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

Manolis Katevenis

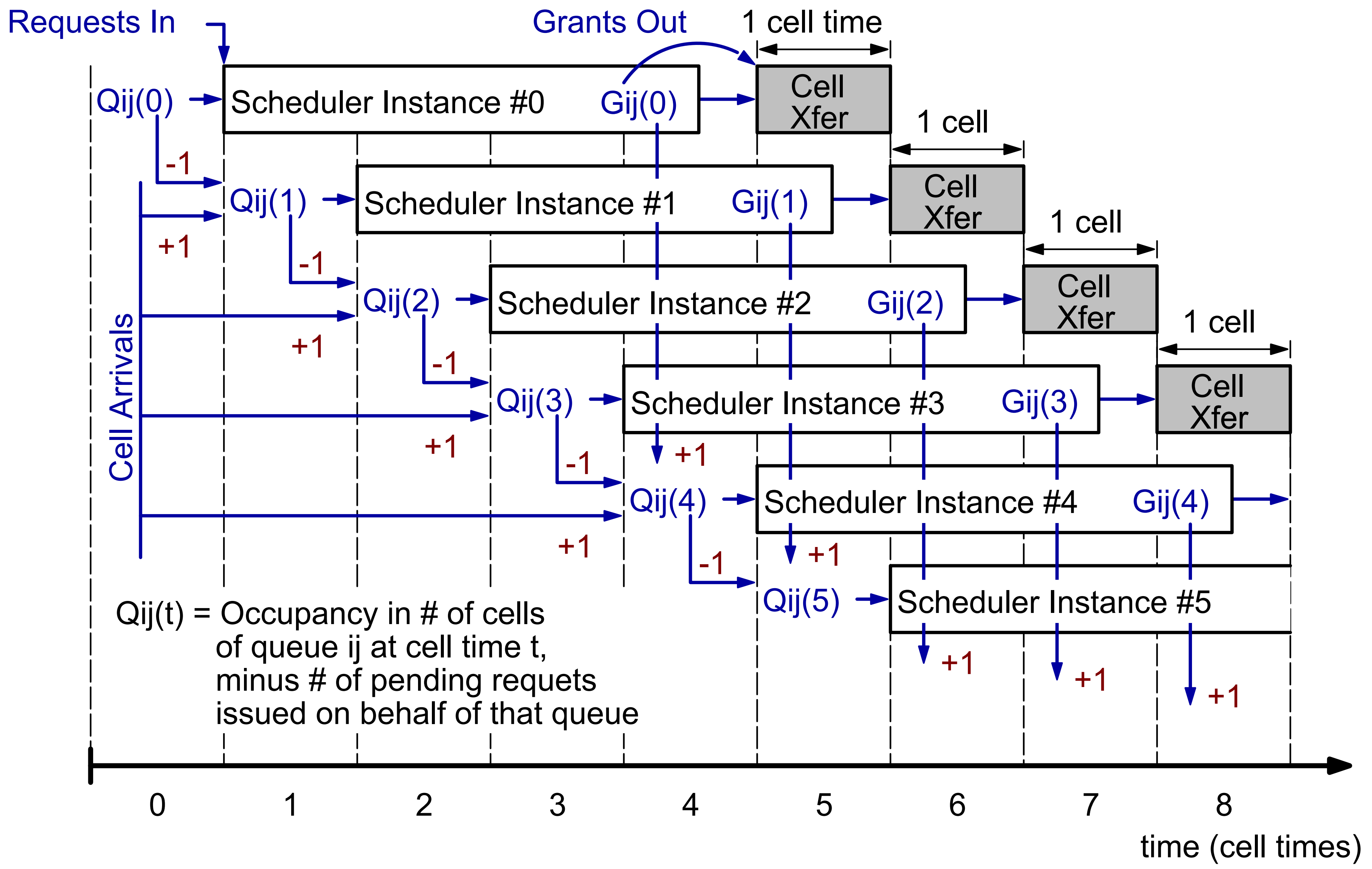
CS-534 – Univ. of Crete and FORTH, Greece

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

4.3 Advanced Topics in IQ’ing & Crossbar Scheduling

Table of Contents:

4.3.1 Pipelined Scheduling for Cell-based Crossbars



Qij(t) = Occupancy in # of cells of queue ij at cell time t, minus # of pending requets issued on behalf of that queue

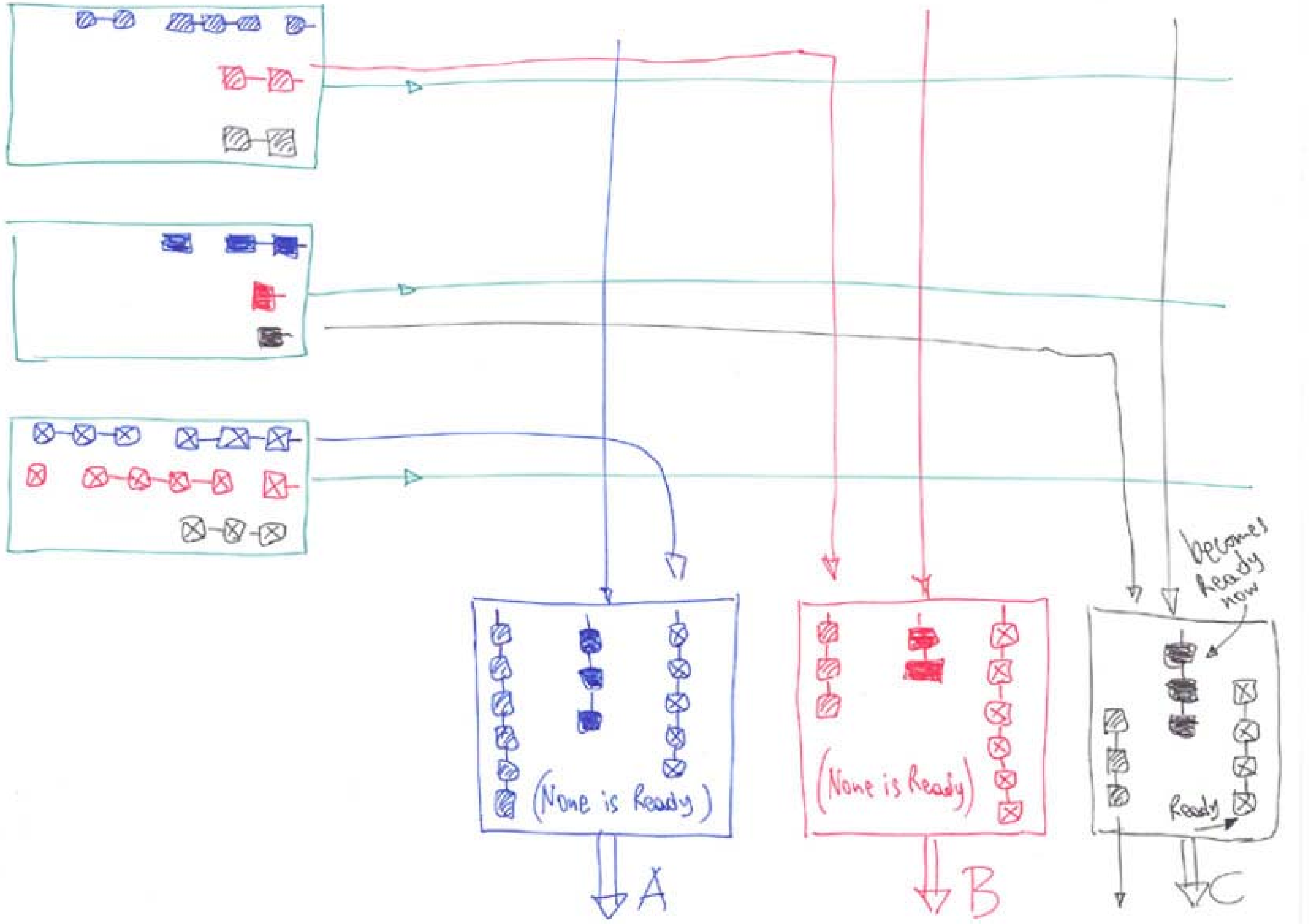
- Cell arrivals increment Qij (Qij = “guaranteed” min. occupancy of Q. i,j)
- When Qij ≥ 1 $\Rightarrow$ issue a request Rij
- Requests issued decrement Qij, pending the transfer decision
- When a pending decision is resolved:
  - successful grants leave Qij as is (already decremented)
  - missed grants re-increment Qij (restore rejected request)
- Prerequisite: *fixed-size cells – interchangeable requests*
  - requests for cells and actual cell transfers are interchangeable with each other: if the “first” request, on behalf of the “first” cell in the queue, is not granted, then the “second” request –that was issued on behalf of the “second” cell– if granted, will result in transfering the “first” cell.
- Reference: Oki, Rojas-Cessa, Chao: “A pipeline-based approach for maximal-sized matching scheduling in input-buffered switches”, IEEE Communications Letters, June 2001

4.3.2 Variable-Size Packets in Fixed-Size-Cell Crossbars

- Crossbars operate on a fixed-cell-time periodicity
  - need periodic time intervals at which to make scheduling decisions
  - need to reconfigure all input-output pairs at once
    (but see about "packet mode" scheduling, below)
  - we refer to "unbuffered" crossbars here – "buffered" crossbars, which contain small buffers at their crosspoints, can operate asynchronously (to be discussed later)
- Variable-size packets can be switched only after segmentation into fixed-size cells
  - Segmentation Overhead: partially filled last cell of packet
  - Reassemble packets after the crossbar – buffers & queues needed
  - Must store-and-forward – no cut-through possible any more

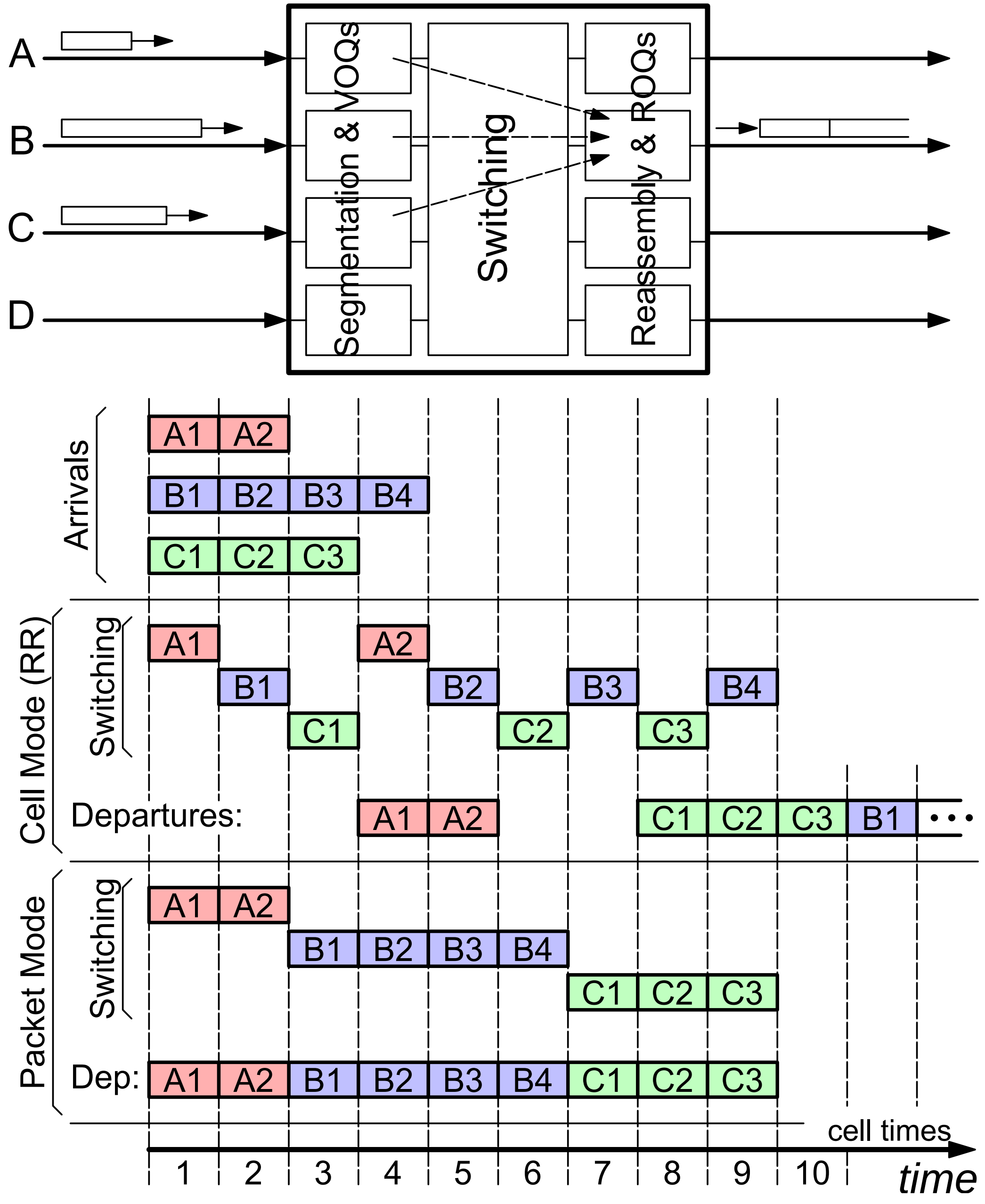
Segmentation and Reassembly (SAR): Reassembly Queues and Buffer Space needed

- Reassembly queues per-input needed at each output
- Reassembly space per output, when this space is shared among all reassembly queues at that output:
  - a max-size packet per input may be almost complete at the output
  - after the first completed packet starts being forwarded to the output, outgoing rate ≈ incoming rate ⇒ fixed total buffer occupancy
- Reassembly space, when partitioned per reassembly queue:
  - much more than with shared space:
  - after a first max-size packet, A, gets completed, and while it is being forwarded to the output (time ~ max.pck.size), one of the other reassembly queues (already containing almost one max. packet) may be receiving cells of total size ~ 1 max. packet, and so on...


(None is Ready)
(None is Ready)
becomes ready now
Ready
A
B
C

4.3.2b Packet-Mode Scheduling

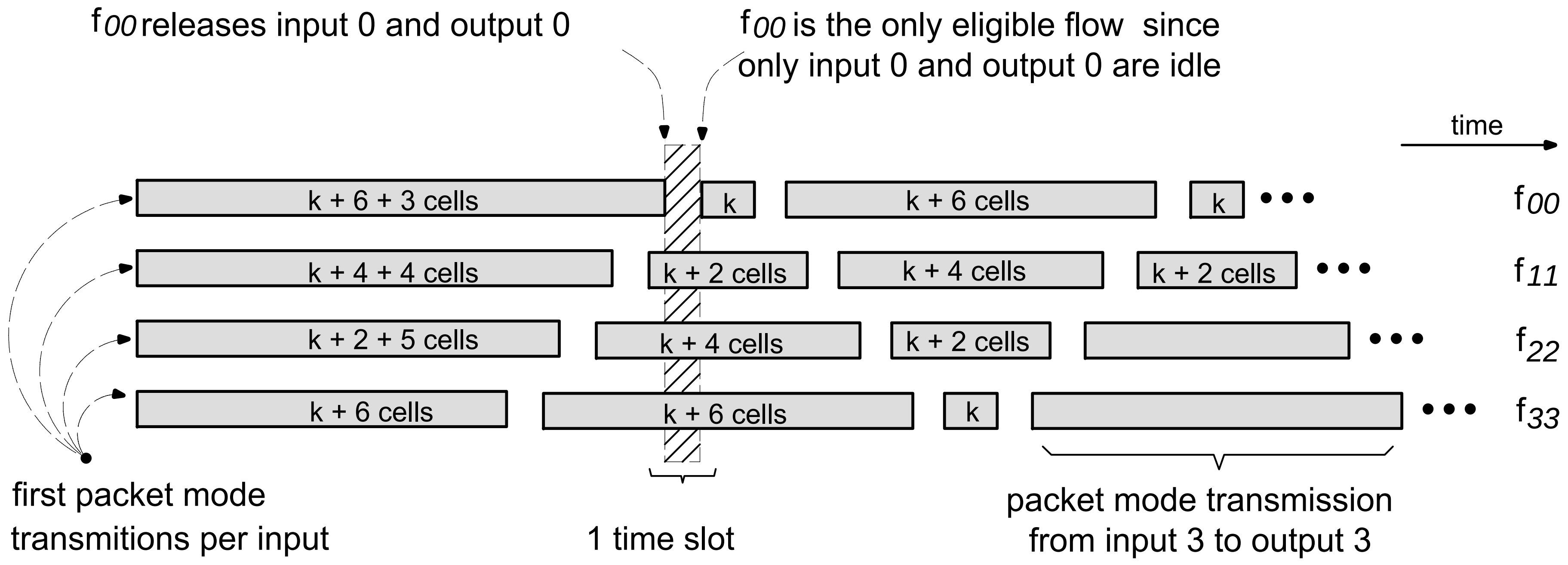
- Variable size packets segmented into fixed-size cells for VOQ and crossbar operation
- If cells of a packet are forwarded through the xbar non-consecutively:
  - increased delay
  - reassembly queues
  - no cut-through forward
- *Packet-mode* scheduling: forward all cells of a packet in consecutive cell-times



Packet-Mode Scheduling, Asynchronous Operation

- *Packet-Mode Scheduling:*
  - Maintain input-output pairings until full packet is forwarded;
  - Scheduler only considers candidacies for those inputs and outputs that are not currently busy in the "middle" of a packet;
  - Reduces delay, eliminates reassembly, allows cut-through;
  - Rare danger of pathological "locking": when a packet-mode "connection" completes, if all other ports are still busy and that flow's VOQ is non-empty, the same connection is made again.
  - Reference: Ajmone Marsan e.a.: IEEE/ACM ToN, Oct. 2002
- *Asynchronous Crossbar Operation:*
  - Schedule ports whenever freed – not necessarily in synchrony
  - Similar to packet-mode, but sched'ngTime >> pck-sz granularity
  - Reference: Passas, Katevenis: HPSR 2007

Packet-Mode Scheduling: Danger of Pathological Locking in one Configuration



f_{00} releases input 0 and output 0
f_{00} is the only eligible flow since only input 0 and output 0 are idle
time
k + 6 + 3 cells
k
k + 6 cells
k
f_{00}
k + 4 + 4 cells
k + 2 cells
k + 4 cells
k + 2 cells
f_{11}
k + 2 + 5 cells
k + 4 cells
k + 2 cells
f_{22}
k + 6 cells
k + 6 cells
k
f_{33}
first packet mode transmitions per input
1 time slot
packet mode transmission from input 3 to output 3

- during the cell-time when a packet transmission is completed at an input-output pair, if this is the only input-output pair that gets released and there are more packets in that VOQ, then the same pair will get re-connected $\Rightarrow$ other flows are locked out

4.3.3 Switches with Reconfiguration Overhead: large, multi-packet "Enveloppes" in lieu of cells

- Reconfiguration Overhead
  - crossbars without central clock ⇒ rcvrs' resynchronization delay
  - optical crossbars
- Reduce reconfiguration frequency

⇒ Increase cell time, i.e. increase cell size

- call the large cells "Enveloppes"
- Allow multiple segments from multiple packets in an enveloppe
- enveloppes are formed in the VOQ's

Switches with Reconfiguration Overhead: Enveloppe Scheduling

- When to "release" an Enveloppe to the Switch?
  - partially filled enveloppes waste switch throughput

    $\Rightarrow$ wait for enveloppes to fill up
  - but if excessive waiting $\Rightarrow$ too much delay introduced

    $\Rightarrow$ use some kind of timeout mechanism, or use a portion of the switch throughput to periodically “scan” all partially-filled VOQ's

- *Reference:* Kar e.a.: “Reduced Complexity Input Buffered Switches”, Hot Interconnects 2000.