# 2. Link and Memory Architectures and Technologies

2.1 Links, Thruput/Buffering, Multi-Access Ovrhds

2.2 Memories: On-chip / Off-chip SRAM, DRAM

2.A Appendix: Elastic Buffers for Cross-Clock Commun.

*Manolis Katevenis*

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

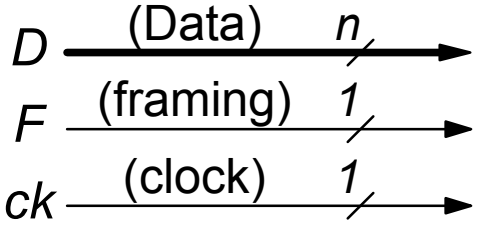
http://archvlsi.ics.forth.gr/~kateveni/534

# 2.1 Links, Throughput/Buffering, Multi-access Ovrhds

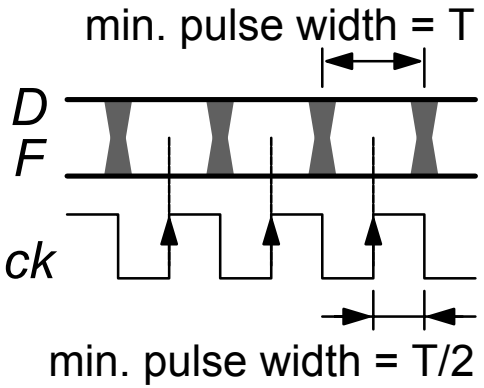
*Table of Contents:*

# 2.1.1 Parallel Transmission Links

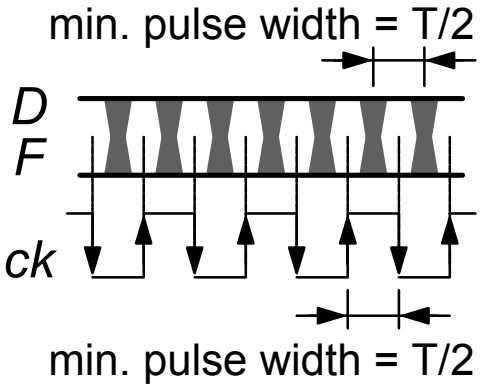
- "Out-of-band" Framing
  - start-of-packet, end-of-packet
  - valid word / idle line
- Usable for short distances (datapaths):
  - requires synchronization among wires
  - at high frequencies: source-synchronous (unidirectional) & partial-word clocking
- Signaling rate on Clock vs. Data Wires:
  - *Plain Clocking:* signaling rate on clock wire is *twice* the rate of data wires
  - *DDR (Double Data Rate):* double the signaling rate on all other wires as well, to match the maximum feasible rate presumably already used for the clock wire



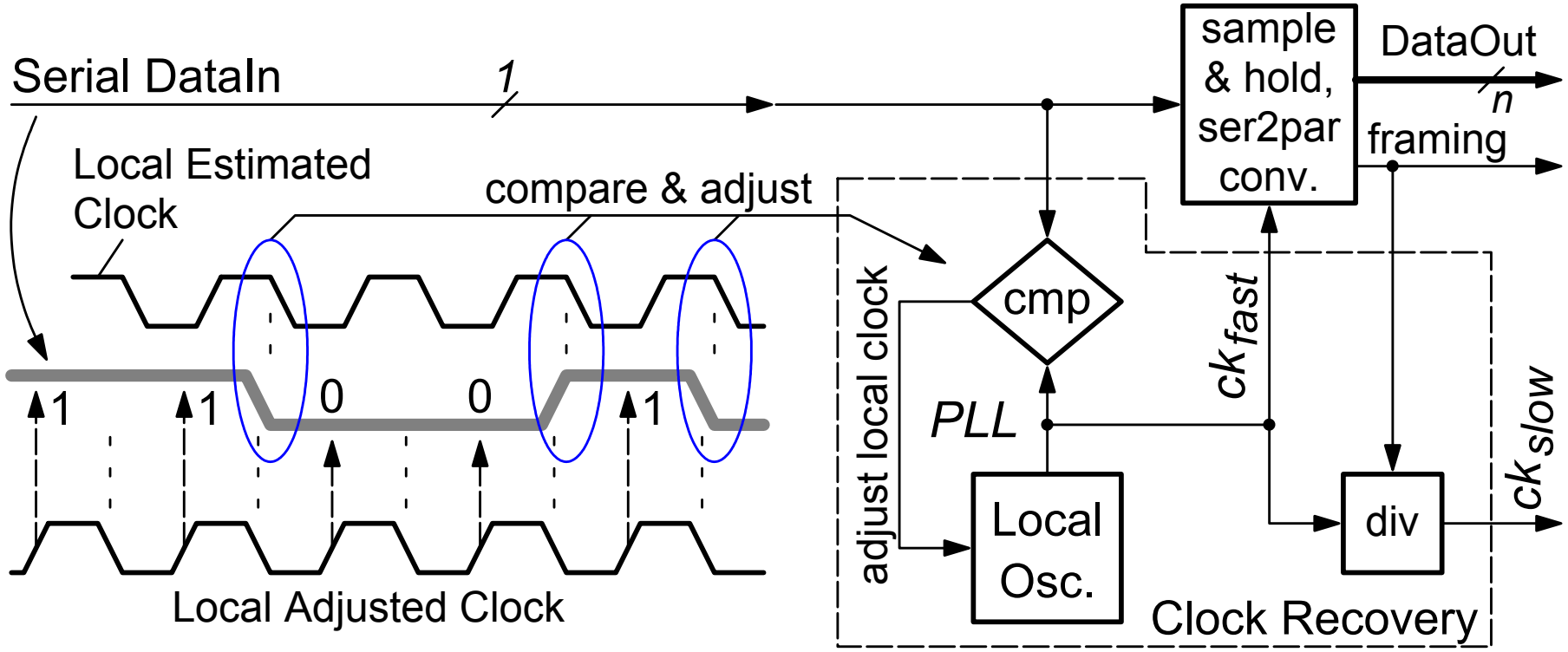

**Plain Clocking:**




**DDR Clocking:**

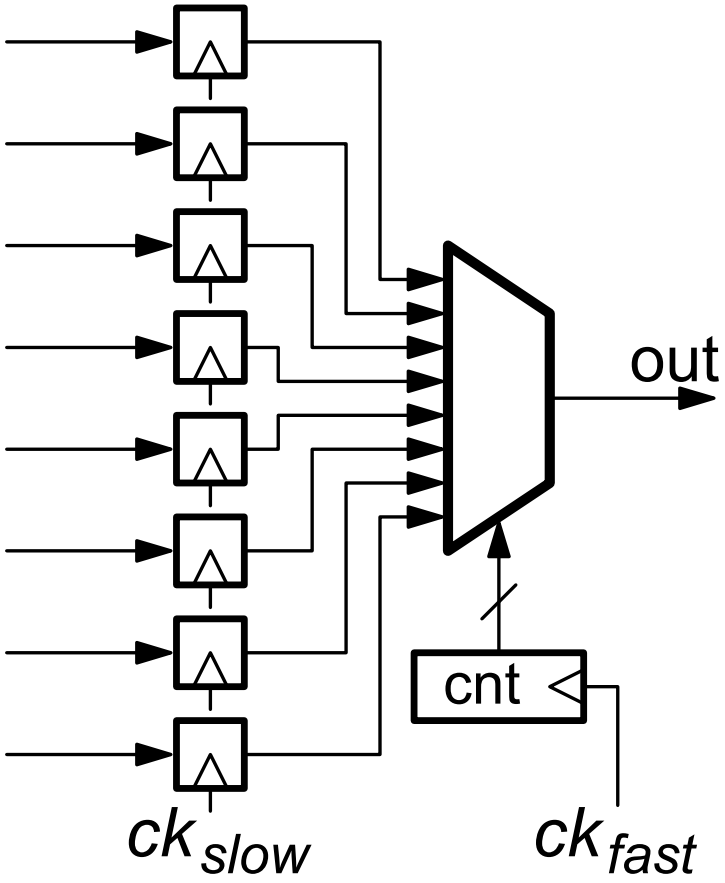


# Serial Transmission Links

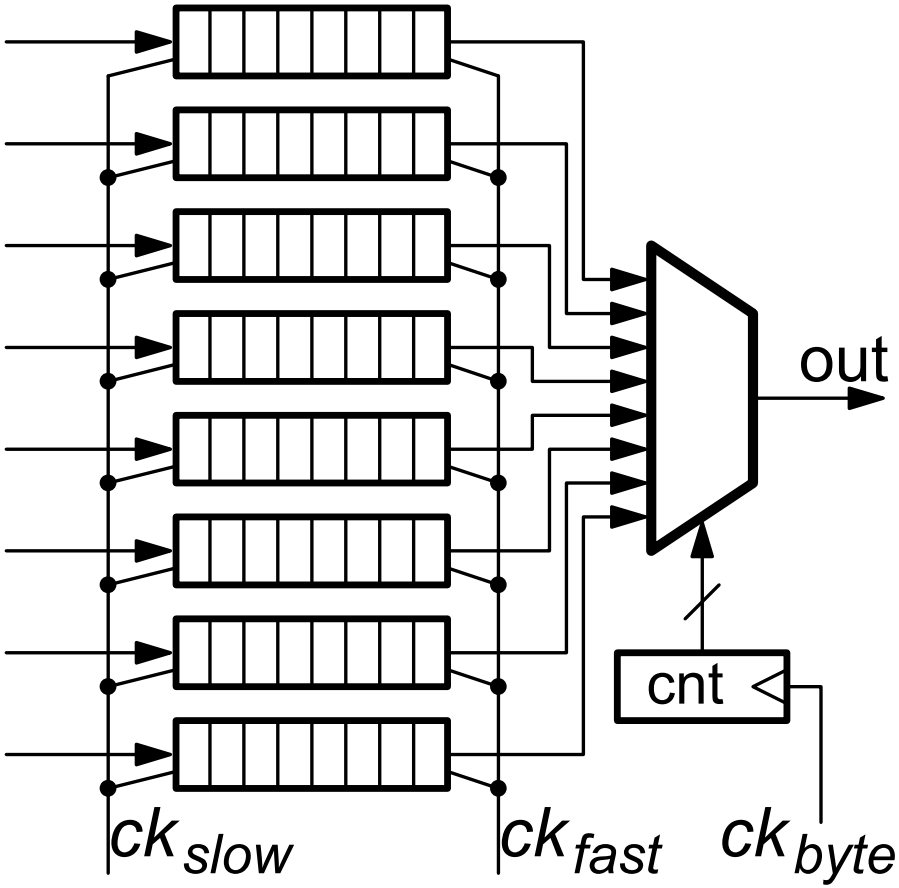



- Eliminate Timing Skew problem among wires; Reduce wire cost
- Need data to contain edges every so often for clock recovery (phase-locked loop – PLL) to work ⇒ line coding (e.g. 8b/10b)

# Parallel-to-Serial Conversion: Multiplexing



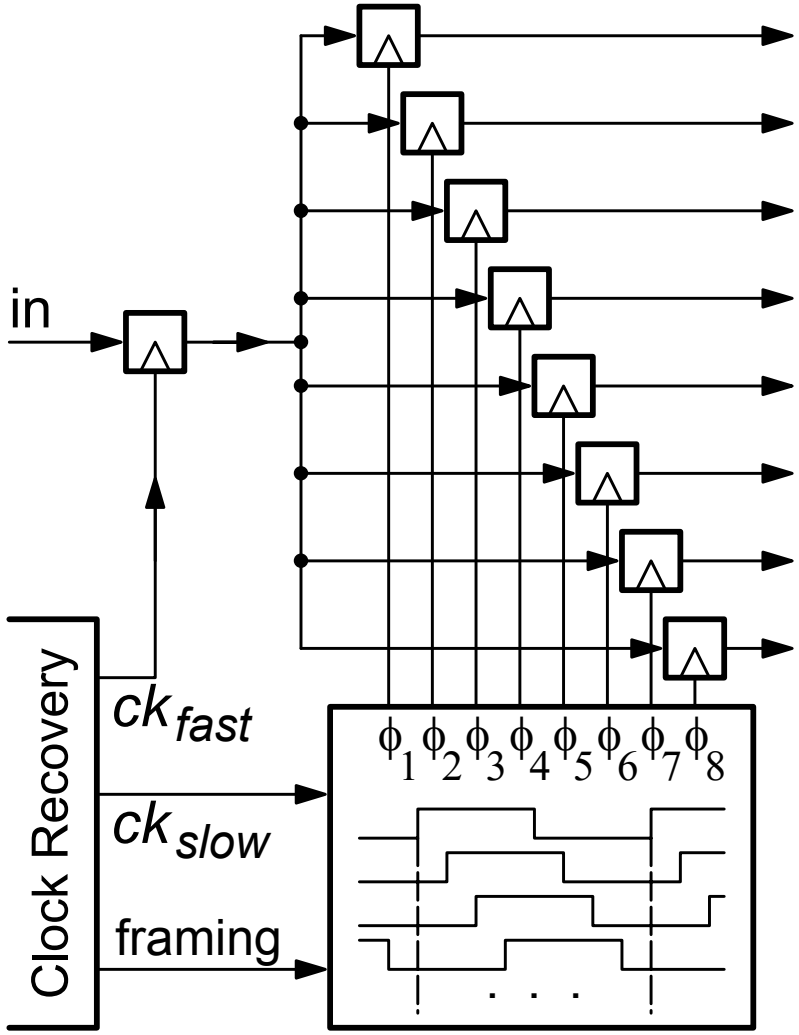

**Bit-Interleaved**



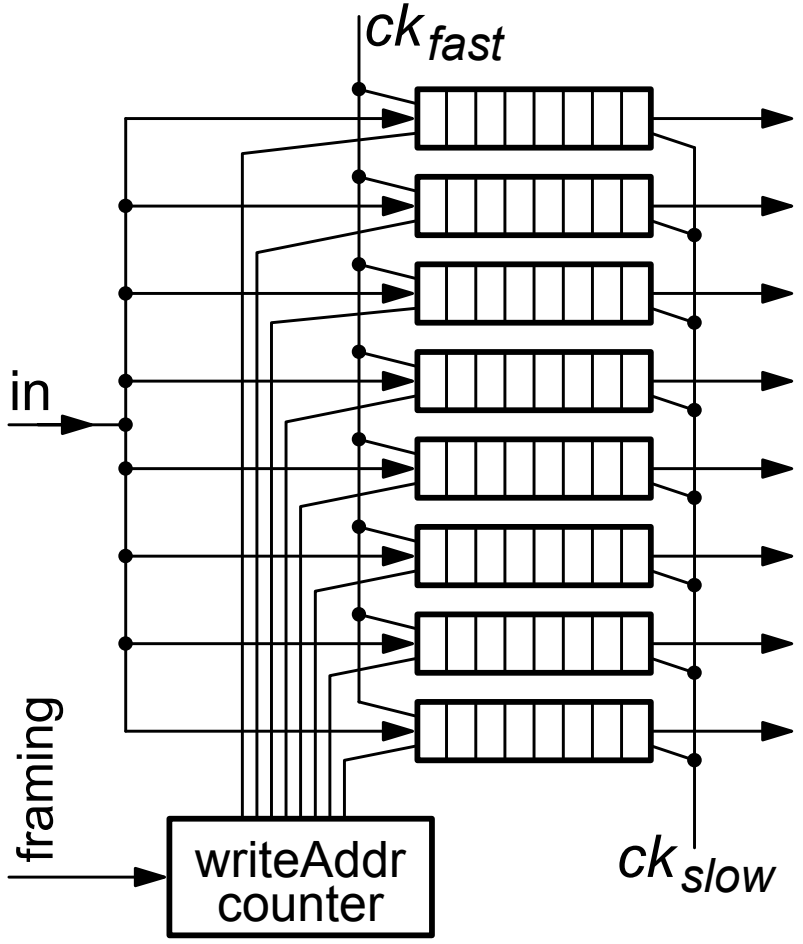

**Byte-Interleaved**

*Note:* buffer space = sizeof (one periodic frame)

# Serial-to-Parallel Conversion: Demultiplexing




**Bit-Interleaved**

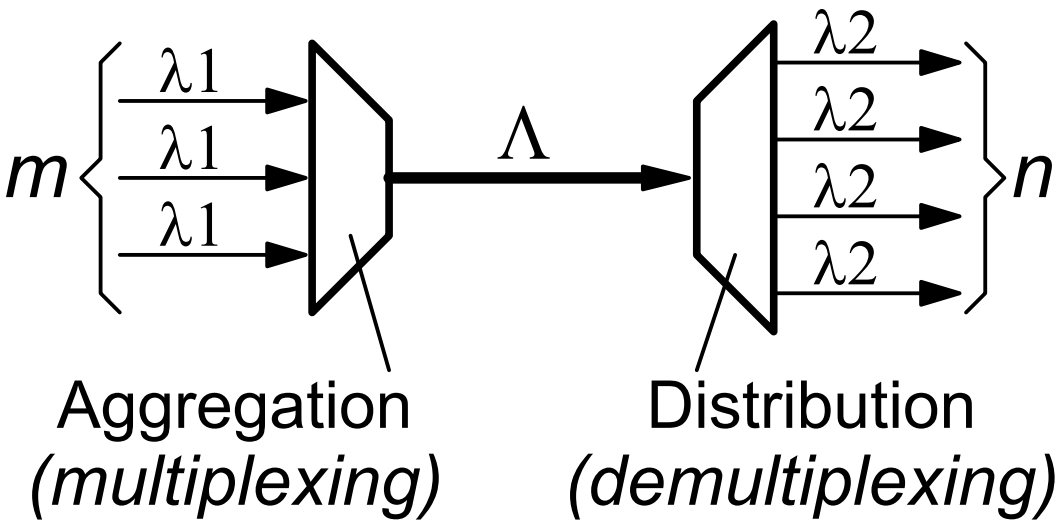



**Byte-Interleaved**

# Codes, Framing, Rate, Throughput,Capacity, Load

- Line coding provides extra *Control Characters*
  - used for framing: start/end-of-packet, idle, etc.
- Signaling Rate *(Baud Rate): electrical "symbols" / second*
  - e.g. quadrature modulation $\Rightarrow$ 1 symbol = 2 bits
- Transmission Rate: *raw* bits / second (raw *bps*)
- Throughput: *useful* bits / second (useful *bps*)
  - Throughput = Transmission Rate *minus* Overhead
- Capacity: *Peak* rate or throughput
- Load: *Current,* actual (average) throughput or (non-idle) rate

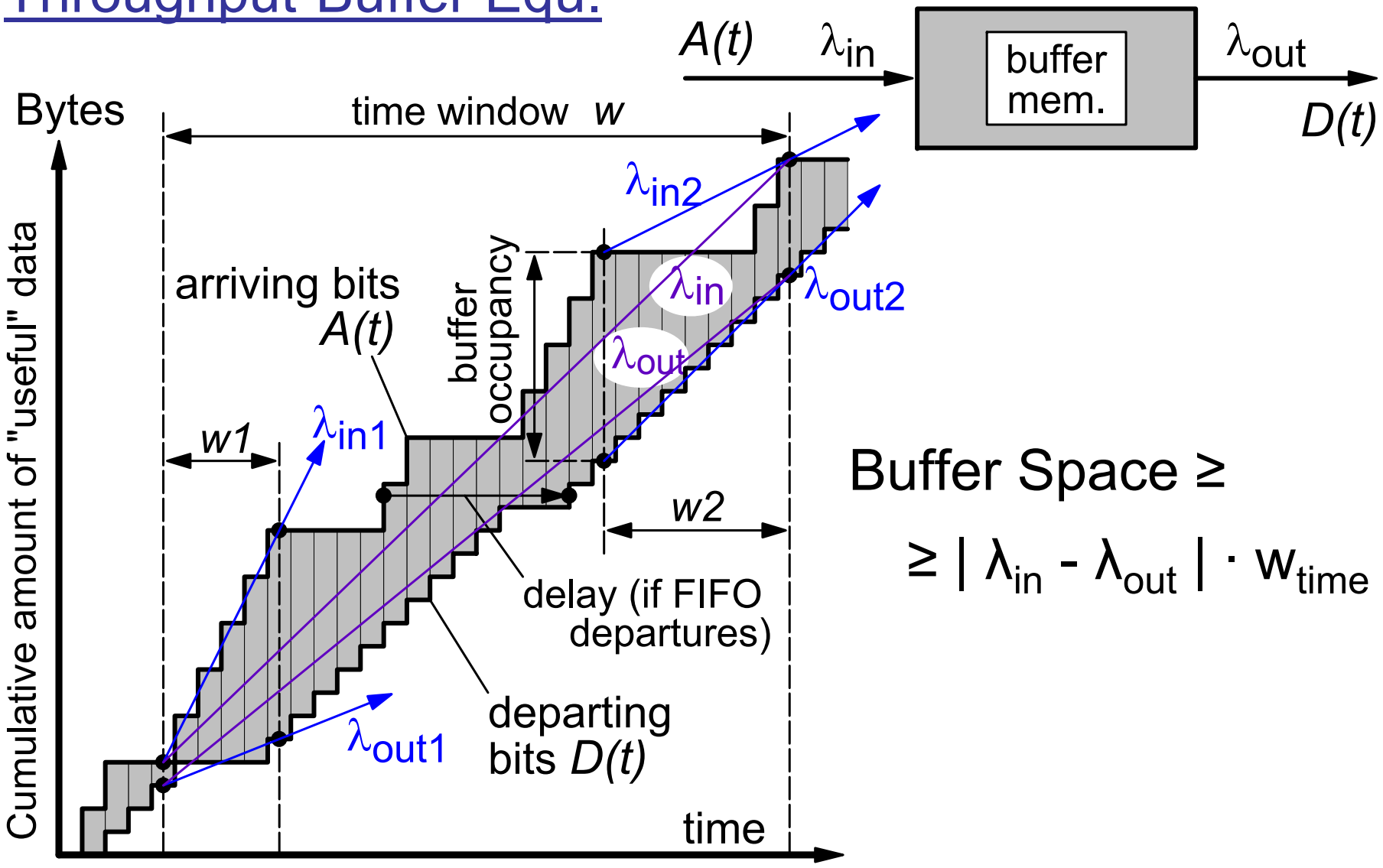
# 2.1.2 Throughput Conservation




$$m \cdot \lambda_1 = \Lambda = n \cdot \lambda_2$$

- either *Instantaneous* (no buffering) or *Average* (with buffering)
- what is conserved is the *useful-information* throughput
  - coding may change, idle bits may be added or removed, information may be filtered and/or selectively dropped, etc.

# Throughput-Buffer Equ.




Buffer Space ≥

$$\geq | \lambda_{in} - \lambda_{out} | \cdot w_{time}$$

# Throughput – Time – Buffer Space Equation

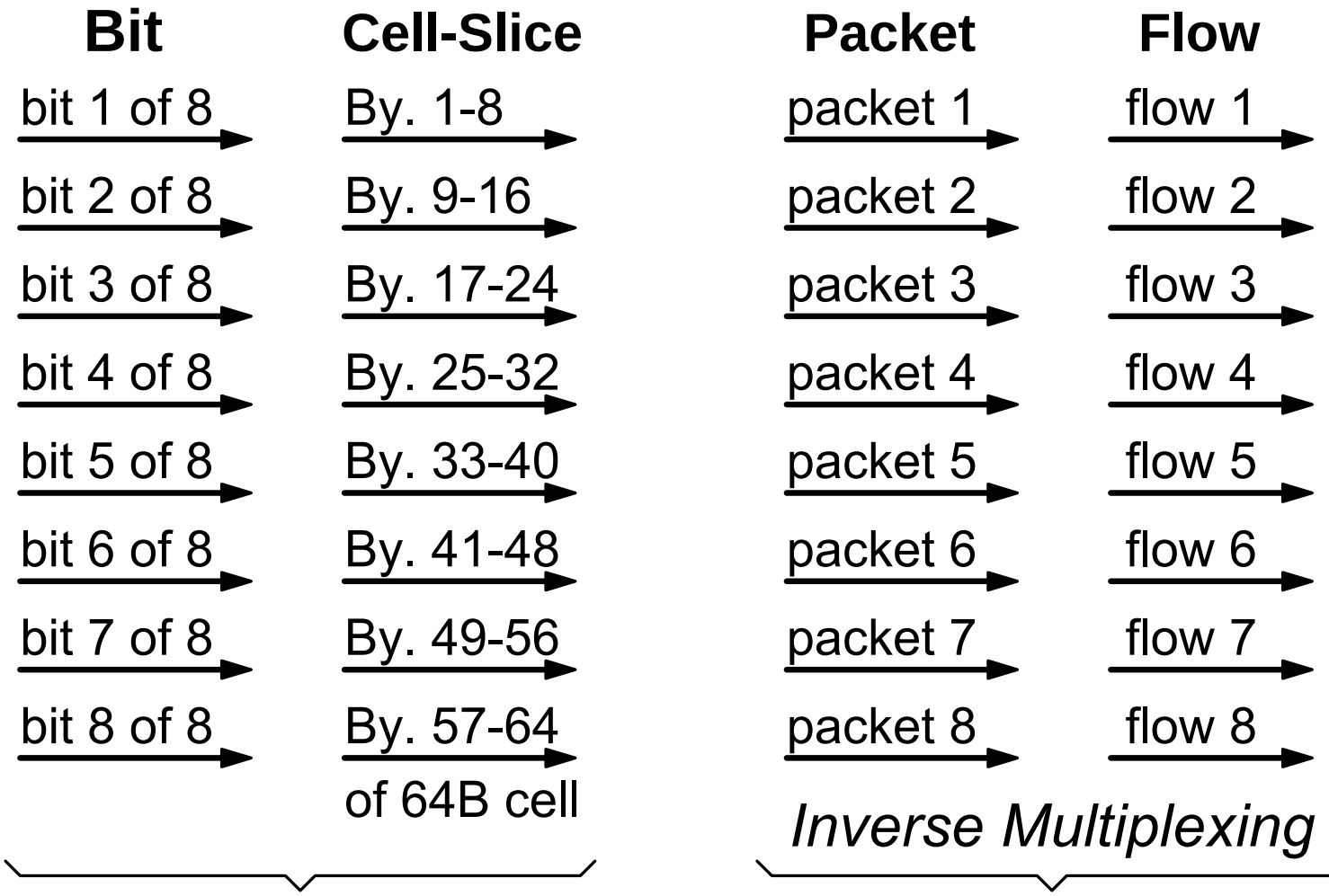
$$\text{Buffer Space} \geq |\lambda_{in} - \lambda_{out}| \cdot w_{time}$$

- Throughput Conservation Law holds in the long-run
- Time scale for "long run" is proportional to Buffer Space
- Buffer is proportional to *Burst Size*
  - *burst:* a large throughput difference that persists for certain time
- Average Delay = (Average Buffer Occupancy) / λ

  express the area between arrival and departure curves as either:
  - vertical slices: (average buffer occupancy) · (time window)
  - horizontal slices: (avg. delay) · (# of Bytes) = (delay) · λ · *w*

    (assume FIFO, but average is same under any service policy)

# Parallel Link Forms / Concepts:




same handling for all wires
(same time, same destination)

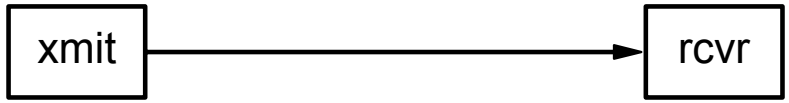
different handling
(diff. times & destinations)

# Parallelism Styles: Data-only vs. Data-and-Control

- Data-only link parallelism – SIMD Style:
  - multiple wires or sub-links carry data portions of a same packet
  - same control (routing, scheduling, contention resolution, buffering) for all wires / sub-links / portions of the packet
- Data-and-Control link parallelism – MIMD Style:
  - sub-links carry different packets each
  - control is separate for each sub-link
  - called *"Inverse Multiplexing"* – the key to scalable switching

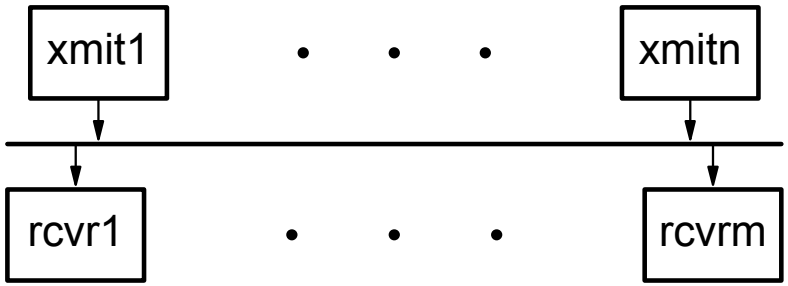
# 2.1.3 Point-to-Point versus Multi-Access Links

Point-to-Point:




(e.g. high-speed copper or fiber links)

Shared Medium:




(e.g. wireless links, old ethernet, buses)

⇒ Higher Performance:

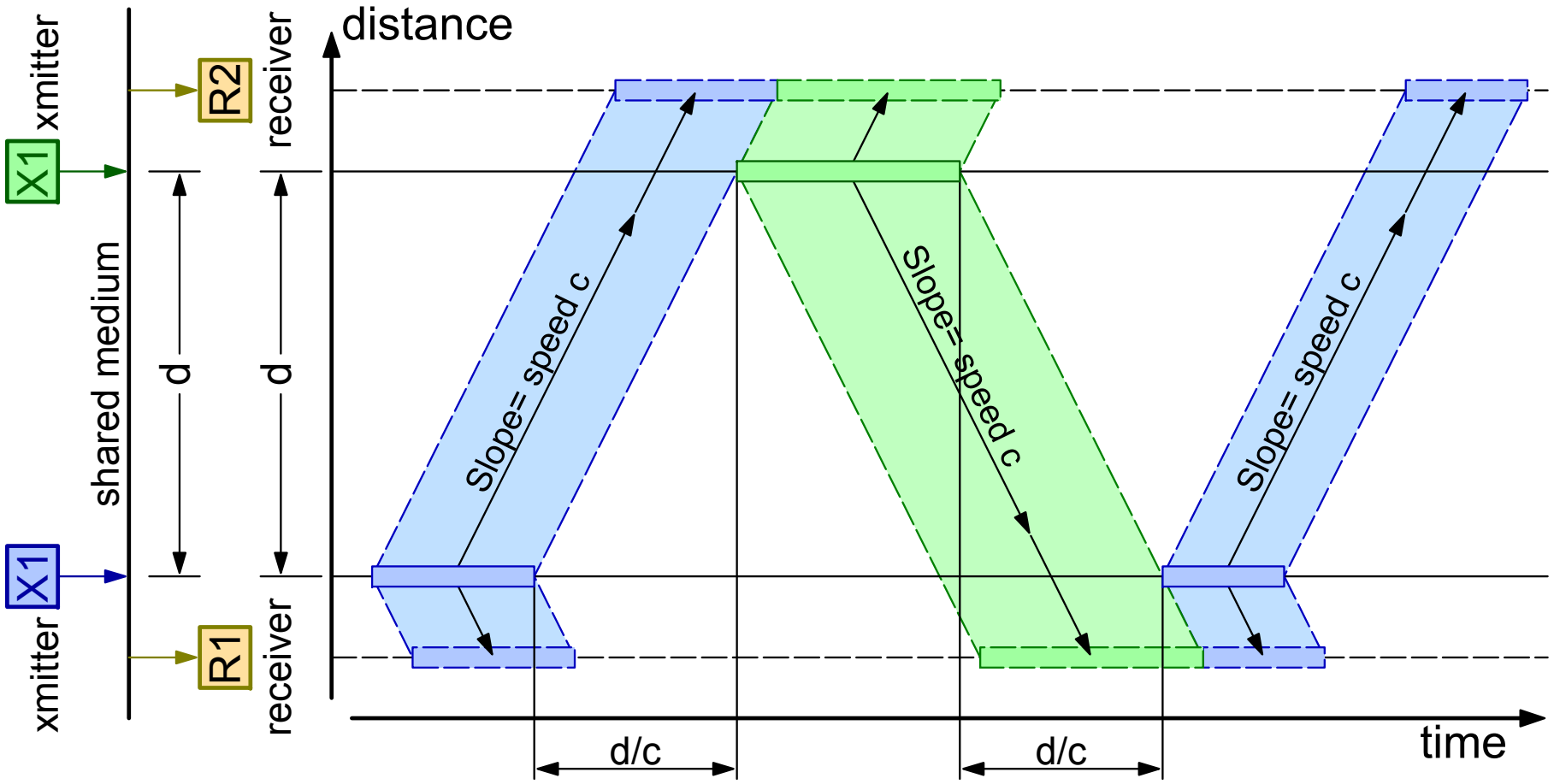
- no turn-around delay
- no arbitration overhead
- increased parallelism
  when used with switches

⇒ Lower Cost:

- broadcast & select: switching is inherent in the medium
- natural in some environments
  --e.g. wireless without directional antenas

- This course: point-to-point!

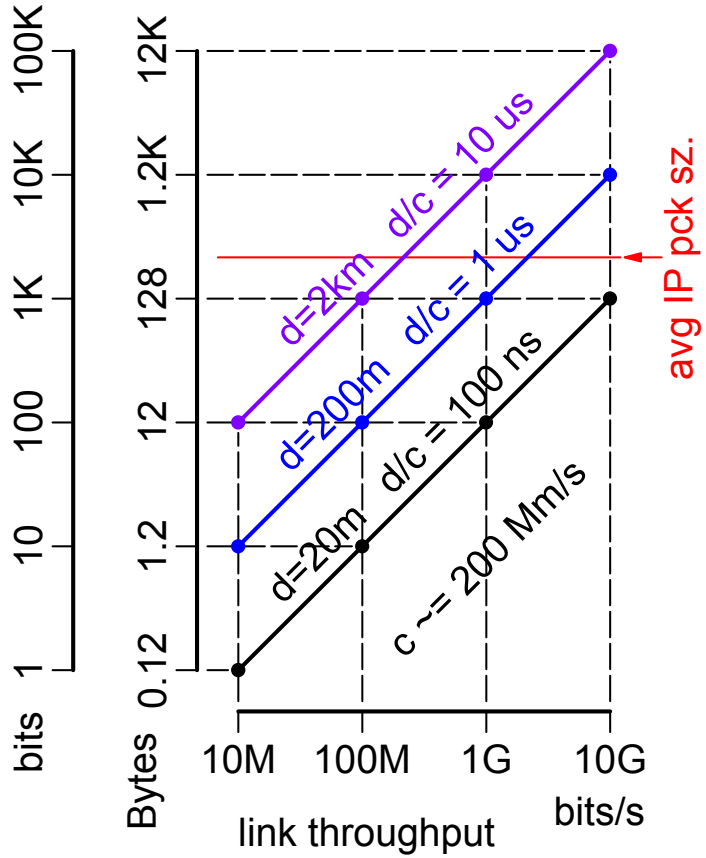
# Multiple-Access Links (Buses): Turn-Around Overhead




- For non-overlapping reception: d/c idle time on every change of transmitting station, where c = speed of light in the link medium

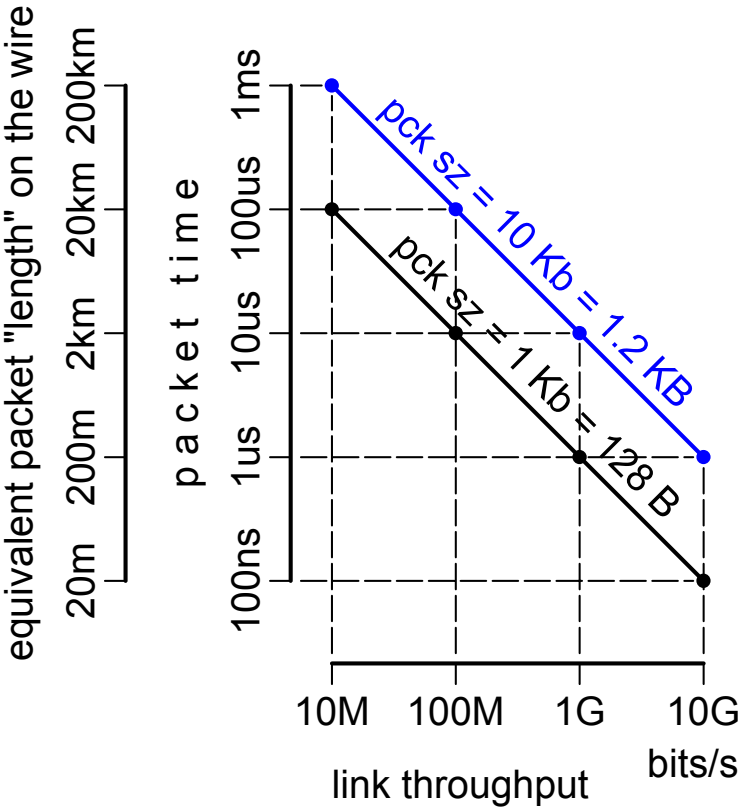
# How many bits is the length of the wire?

Turn-around delay expressed as a lost opportunity to transmit an amount of information equivalent to:



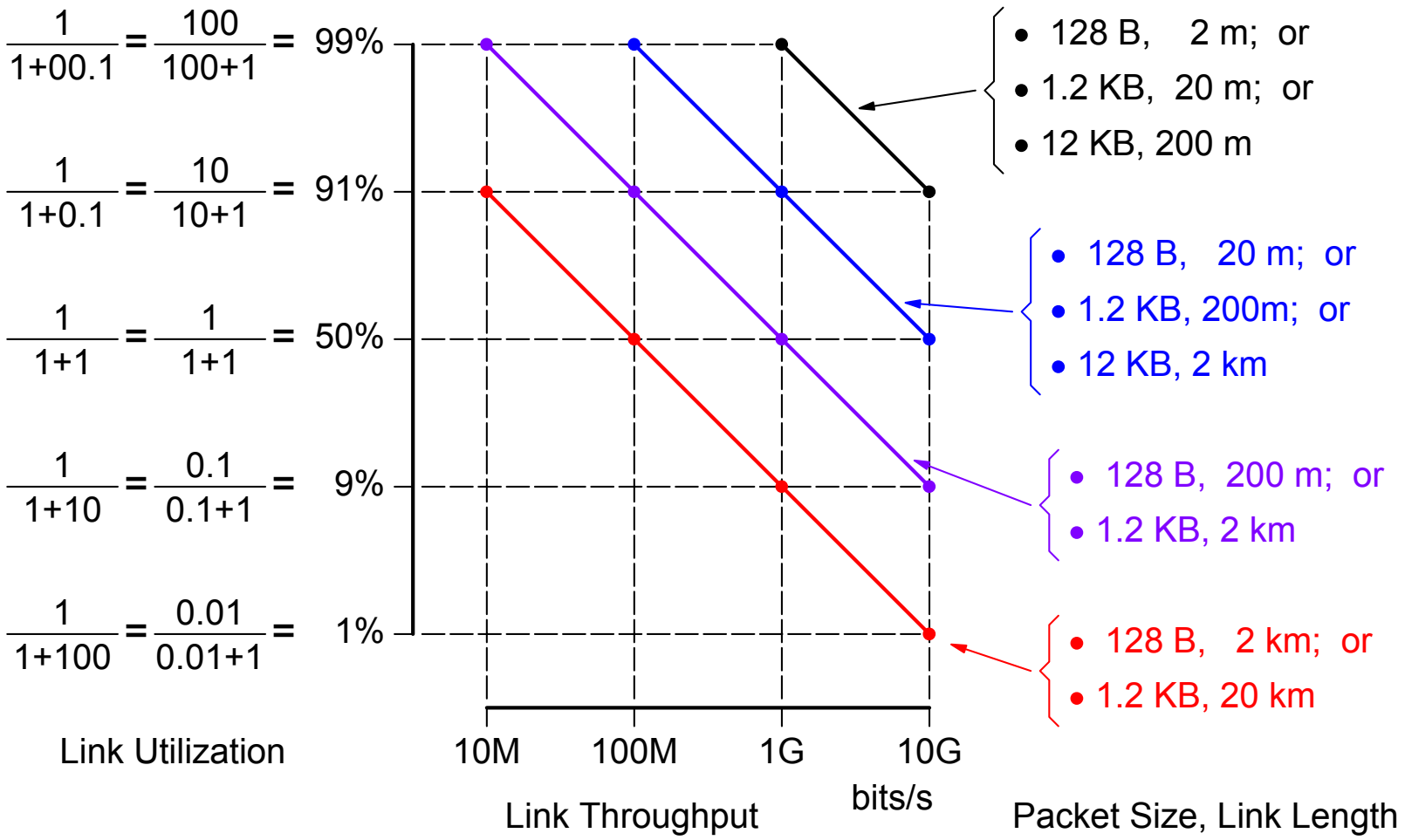

# How long (in meters) is a packet on the wire?

Duration of each transmit session (assume one packet), in time or in equivalent packet "length" on the wire:

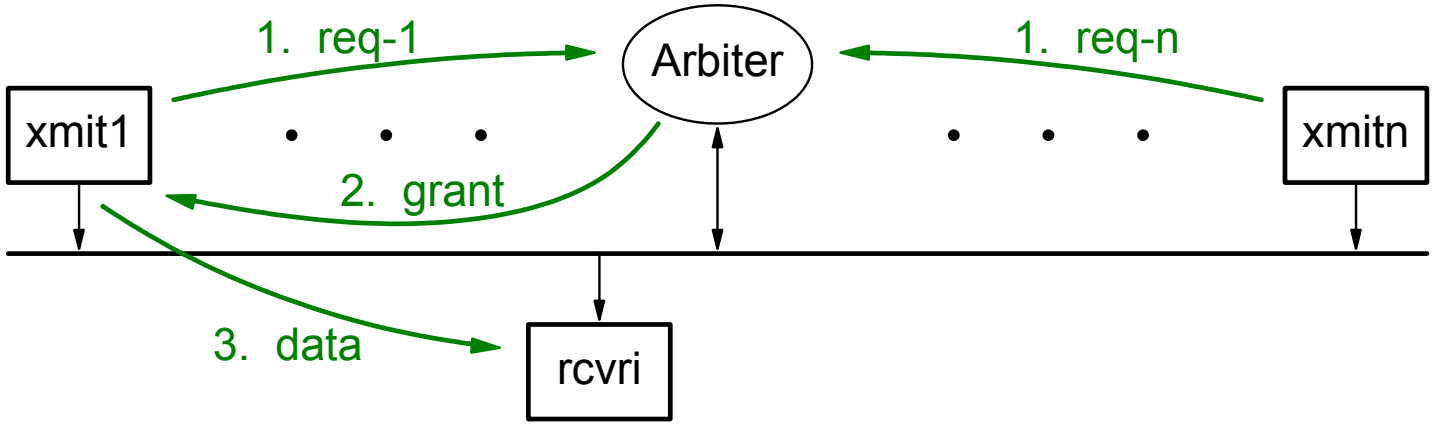


# Link Utilization = *f(* Packet Sz, Wire L, Throughput *)*

$$\text{Link Utilization} = \frac{\text{packetSize}}{\text{packetSize + turnAroundBitEquiv}} = \frac{\text{packetTime}}{\text{packetTime + turnAroundDelay}}$$



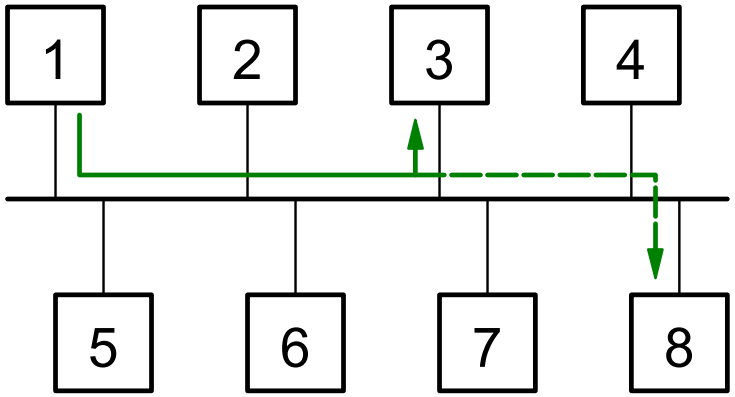
# Multiple-Access Links (Buses): Arbitration Overhead




- Separate medium for requests and grants?
  - ➞ increased media cost, increased latency.
- Shared medium for all of request, grant, and data?
  - ➞ reduced throughput, increased latency.
- Optimistic arbitration (CDMA/ethernet style) ?
  - ➞ limited peak throughput, very high latency at high loads.

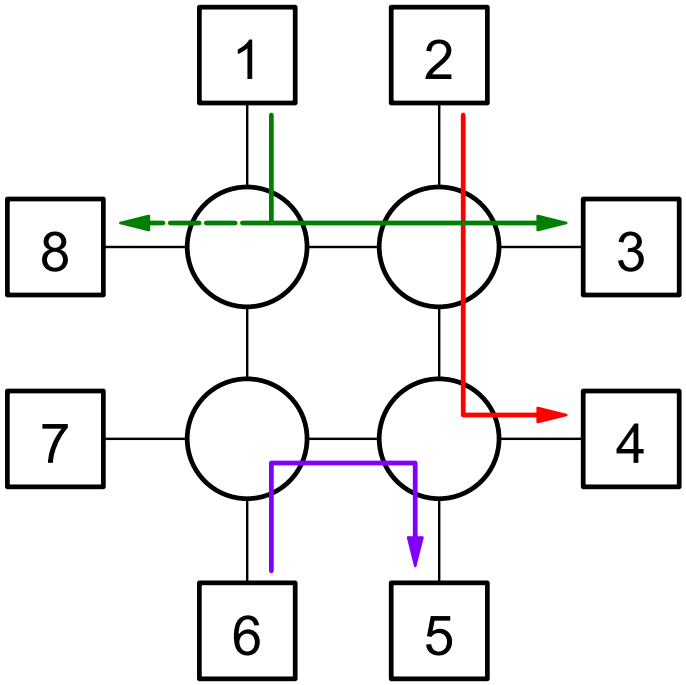
# Sequential versus Parallel Transmissions

Shared Medium:




*Single transmission at a time*

Point-to-Point Links + Switches:




*Multiple transmissions in parallel*