

# P51: High Performance Networking

## Lecture 2: High Throughput Devices

# Bandwidth, Throughput and Goodput

- Bandwidth – how much data can pass through a channel.
- Throughput – how much data actually travels through a channel.
- Goodput is often referred to as application level throughput.

But bandwidth can be limited below link's capacity and vary over time, throughput can be measured differently from bandwidth etc.....

# Speed and Bandwidth

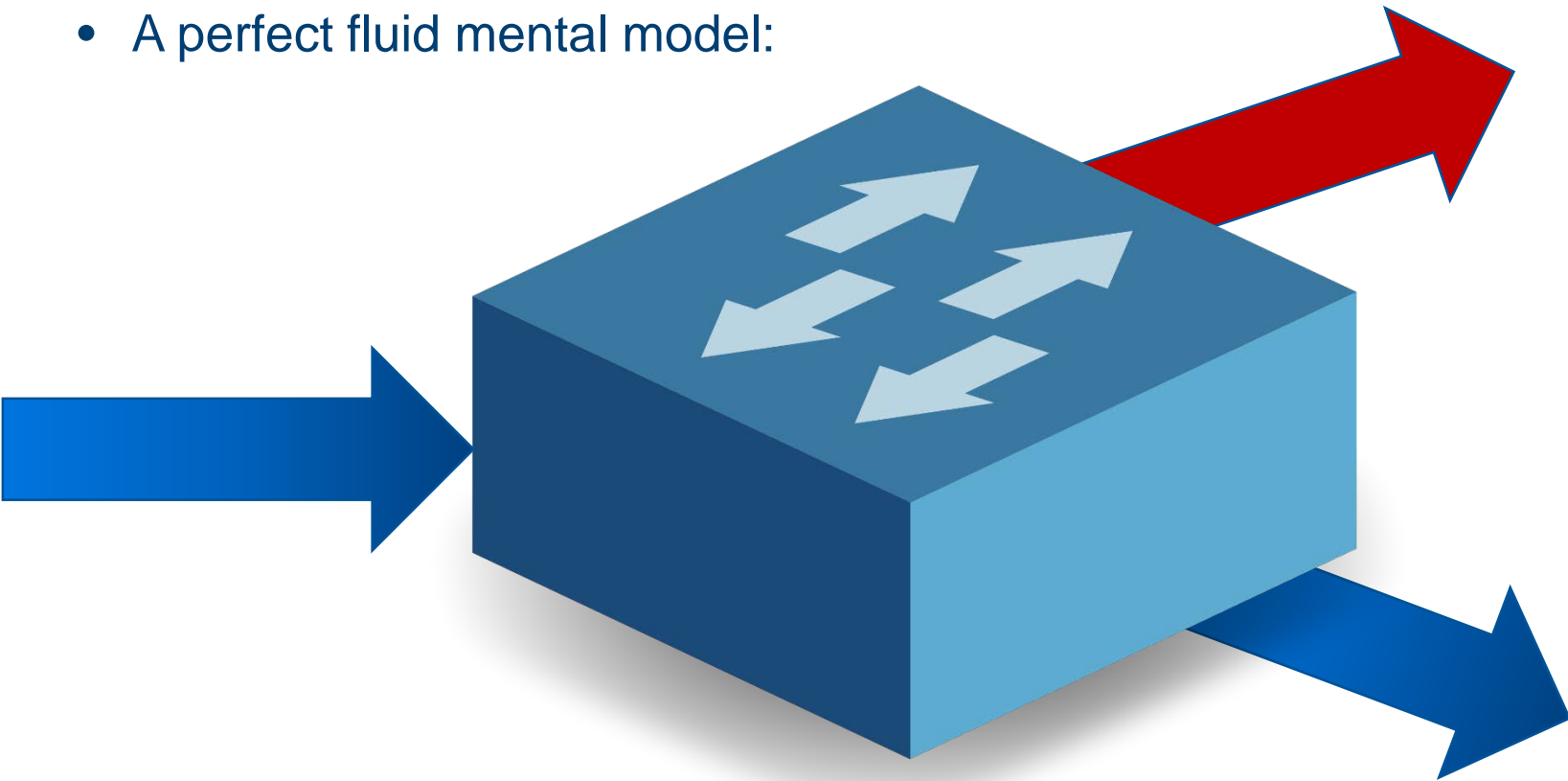
- Higher bandwidth does not necessarily mean higher speed
- E.g. can mean the aggregation of links
  - $100\text{G} = 2 \times 50\text{G}$  or  $4 \times 25\text{G}$  or  $10 \times 10\text{G}$
  - A very common practice in interconnects

# Packet Rate

- Throughput may change under different conditions, e.g. packet size
- Packet Rate: how many packets can be processed in a given amount of time
  - Also changes under different conditions
  - But often provides better insights

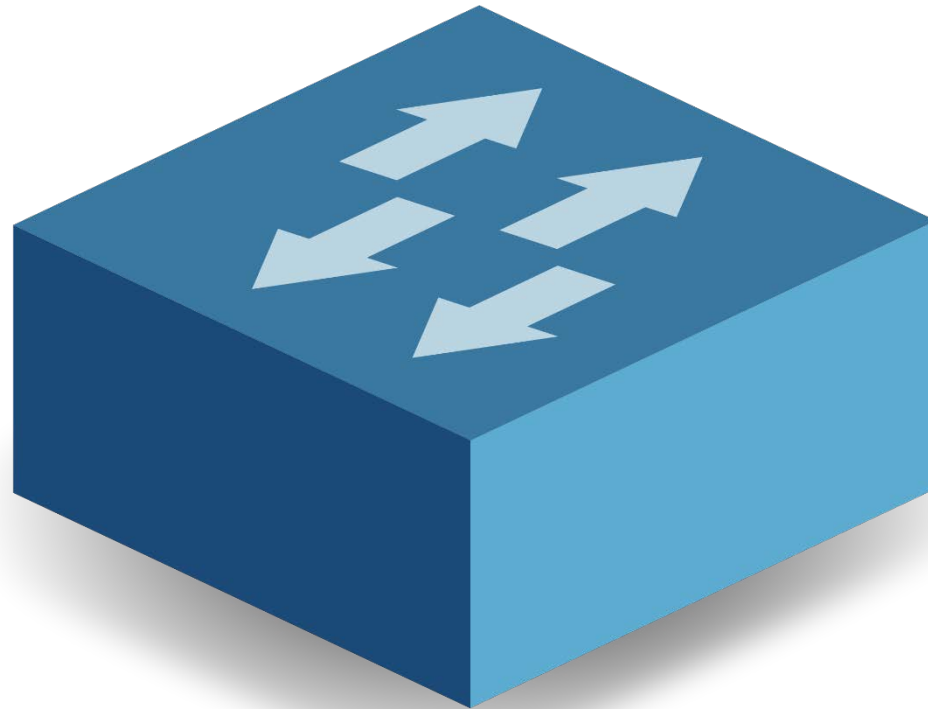
# Switch Models

- A perfect fluid mental model:



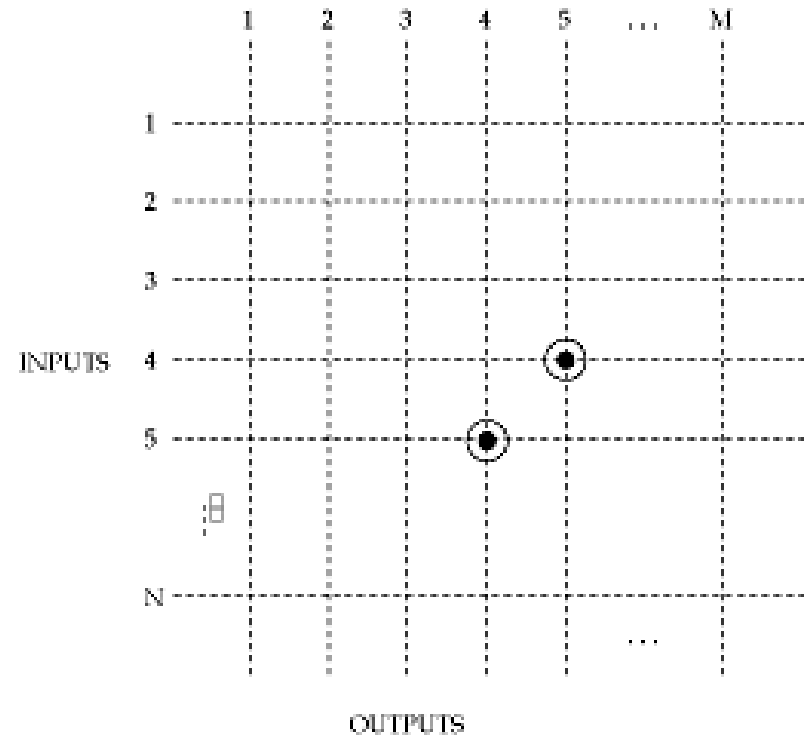
# Switch Models

- A single packet mental model:



# Circuit Switches

- Input A is connected to output X
- Example: a crossbar
  - Not the only option
- Used mostly in optical switching
  - No header processing!
- But also in electrical switching
  - E.g. high frequency trading (HFT)
- Scheduling is a limiting factor



# Packet Switches

- In a circuit switch:

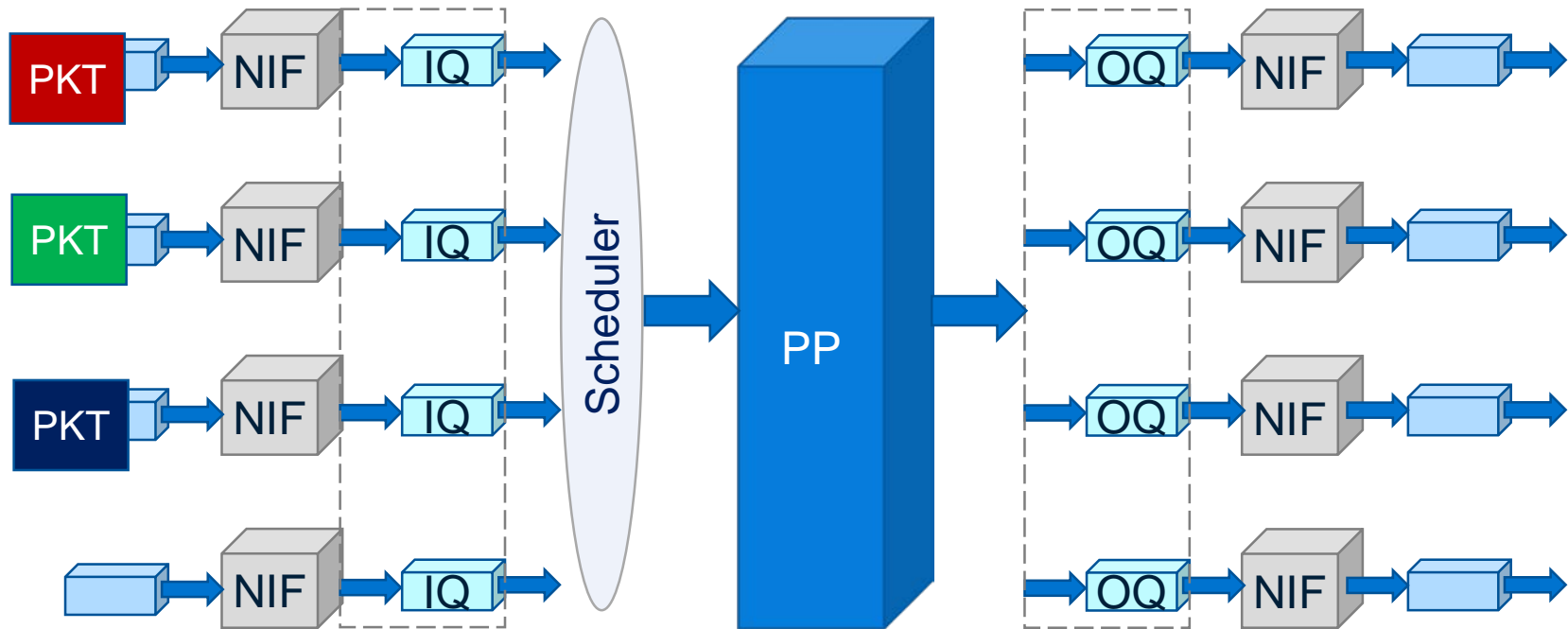
The path of a sample is determined at time of connection establishment

- In a packet switch, packets carry a destination field
- Need to look up destination port on-the-fly
  - Two sequential packets may head to different destinations



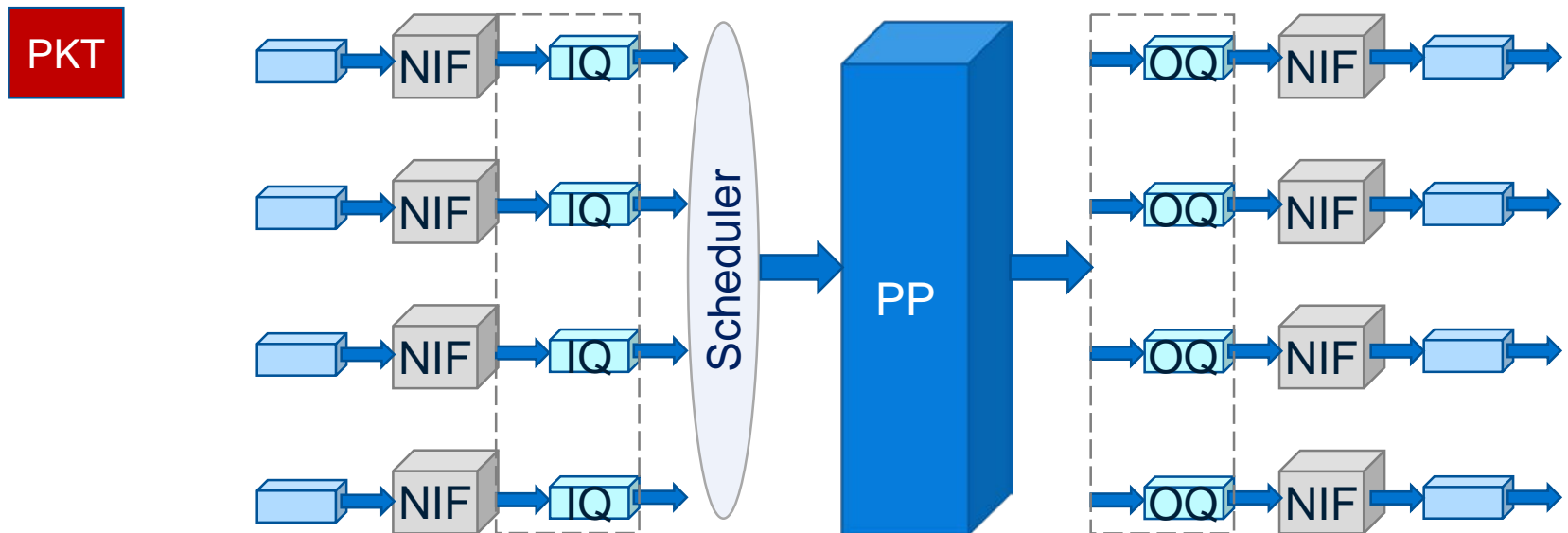
# Pipelining

To achieve high throughput, packet switches are pipelined:



# Store and Forward

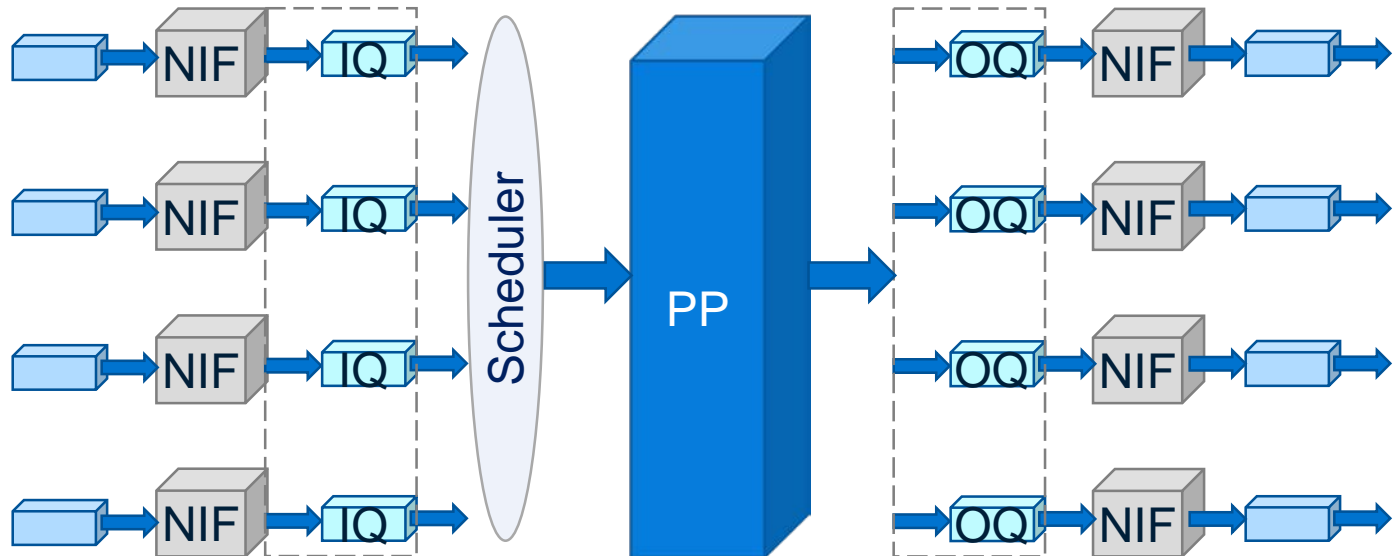
- Wait for the entire packet to arrive
- Check the FCS, then start processing
  - FCS – frame check sequence, terminates the packet
- Once the packet is checked, it starts propagating through the pipeline
  - Not necessarily the entire packet



# Cut Through

- Start processing the packet as soon as the first chunk arrives
  - Do not wait for the FCS
- If FCS error is detected, the packet is dropped somewhere along the pipeline

PK13

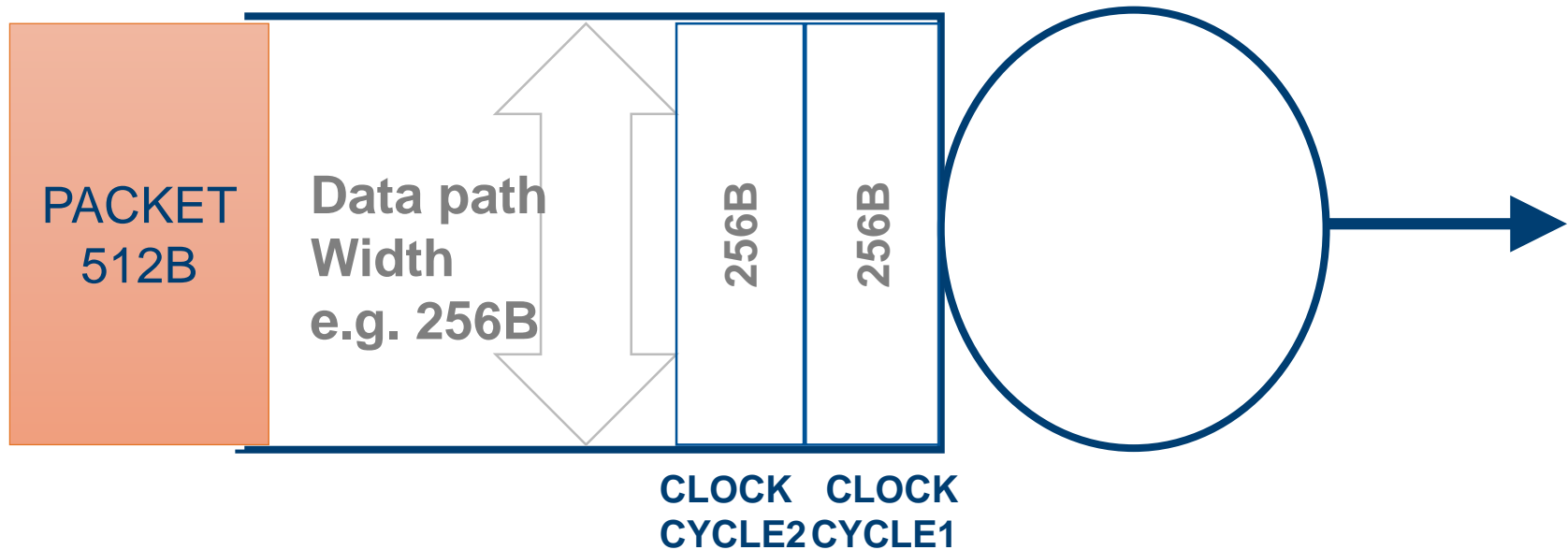


# Measuring Performance

- Bandwidth: number of bits (or bytes) through the channel every unit of time
  - One way to calculate: *bus width*  $\times$  *clock frequency*

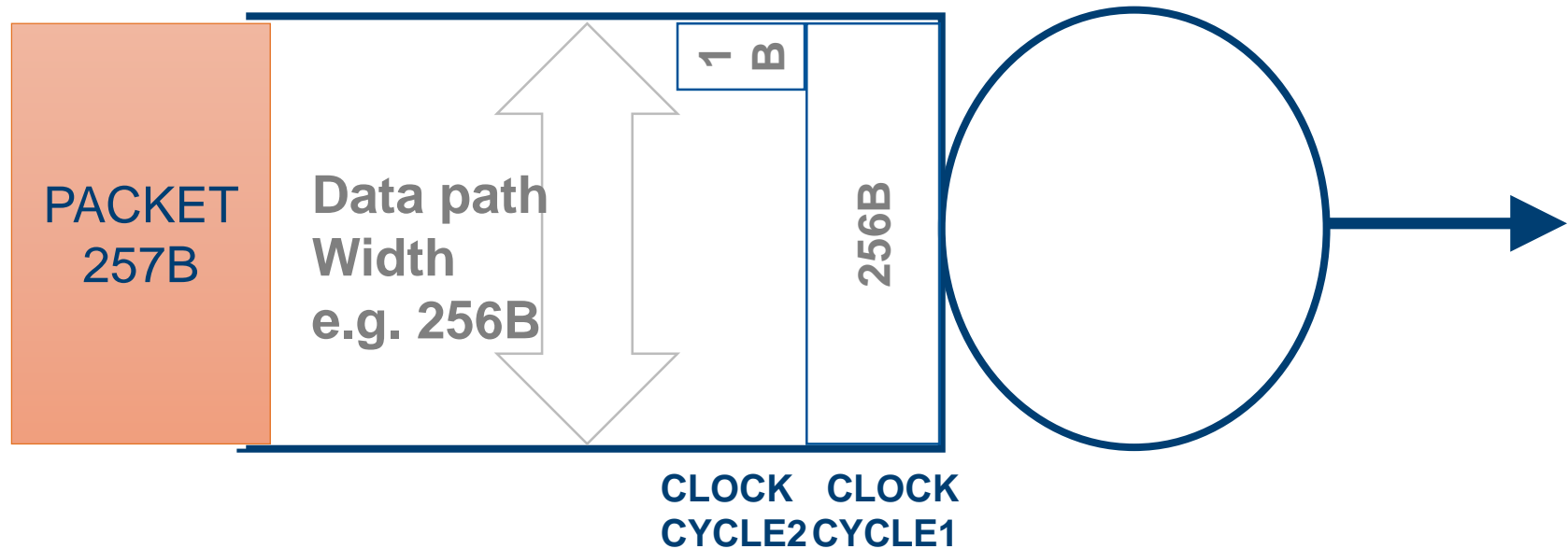
# Measuring Performance

Throughput = clock frequency x bus width ?

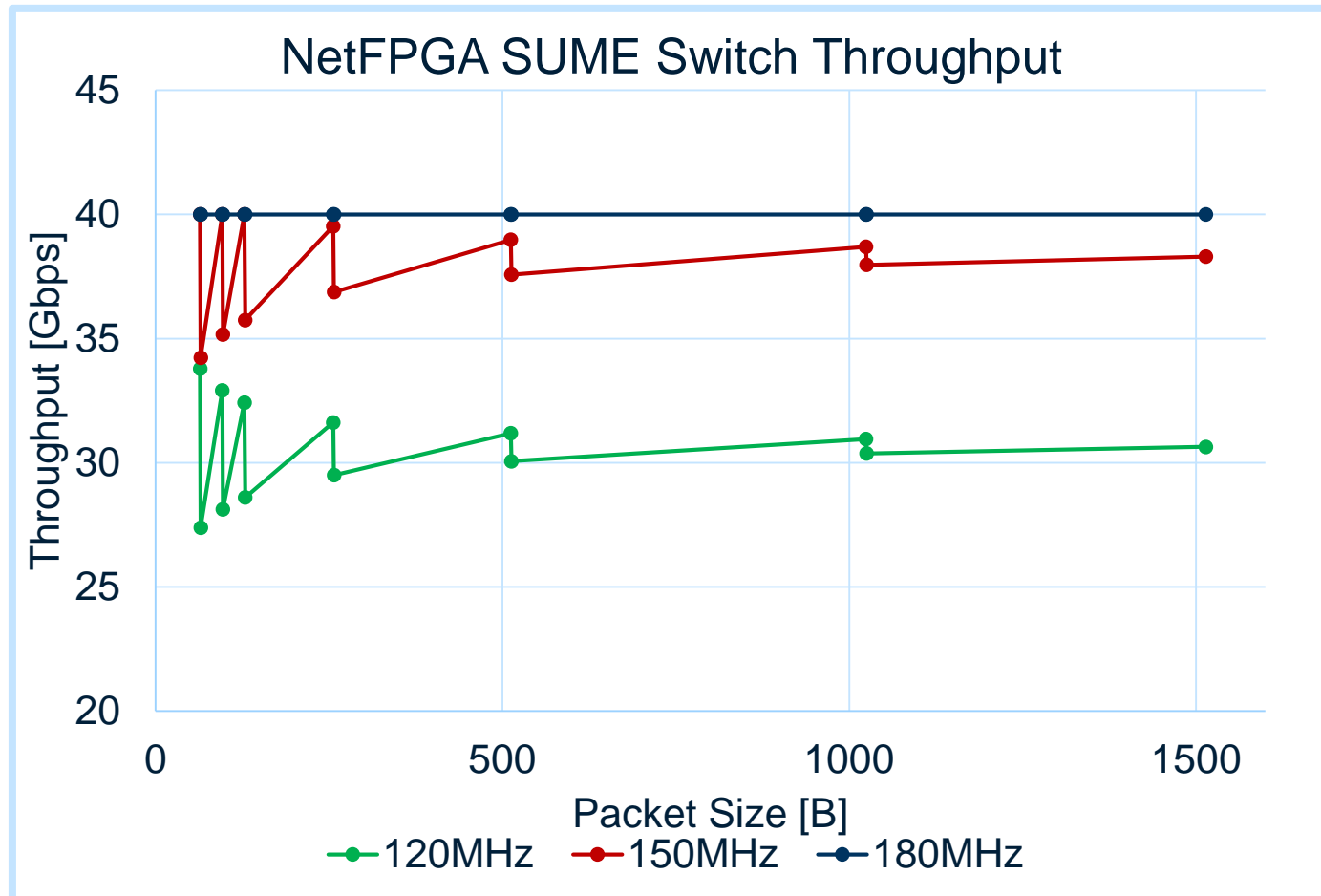


# Measuring Performance

Throughput  $\neq$  clock frequency  $\times$  bus width !



# Example: NetFPGA SUME – Switch Throughput



# Performance Profile

- An aggregation of the performance estimates of a device
- Considering both I/O and modules
- Helps identify bottlenecks
- Informs design decisions, e.g.:
  - Bus width
  - Clock frequency



# Packet Rate “on the wire”

- Calculating the packet rate on (e.g.) 10GE port:
- Line rate = 10.3125Gbps
- Line coding 64b/66b
  - $10.3125\text{G} \times 64/66 = 10\text{Gbps}$
- Before each Ethernet packet there is 8B preamble and 12B Inter Packet Gap (IPG)
  - Data rate:  $10.3125\text{G} \times 64/66 \times (\text{packet size}) / (\text{packet size} + 12\text{B} + 8\text{B})$
- Divide it by packet size and convert bytes to bits:
  - Packet rate:  $10.3125\text{G} \times 64/66 / ((\text{packet size} + 12\text{B} + 8\text{B}) * 8)$

# Packet rate within a device

- “Cycles per packet”
  - How many cycles are required to process a single packet within a module?
- Example 1: Cycles required to fit a packet into the data bus
- On NetFPGA SUME:
  - 10G Port: 64b (8B) wide
    - 64B packet: 8 clocks
    - 65B packet: 9 clocks
  - Data path: 256b (32B) wide
    - 64B packet: 2 clocks
    - 65B packet: 3 clocks

# Packet rate within a device

- Why 64b bus in the 10G port?
  - $64\text{b} \times 156.25\text{MHz clock} = 10\text{Gbps}$
  - Also, 64b fits the 64b/66b coding
    - No need for a gear box
- Why 256b bus in the data path?
  - Need to process data from  $4 \times 10\text{G}$  ports
  - $4 \times 64\text{b} = 256\text{b}$
- Important! Packet size may differ between modules
  - 10G port packet size includes FCS, in the data path FCS is not included
    - 4B difference

# Packet rate within a device

- “Cycles per packet”
  - How many cycles are required to process a single packet within a module?
- Example 2: Packet processing
  - How many cycles are required to process the packet’s header?
  - E.g. Look up in the memory
  - Actions that are not pipelined
- Example 3: Scheduling / Arbitration
  - How many cycles are required to schedule a packet?
  - E.g. can a new packet enter the arbiter every clock cycle?

# Performance profile - example

- 10G Port clock – 156.25MHz
- Data path clock – 200MHz

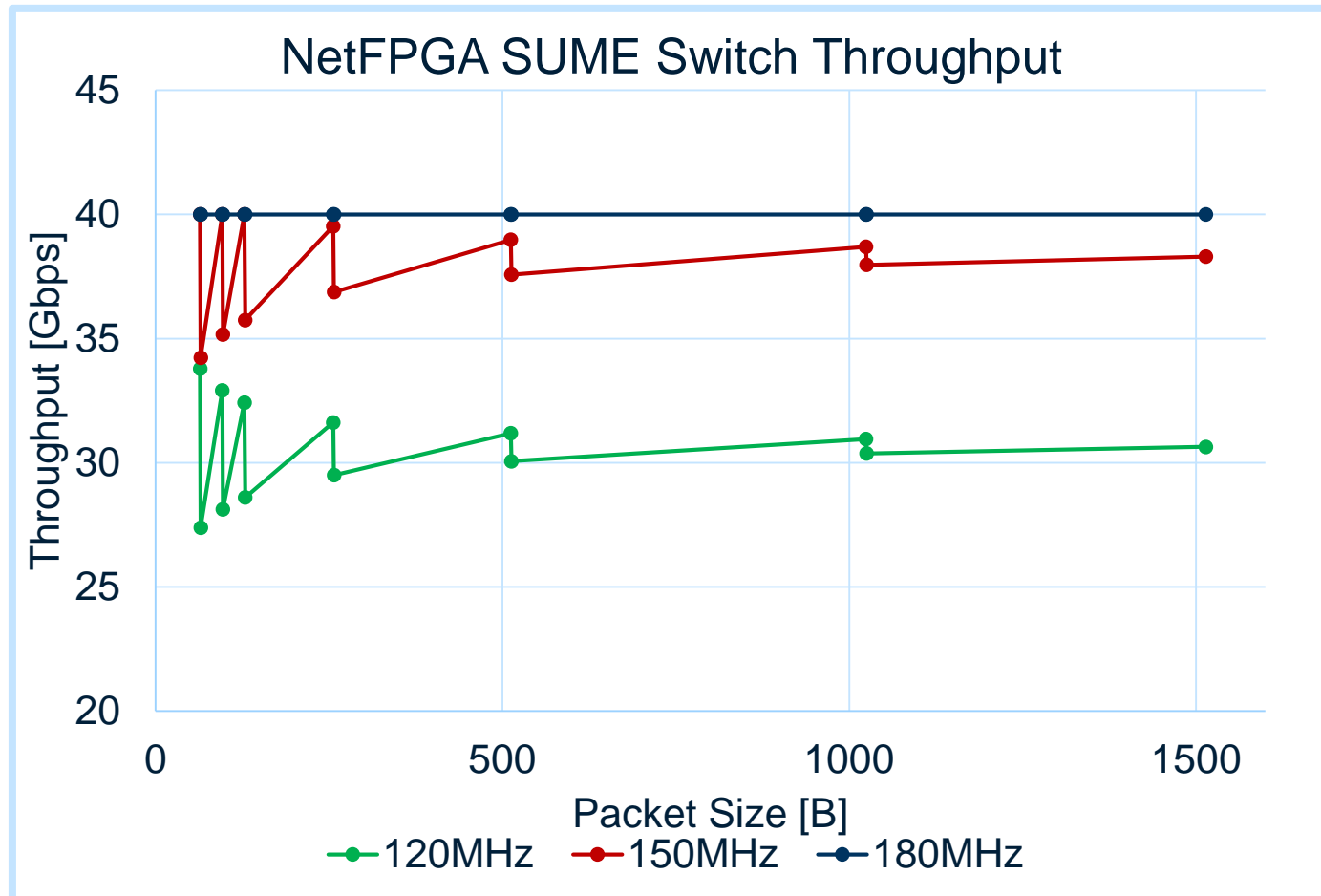
Packet Size [B]	“Wire” packet rate [Mpps]	Clocks per packet (10G port)	MAX packet rate (10G port) [Mpps]	“data path” packet rate [Mpps]	Clocks per packet (data path)	MAX packet rate (data path) [Mpps]	Speed Up
64	14.88	8	19.53	59.52	2	100	1.68
65	14.71	9	17.36	58.82	3	66.66	1.14
66	14.53	9	17.36	58.14	3	66.66	1.15
...							

# Performance profile - example

- 10G Port clock – 156.25MHz
- Data path clock – **150MHz**

Packet Size [B]	“Wire” packet rate [Mpps]	Clocks per packet (10G port)	MAX packet rate (10G port) [Mpps]	“data path” packet rate [Mpps]	Clocks per packet (data path)	MAX packet rate (data path) [Mpps]	Speed Up
64	14.88	8	19.53	59.52	2	75	1.26
65	14.71	9	17.36	58.82	3	50	<b>0.85</b>
66	14.53	9	17.36	58.14	3	50	<b>0.86</b>
...							

# Example: NetFPGA SUME – Switch Throughput



# Packet rate within a device

- Question:
- 10G Port supports full line rate at 156.25MHz
- Data path supports full line rate for 64B packets at 150MHz (measured!)
- But data path input is 4 x 10G port...
  
- *How do we get 100% throughput at a lower data path frequency?*
- *And what did we neglect?*



# The Truth About Switch Silicon Design

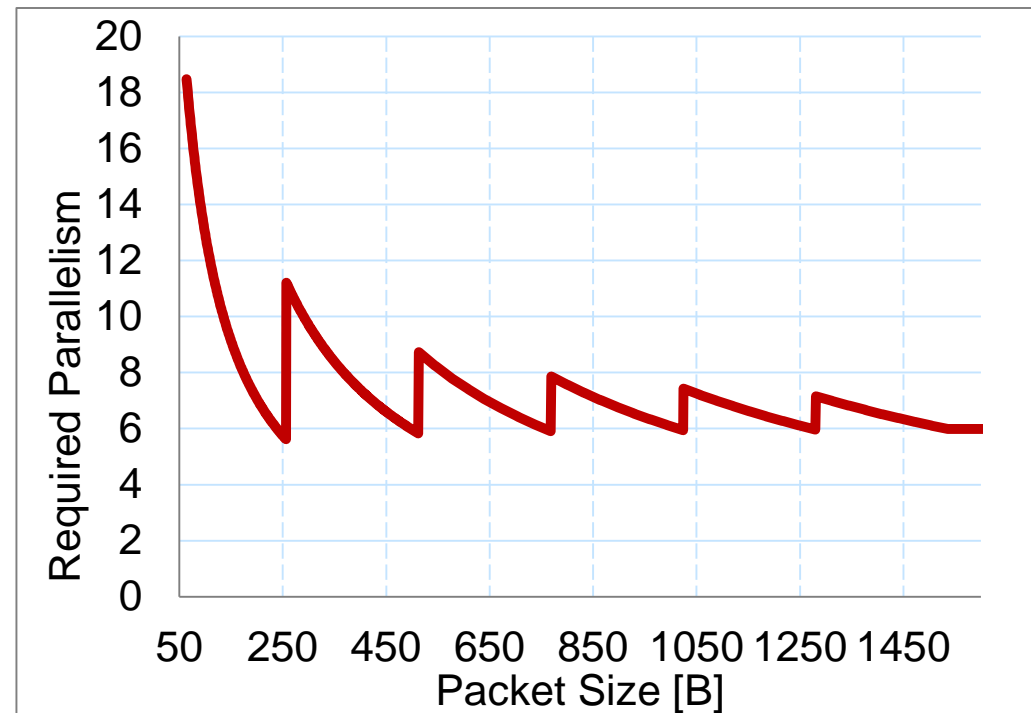
12.8Tbps Switches!

Lets convert this to packet rate requirements:

5.8 Gpps @ 256B

19.2 Gpps @ 64B

But clock rate is only ~1GHz.....



# Multi-Core Switch Design

