UNIVERSITY OF CAMBRIDGE

P51: High Performance Networking

Lecture 3: Low Latency Devices

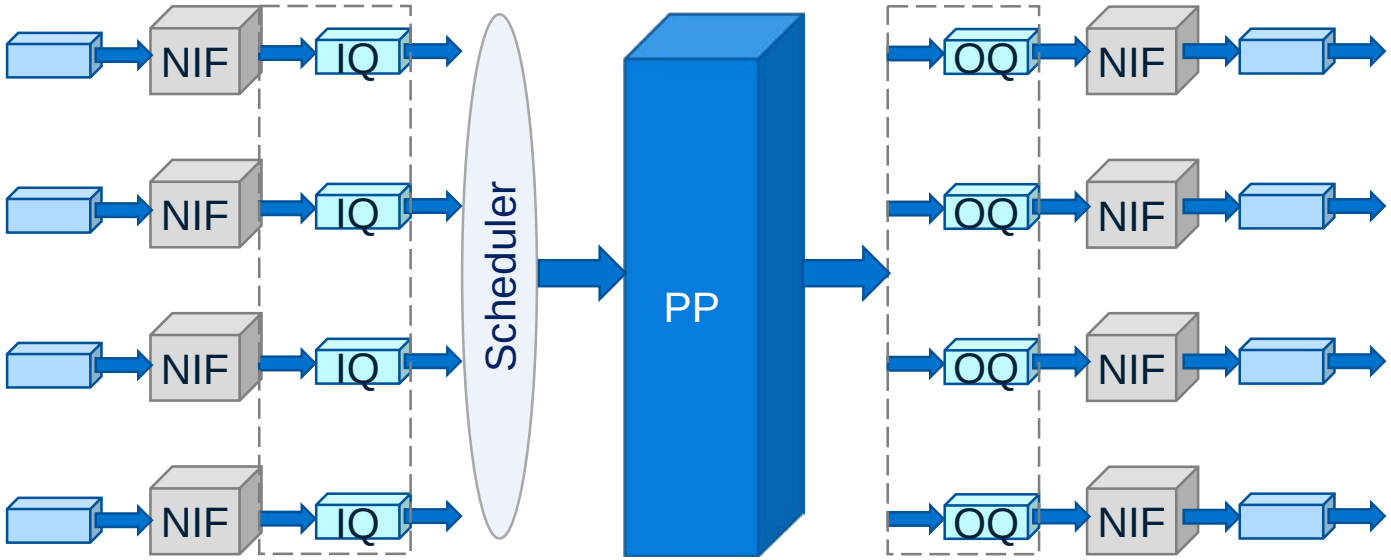
Dr Noa Zilberman
noa.zilberman@cl.cam.ac.uk

Lent 2017/18

Low Latency Switches

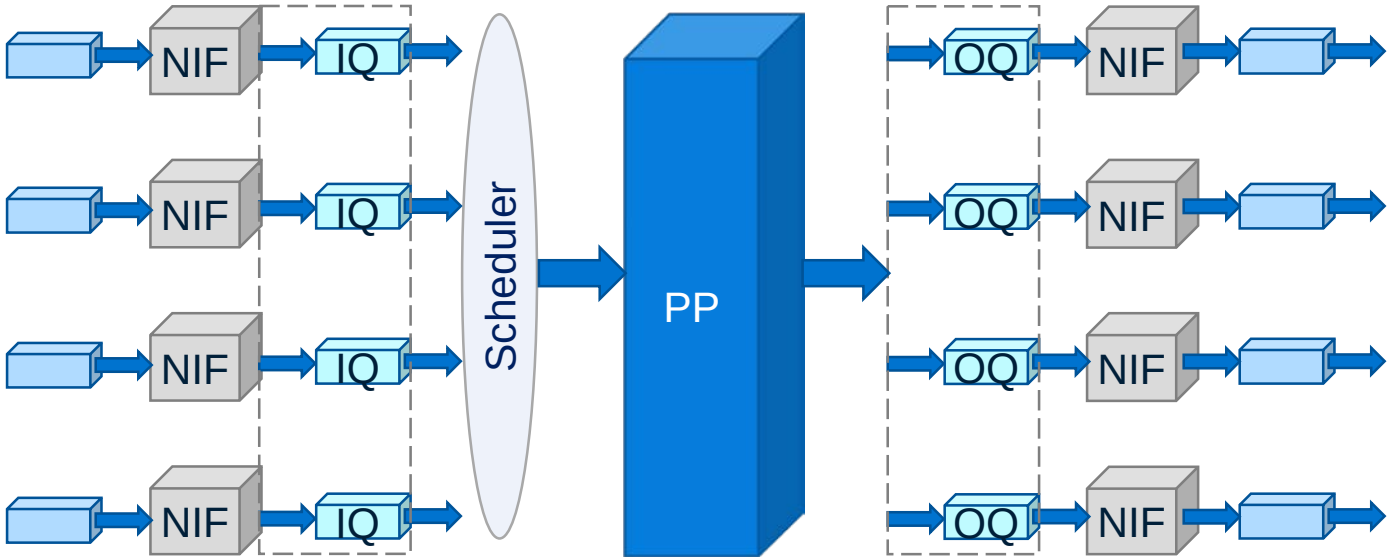
How to lower the latency of a switch?

- Obvious option 1: Increase clock frequency
  - E.g. change core clock frequency from 100MHz to 200MHz
  - Half the time through the pipeline



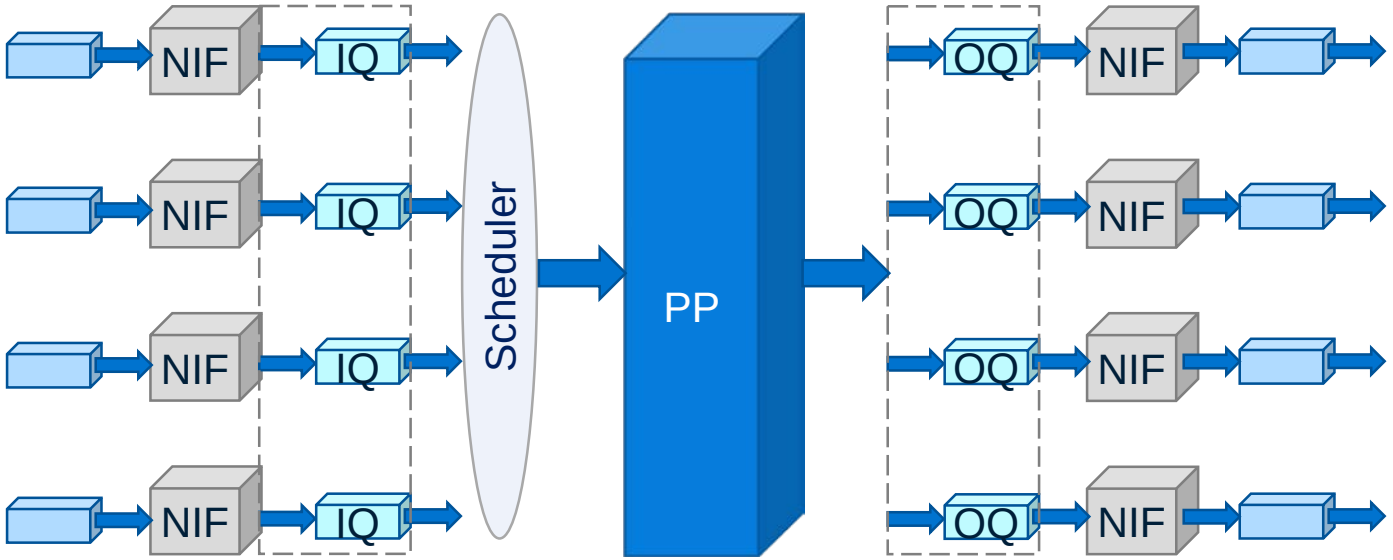
How to lower the latency of a switch?

- Obvious option 1: Increase clock frequency
- Limitations:
  - Frequency is often a property of manufacturing process
  - Some modules (e.g. PCS) must work at a specific frequency (multiplications)



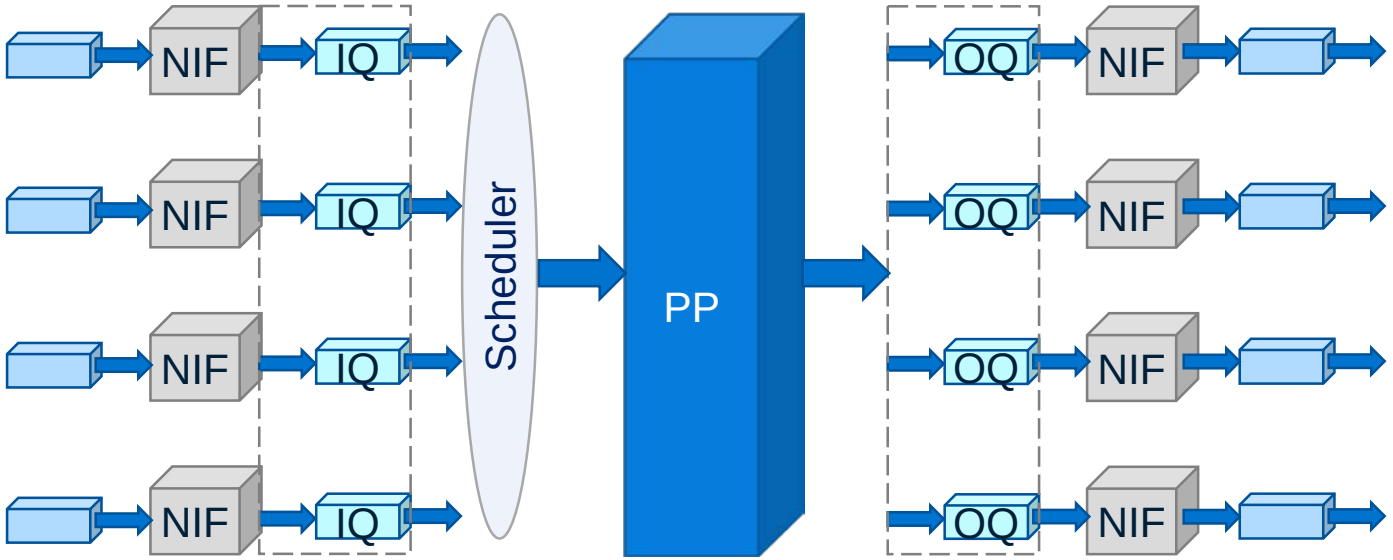
How to lower the latency of a switch?

- Obvious option 2: Reduce the number of pipeline stages
  - Can you do the same in 150 pipeline stages instead of 200?
  - Limitation: hard to achieve.



How to lower the latency of a switch?

- Can we achieve ~0 latency switch?
  - Is there a lower bound on switch latency?



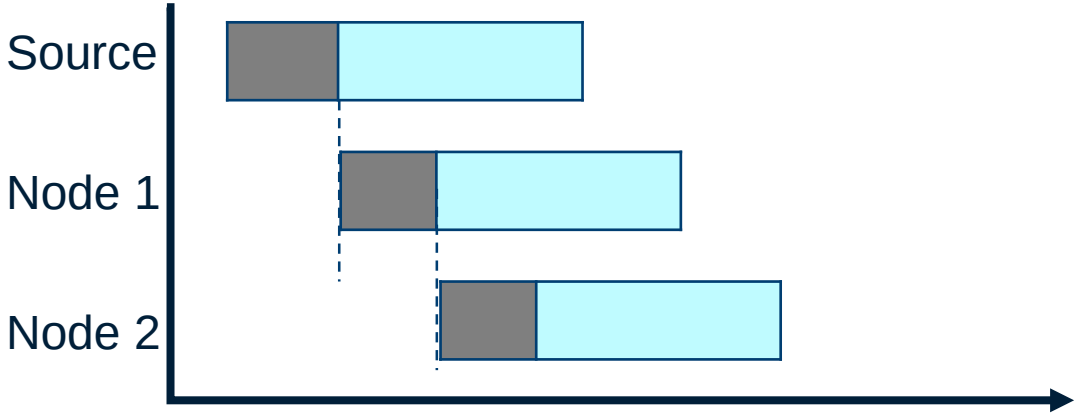
Cut Through Switching

Cut Through Switch

- Cut through switch ≠ Low latency switch
  - A cut through switch can implement a very long pipeline…
- But:
  - For the smallest packet, the latency is ~same
  - As packet size grows, latency saving grows

What is a cut-through switch?

- Kermani & Kleinrock, “Virtual cut-through: A new computer communication switching technique”, 1976
- “when a message arrives in an intermediate node and its selected outgoing channel is free (just after the reception of the header), then, in contrast to message switching, the message is sent out to the adjacent node towards its destination before it is received completely at the node; only if the message is blocked due to a busy output channel is a message buffered in an intermediate node.”

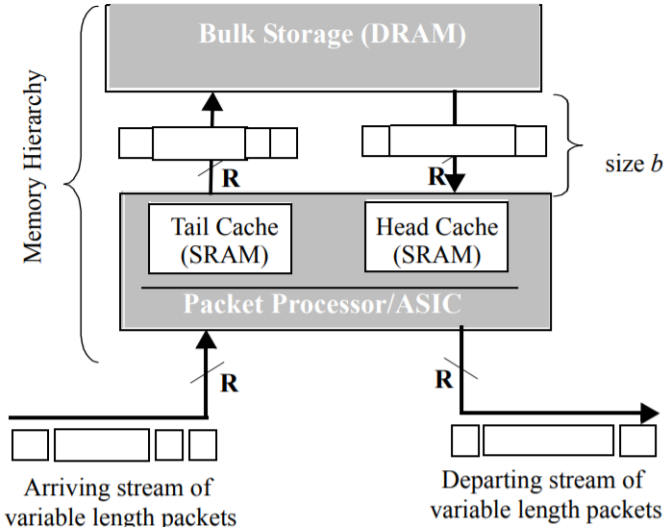


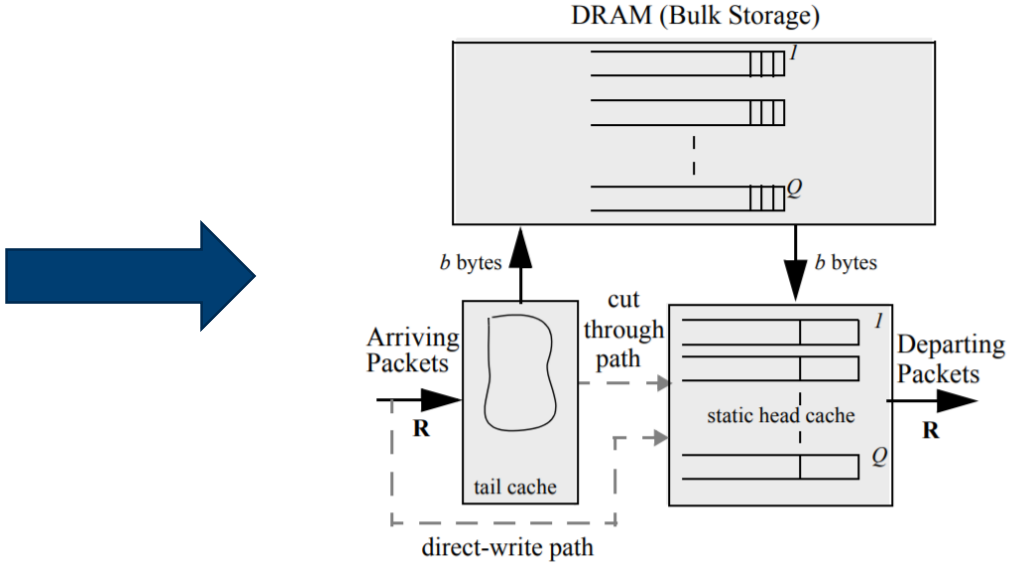
What is a cut-through switch?

- Past (far back):
  - Networks were slow
  - Memory was fast
  - Writing packets to the DRAM took “negligible” time
- With time:
  - Networks become faster
  - Memory access time is no longer “negligible”

What is a cut-through switch?

- Sundar, Kompella, and McKeown. "Designing packet buffers for router line cards." *2002.*





What is a cut-through switch?

- But what does a REAL silicon implementation looks like?
- Tip 1: search for patents on Google Scholar
- Tip 2: read *carefully* performance evaluation reports
  - We'll discuss some examples in the next lectures

Latency considerations within modules

Network Interfaces

- Data arrives at (up to) ~50Gbps per link.
- Let us ignore clock recovery, signal detection etc.
- Feasible clock rate is ~1GHz
- But if data rate is ×50 times faster…

- Observation: data bus width will be no less than incoming data rate and feasible clock rate

Network Interfaces

- Line coding often directs the bus widths:
  - E.g., 8b/10b coding led to bus widths of 16b (20b) or 64b (80b)
- A port is commonly an aggregation of multiple serial links
  - 10G XAUI = 4 × 3.125Gbps
  - 100G CAUI4 = 4 × 25Gbps
  - 400G PSM4 = 8 × 50Gbps
  - Need to take care of aligning the data arriving from multiple links on the same port.

Network Interfaces

- Role: check the validity of the packet (e.g., FCS)
- What to do if an error is detected?
  - Forward an error using a “fast path”
  - Mark the last cycle of the packet
    - E.g., to cause drop in the next hop
- Other roles need to be maintained too
  - Frame delimiting and recognition, flow control, enforcing IFG, …

Packet Processing

- A likely flow:

- Possible implementations:
  - The entire packet goes through the header processing unit
  - Just the header goes through the header processing unit
  - “Better” depends on your performance profile (what are the bottlenecks? Resource limitations?)

Packet Processing

- A likely flow:

- Challenges:
  - A field may arrive over multiple clock cycles (e.g. 32b field, 16b on clock 2 and 16b on clock 3)
  - Memory access taking more than 1 clock cycle
    - E.g. request on clock 1, reply on clock 3
    - Some memories allow multiple concurrent accesses, some don’t
    - The bigger the memory, the more time it takes

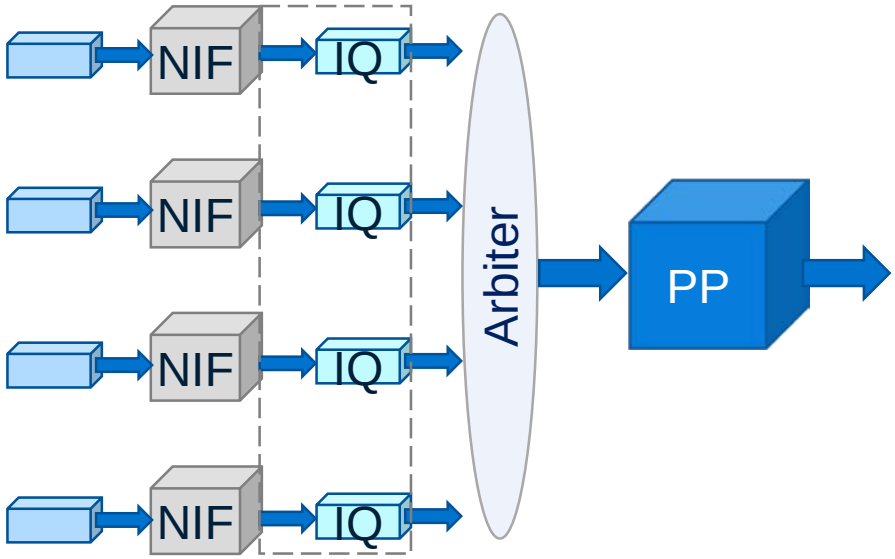
Packet Processing

- A likely flow:

- Solutions:
  - Pipelining!
    Don’t stall, add NOP stages in your pipe.
  - Reorder operations (where possible)
    - E.g. Lookup 1 → Action 1 → Lookup 2 → Action 2 turns:
      Lookup 1 → Lookup 2 → Action 1 → Action 2
    - Don’t create hazards!

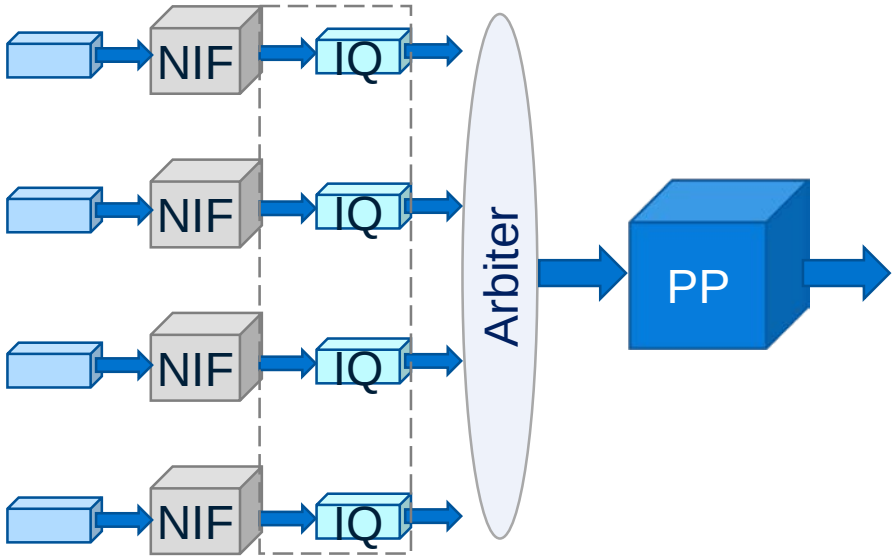
Arbitration

- Simple example:
  - Packets arriving from 4 ports
  - (approximately) same arrival time
  - Arbiter uses Round Robin
- Problem: arbitration on packet boundaries?
  - No: interleaved packets within the pipeline
    Need to track which cycle belongs to which packet
    May require multiple concurrent header lookups
    Order is not guaranteed (e.g. P1-P2-P3-P1-P2-P2-…), due to NIF timing



Arbitration

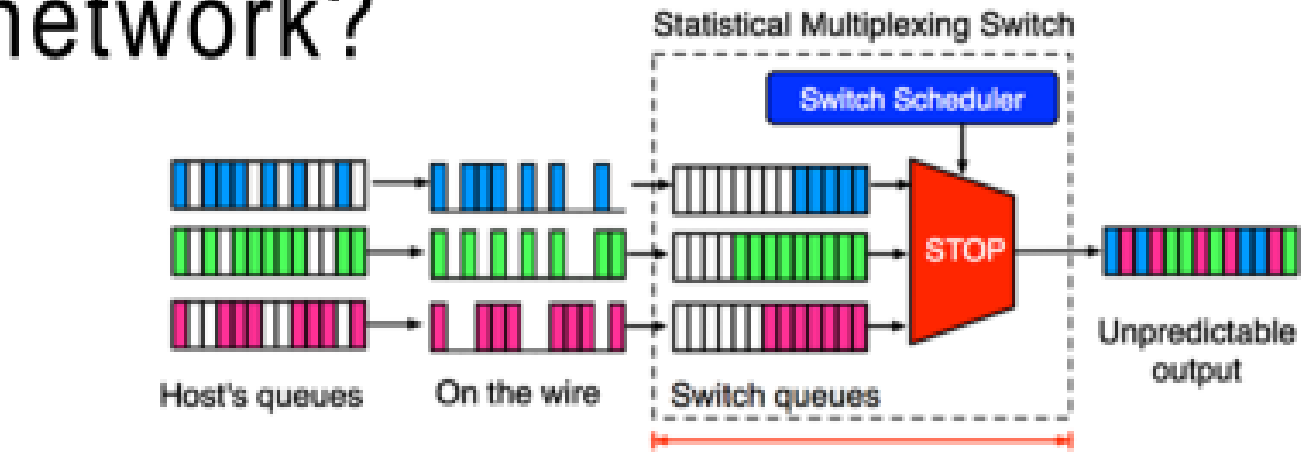
- Simple example:
  - Packets arriving from 4 ports
  - (approximately) same arrival time
  - Arbiter uses Round Robin
- Problem: arbitration on packet boundaries?
  - Yes: packets need to wait for previous packets to be handled before being admitted.
    Worst case waiting with <N> inputs is <N-1>×Packet time

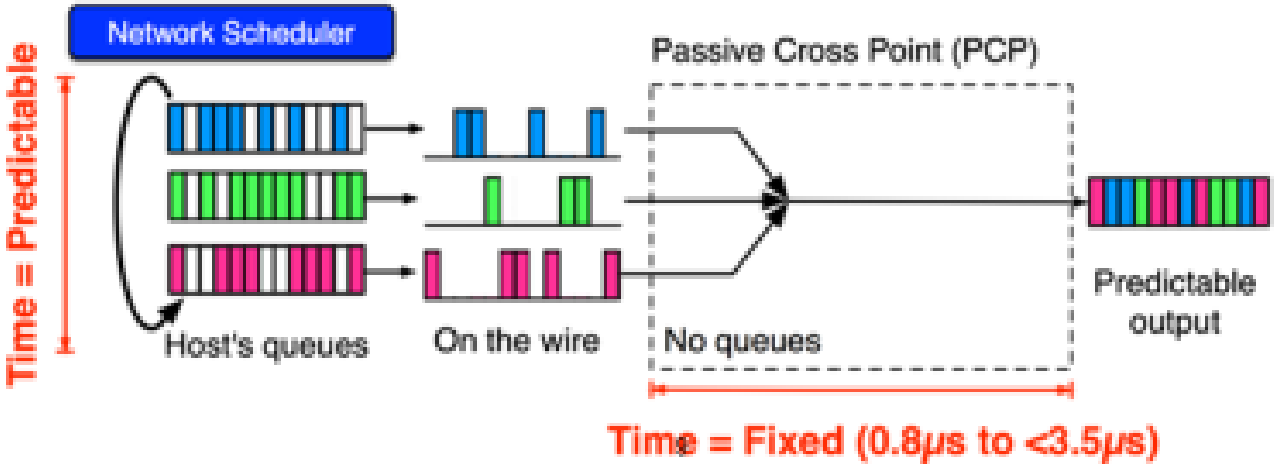


Arbitration

- Solutions to the previous problem:
  - Scheduled (or slotted) traffic
  - Multiple pipelines
  - …

How do you build a bufferless network?

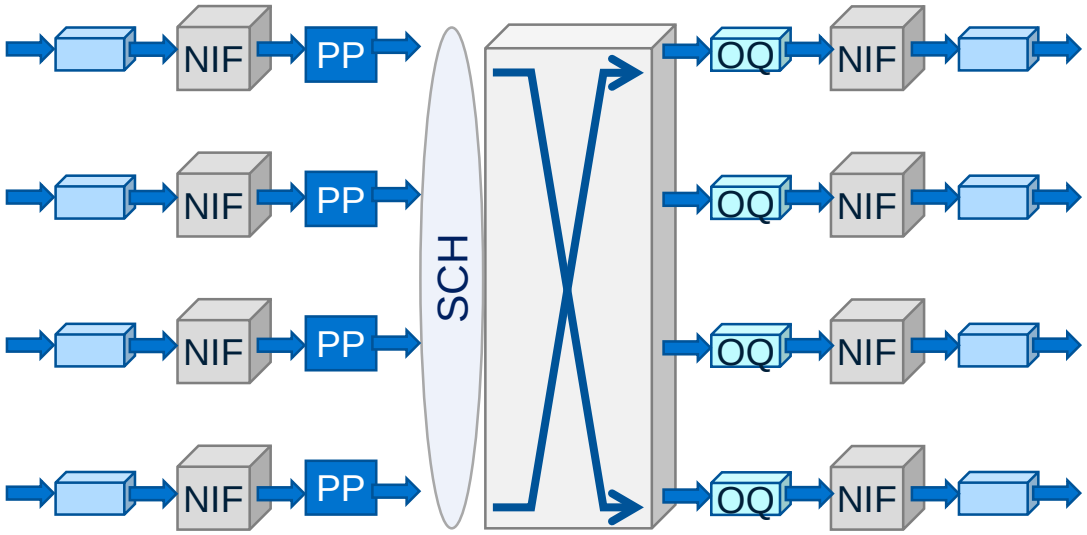




M Grosvenor *et al* "The Resilient Realtime Data Delivery Network Architecture" (R2D2) 2013

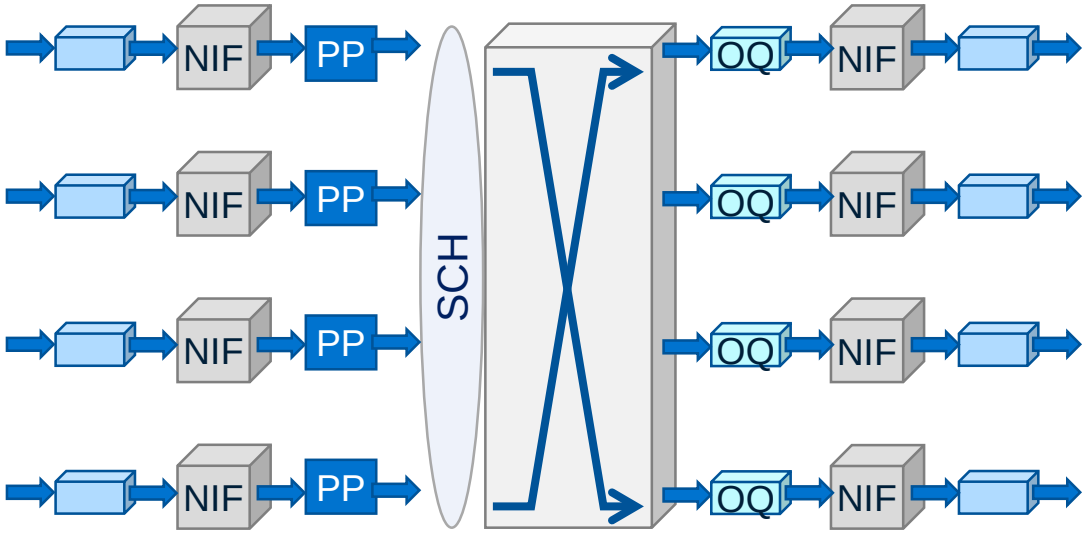
Arbitration

- This example solves the arbitration problem entering the device
- Resource inefficient:
  - Pipeline overdesign
  - Inefficient use of memories
  - Concurrency issues
- One solution:
  - Shared memories / tables
  - Highly complex

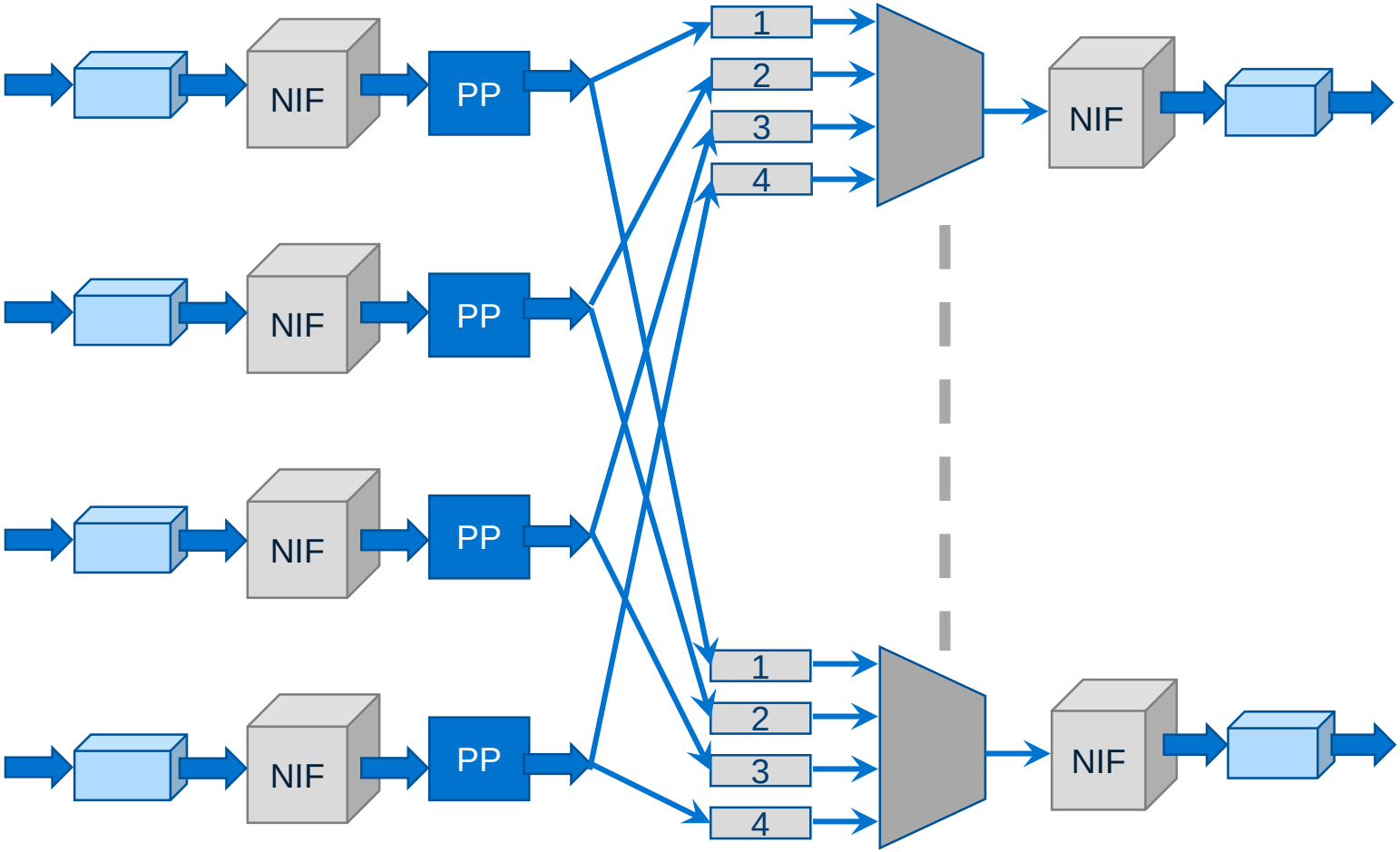


Switching

- The previous arbitration solution “pushed” the problem to the switching unit
- But now the problem is only when multiple packets compete over the same output – that’s fine!
- Assuming your switch can handle multiple packets per cycle
  - E.g. crossbar



Switching



Switching

- … This is also queueing
- Challenge: SCALE
  - So you can do it with 4 ports
  - Can you do it with 32? 128? 256?
- Not just resource / area
  - Computation time – being able to examine and choose between all available inputs
- Eventually:
  Packets must be sent out on packet boundaries
  do not interleave packets!