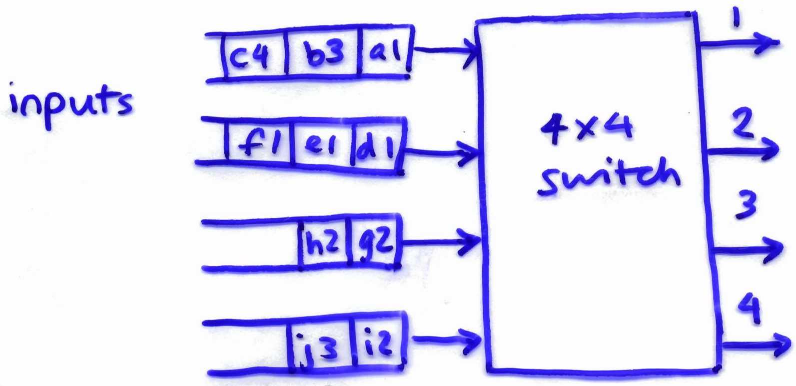
- <u>queueing & scheduling</u>
  - consider a cell switch with N inputs & N outputs with identical link (line) rates
  - assume cell arrivals occur at the beginning of a slot & cell departures are completed by the end of the slot
  - a switch can be <u>input-queued</u> (IQ), <u>output-queued</u> (OQ) or <u>combined input-output-queued</u> (CIOQ)
  - let p be the maximum no. of packets that can be switched by the fabric in 1 slot, & let q be the maximum no. of packets that can be switched to an output in 1 slot
    - if $p < N$, input queueing is needed
    - if $q > 1$, output queueing is needed
  - we can achieve $p, q > 1$ by operating the switching fabric at a rate higher than the line rate, i.e. a <u>speedup</u> of the fabric (time-based), or parallelism in the fabric (space-based), or a combination of both
  - the placement of the queues, the buffer sizes of the queues, & the switch fabric & memory speeds are important design parameters

- <u>OQ switch</u>
  - should operate at N times the line rate & have an output queue memory transfer rate of N+1 times the line rate (in the worst case, N cells are transferred from the N inputs to an output & one of them is transmitted on the output link in 1 slot)
  - work-conserving (no output link is idle if there is a cell meant for that output in the switch), easy to do QoS scheduling, eg WFQ
- <u>IQ switch</u>
  - if there is contention for some output, only 1 cell is switched to that output in the slot
  - can be operated at the line rate (ie. each input sends at most 1 cell & each output receives at most 1 cell in a slot) with a memory transfer rate twice the line rate (ie. 1 cell can be written to & read from each queue in a slot)
  - if each input has a FIFO queue, <u>head-of-line</u> (HOL) blocking occurs if more than 1 HOL cell from the input queues has the same destination — so all but 1 of them remain in their queues — and one of these queues contains a non-HOL cell whose

destination is free — this cell is not switched and its destination is idle during the slot

→ the IQ switch with FIFO queueing is not work-conserving

Eg.

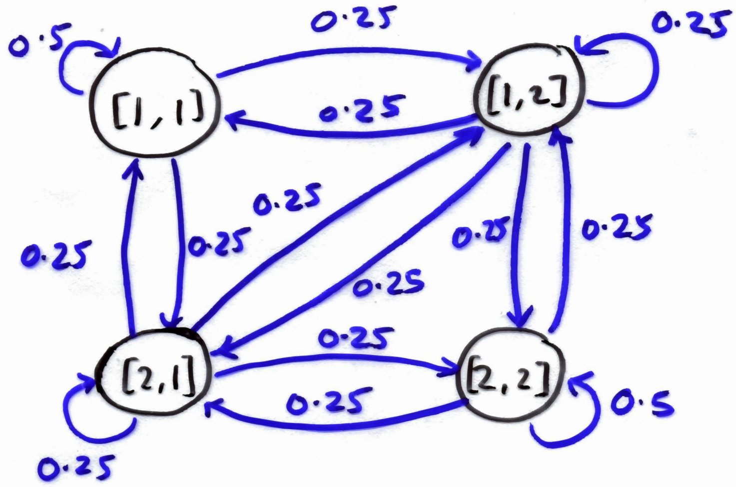


timeslot	6	5	4	3	2	1
1	x	x	f	a	e	d
2	x	x	x	i	h	g
3	x	b	j	x	x	x
4	c	x	x	x	x	x

(input cells are labelled with a letter ID & a no. indicating the destination port of the cell)

(output cell departures in each timeslot — 'x' indicates no transmission)

- the loss in throughput relative to the OQ switch tends to grow with switch size
- consider the case of saturated inputs for an N×N IQ switch with FIFO queueing
- assume each cell's destination is chosen independently & uniformly at random over the N outputs, & that if there is output contention, one of the cells is chosen randomly to be switched
- consider the case $N=2$, & let $d_n^{(i)}$ denote the destination of input i's HOL cell in slot n
- $d_n \triangleq [d_n^{(1)}\ d_n^{(2)}]$ evolves according to a Markov chain



- the stationary probability of the switch being in each of the 4 states is 0.25
- if $d_n = [1,1]$ or $[2,2]$, only 1 cell is switched in slot n, & the throughput is 0.5 cells per port; if $d_n = [1,2]$ or $[2,1]$, both cells are switched in slot n, & the throughput is 1 cell per port

→ the stationary saturation throughput of the switch is thus 0.75 cells per port

- for $N \to \infty$, the saturation throughput can be shown to converge to $2 - \sqrt{2} \approx 0.586$ cells per port

• the IQ switch with FIFO queueing operated at line rate & the OQ switch are at 2 ends of a tradeoff between throughput & resources/complexity

• the line-rate IQ switch can be improved w/ non-FIFO queueing, e.g.
  - windowing: the scheduler can select 1 of the first w cells from each input queue (↓s HOL blocking)
  - virtual output queueing: each input maintains a separate queue for each output (eliminates HOL blocking) & the scheduler chooses a matching

between inputs & outputs (because of line-rate switching constraint)

- ideal throughput (i.e. queues are stable for any feasible arrival rates where the desired rate for each input & output is $\leq$ 1 cell / timeslot) is achieved with maximum weight matching, where the weight w_{ij} for the edge from input i to output j is given by either
  - the length of the queue Q_{ij} for output j at input i, or
  - the no. of slots that the oldest cell in Q_{ij} has waited
- practical switches use suboptimal matching algorithms which are less complex

- CIOQ switch

- up to L cells, $1 < L < N$, can be switched to an output in 1 slot
- saturation throughput for $N \to \infty$ can be very close to 1 even for small L (exact performance depends on details of the input scheduler, etc)
- a CIOQ switch can be used to emulate an OQ switch with QoS scheduling, but this utilizes a relatively complex centralized scheduler