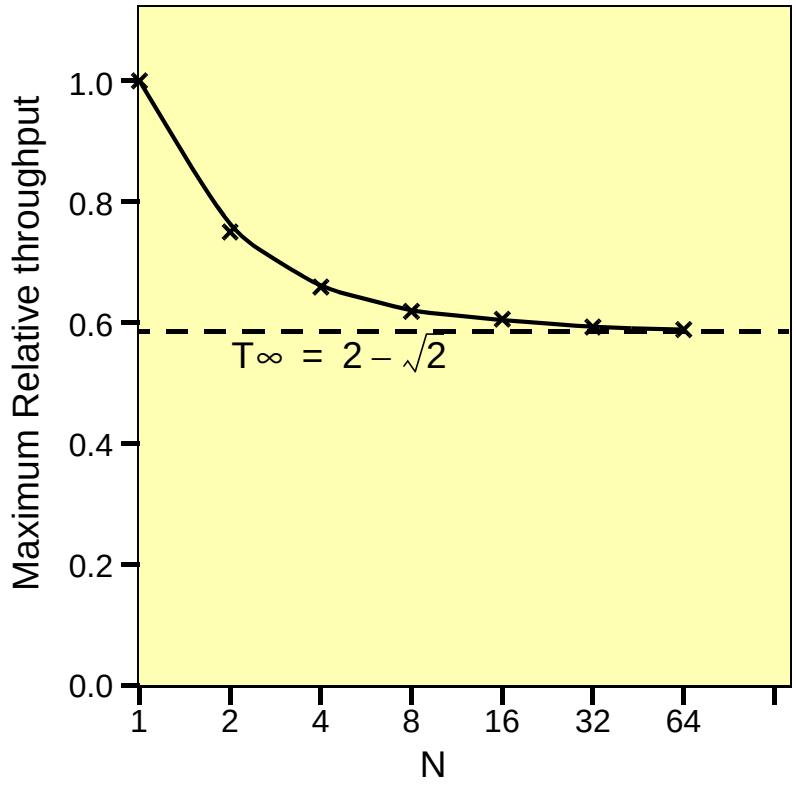
Performance of non-blocking switches

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

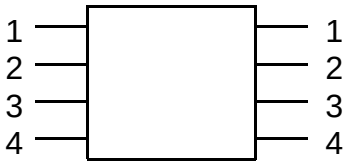
Event Building traffic: Ideal case

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:

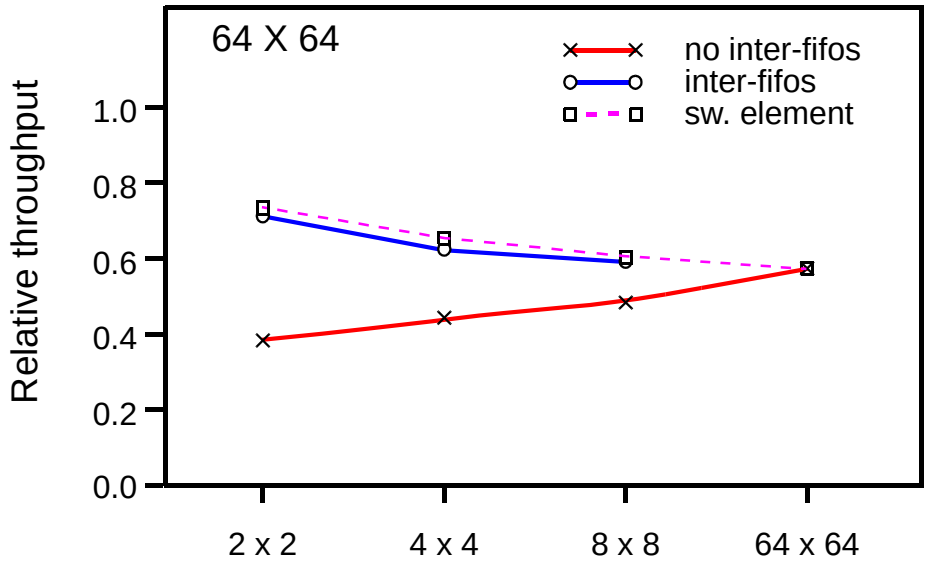
- Throughput = 100%

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



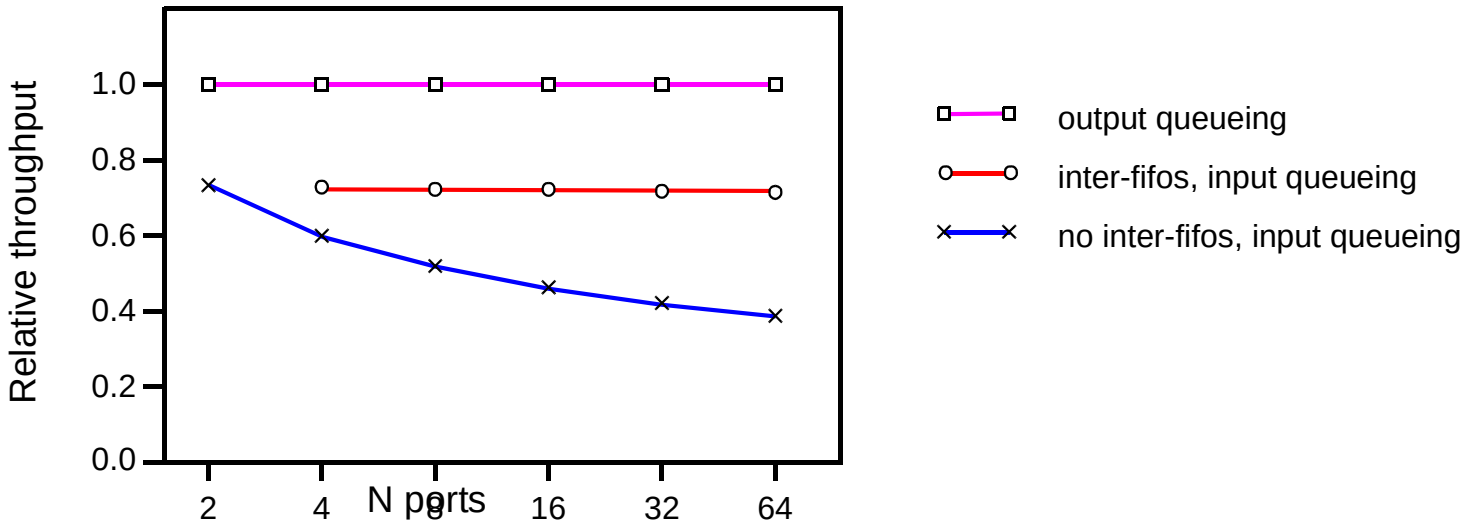
- No inter-stage fifos
  => choose largest elements
- with inter-stage fifos
  => choose smallest elements

Inter-stage fifos restore the throughput of individual switching elements

Performance of Switching Fabrics

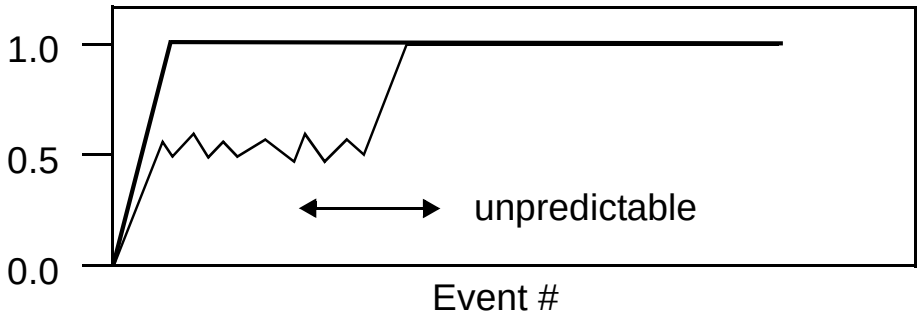
B) Scalability

2 x 2 switching elements
Random traffic



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
- Introducing a perturbation lowers the max. throughput to ~ 60% (random traffic)

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

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

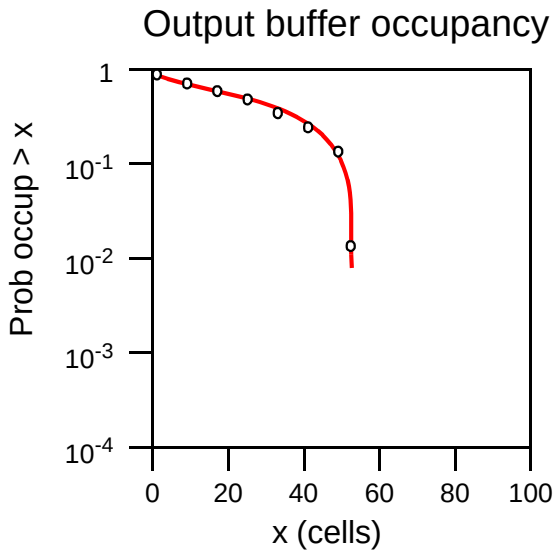
- Can one gain with *intermediate buffers* ?

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

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

- *Output queueing:*

  throughput can be very close to 100%



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 (cntd)*

Input queueing:

Example: 64 X 64, 2 stages of 8 X 8

Traffic:	Event Building	Random
no inter-buffers:	51 %	50 %
with inter-buffers:	59 %	61 %

With input queueing and variable event fragment size,

the event building traffic is equivalent to a random traffic

Output queueing:

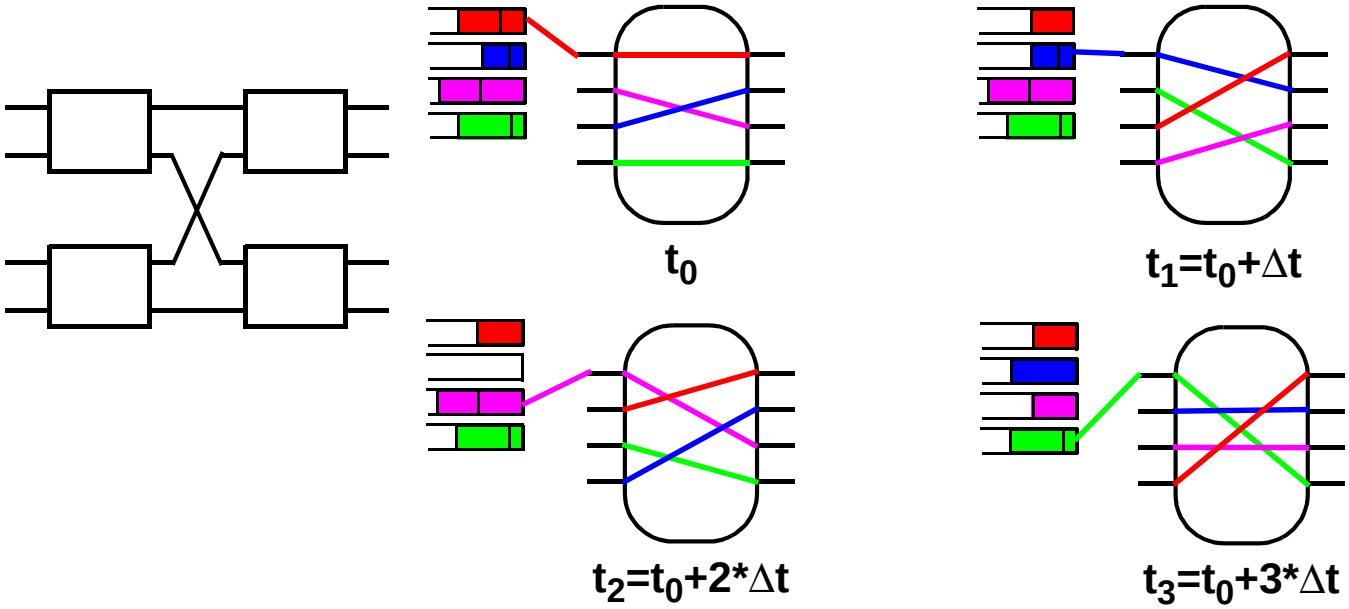
Throughput of ~ 100 % is possible, also with fragments of variable size

Barrel Shifter

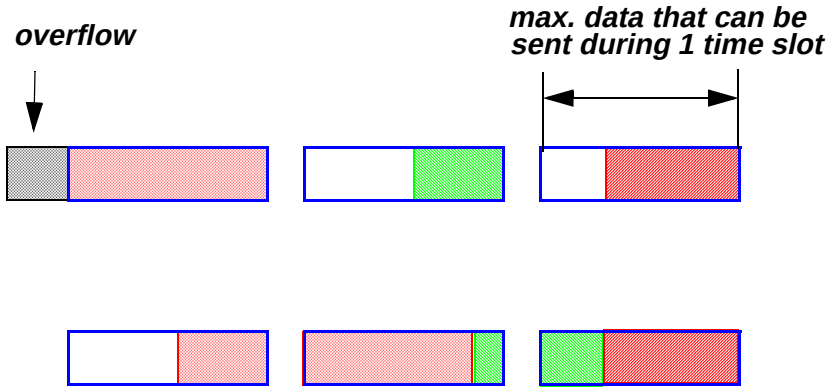
Motivation: *Reach 100 % throughput with input queueing by fixed size packets and non-blocking use of the switch*

Principle:

- N X N Blocking switch (useful also for non-blocking switch)
- Set of N non-blocking switching patterns such that each pair I_k -> O_j is connected exactly once



Barrel Shifter: Segmentation and Reassembly *?*



Inefficient use of the bandwidth

Loss of data by overflow

Use of full bandwidth is possible
by segementing the event fragments
in the sources and reassembling them
in the destinations

Segmentation and Reassembly (SAR) is a simple problem of software

Some technologies implement SAR by hardware

Analysis of the Barrel Shifter

An analysis of barrel shifter has been done by M. Nomachi (1993),

A verification by simulation has been done by Nagasaka et al. (1996),

I. Mandjavidze has proposed a "true barrel shifter" in 1994,

D. Calvet et al. have also analysed the barrel shifter type traffic shaping with applications in ATM

Ref [2]: M. Nomachi, "Event Builder Queue Occupancy", SDC-93-566, Aug. 1993

Ref [3]: Y. Nagasaka et al., "Performance Analysis of a Switch-type Event Builder with Global Traffic Control System", *IEEE Trans. on Nucl. Science*, vol 43, Feb. 1996 (RT95 issue)

Ref [4]: I. Mandjavidze, "A New Traffic Shaping Scheme: the True Barrel-Shifter", RD-31 Note 94-03.

Ref [5]: D. Calvet et al., "Evaluation of a Congestion Avoidance Scheme and Implementation on ATM Network based Event Builders", Proc. of the 2nd Intnl. Data Acquisition Workshop (DAQ96), RCNP, Osaka, Nov. 1996

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 2 x 2 switches are available. Input queueing.

- **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.
- Perturbations (essentially variable event fragment size) disturb a self-organization and lower the throughput
- Forcing a coherent organization of the event builder (barrel shifter) and allowing for SAR leads to:
  - *high throughput and, consequently, reduced numbers of sources and destinations,*
  - *low buffer requirement in sources and destinations,*
  - *No need for inter-stage buffers in the switching fabric,*
  - *"more deterministic" Event Building Latency (not shown here).*
- Output queueing offers the best characteristics in terms of throughput that can approach 100%.