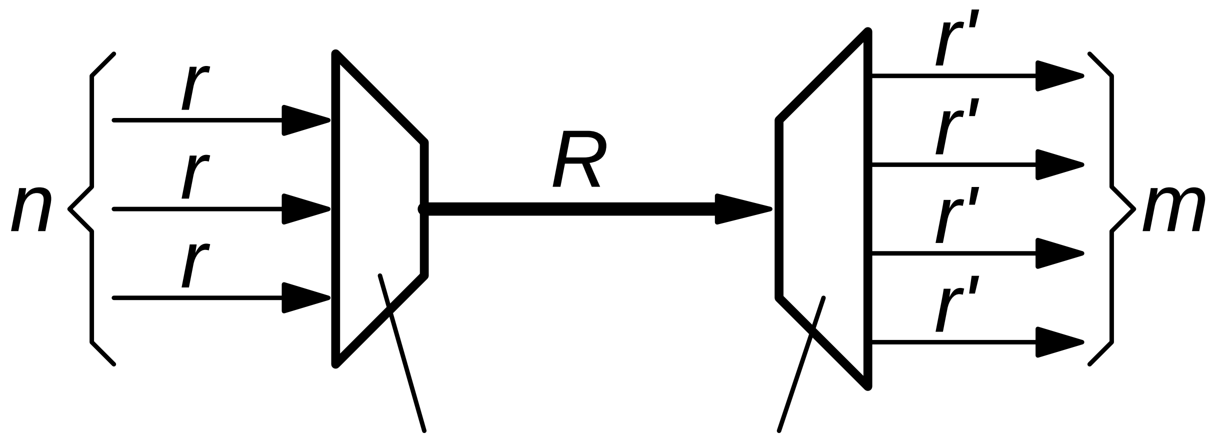
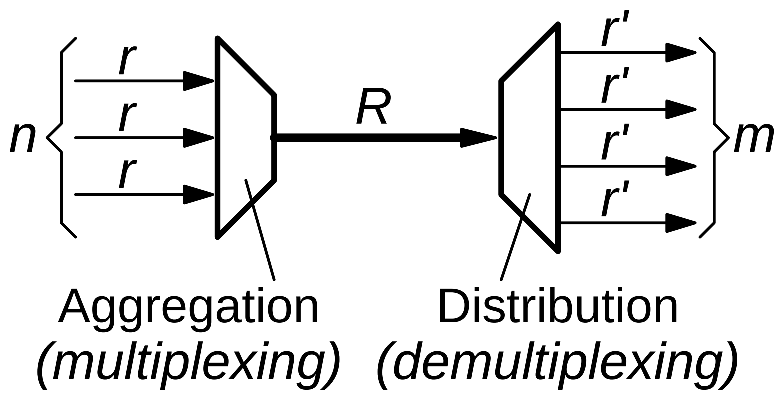
1.3 Multiplexing, Time-Switching, Point-to-Point versus Buses



Aggregation *(multiplexing)*   Distribution *(demultiplexing)*

- Simplest Networking, like simplest programming: Sequential *!*

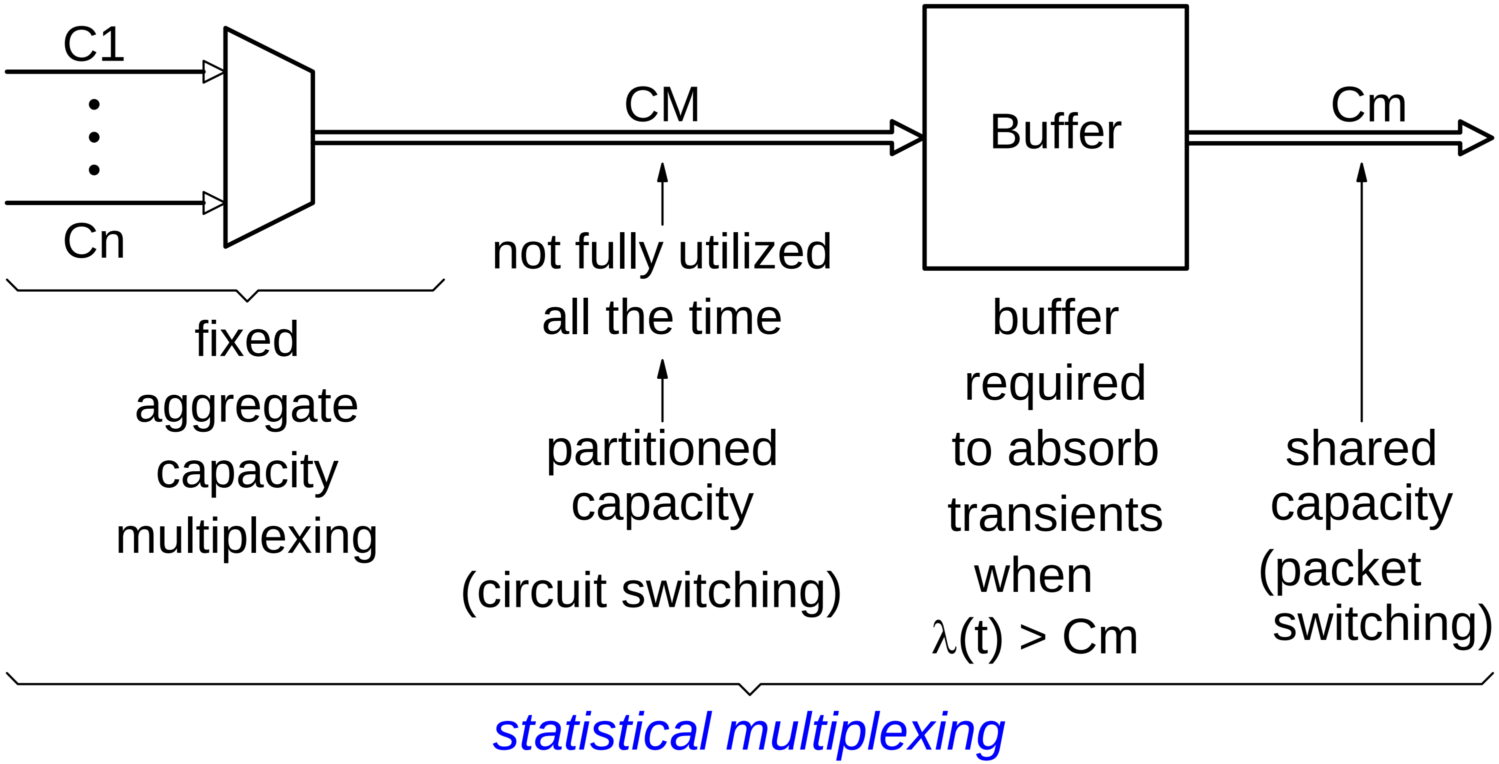
Time-Switching



- Shared Medium Communication
- Demultiplexor determines where each piece of information is routed to by selecting the *time* at which each output receives information from the shared medium
- Simplicity: one thing at a time – no parallelism
- Non-scalable! – cannot increase R indefinitely
- Full-capacity ($R = n \cdot r$) or Statistical Multiplexing ($R < n \cdot r$)
- Implementation issues, point-to-point links versus bidirectional, shared physical medium

Statistical Multiplexing

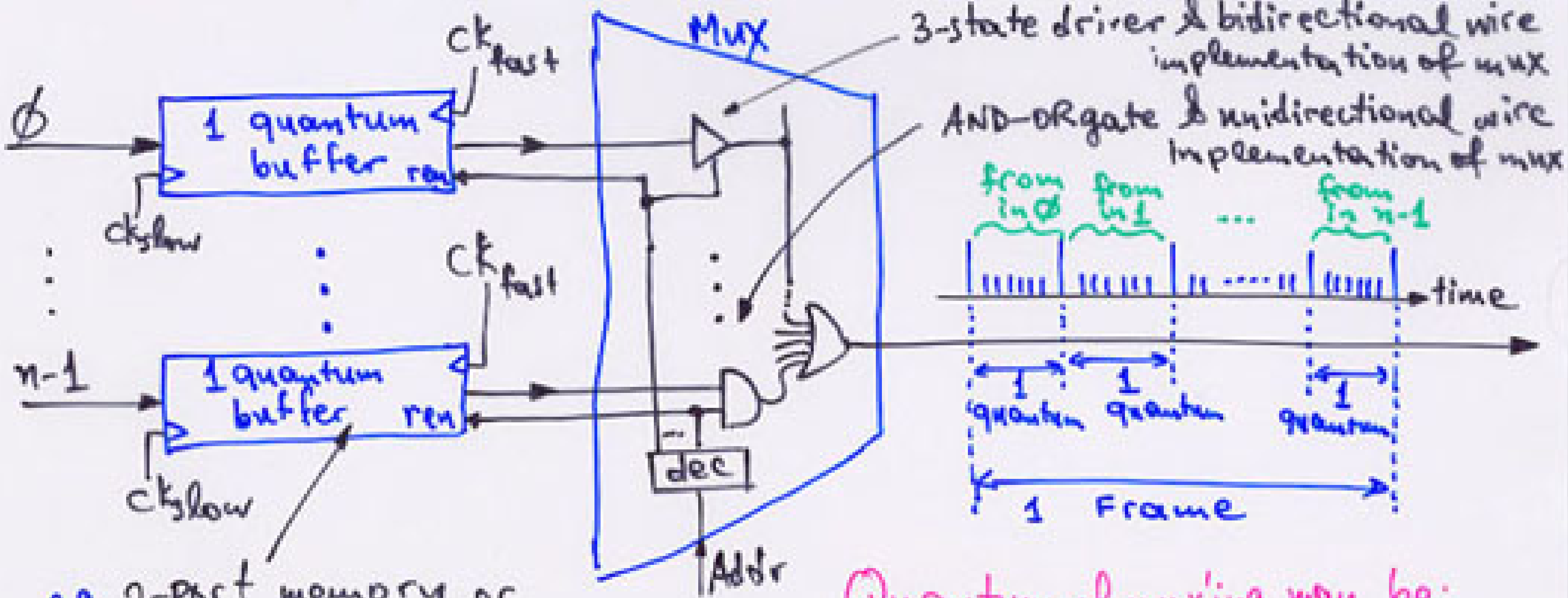
C1 + ... + Cn = CM > Cm



Capacity Cm is better utilized than capacity CM

The average aggregate source rate must not exceed Cm over time windows determined by the size of the buffer memory

Same-Width (per link), Faster-Clock Multiplexor:



e.g. 2-port memory, or two shift registers using double buffering

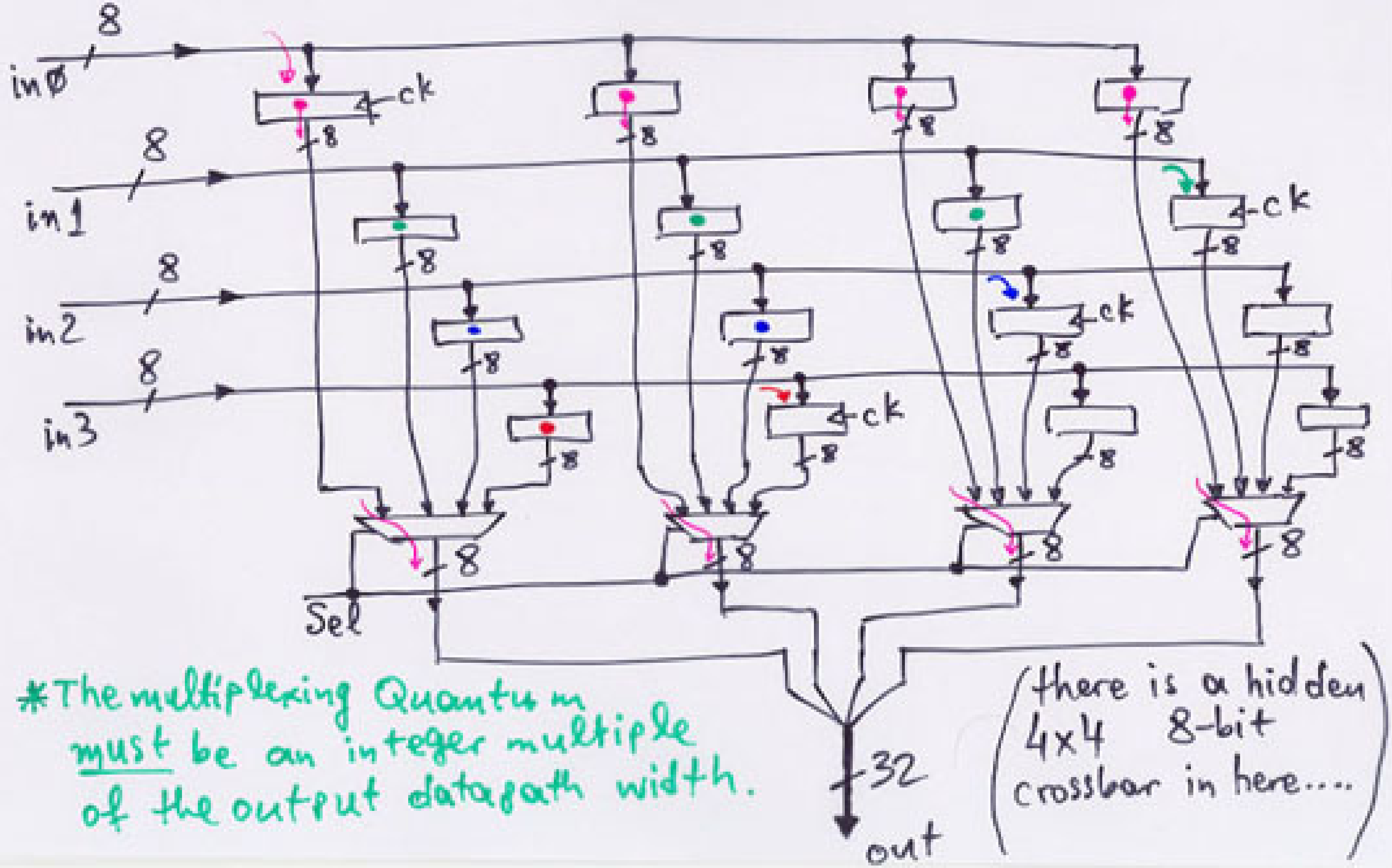
(properly "skew-synchronized" inputs ⇒ buf.size = 1 quantum; non-synchronized inputs ⇒ buf.size = 2 quanta (double buffering needed))

Quantum of mux'ing may be:

- 1 bit
  - parallel to serial converter
  - digital telephony – DS2, DS3
- 1 Byte
  - digital telephony – {DS1, SONET}
- 1 Cell or 1 Block
  - ATM
  - bus-oriented switches
  - other (high-throughput) switches, internally
- 1 Packet
  - packet switching

Same-Clock, Wider-Datapath Multiplexor:

Example: 4-to-1 mux, 8-bit inputs, 32-bit output, common clock
multiplexing quantum = 32 bits, skew-synchronized inputs

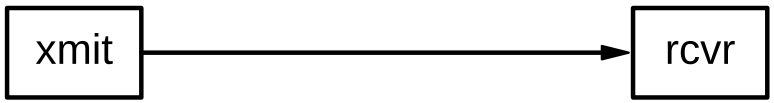


* The multiplexing Quantum must be an integer multiple of the output datapath width.

(there is a hidden 4x4 8-bit crossbar in here....)

Point-to-Point versus Shared Bidirectional Links

Point-to-Point:

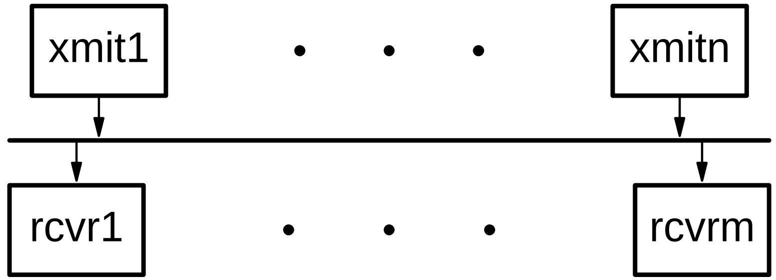


(e.g. high-speed copper or fiber links)

⇒ **Higher Performance:**

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

Shared Medium:

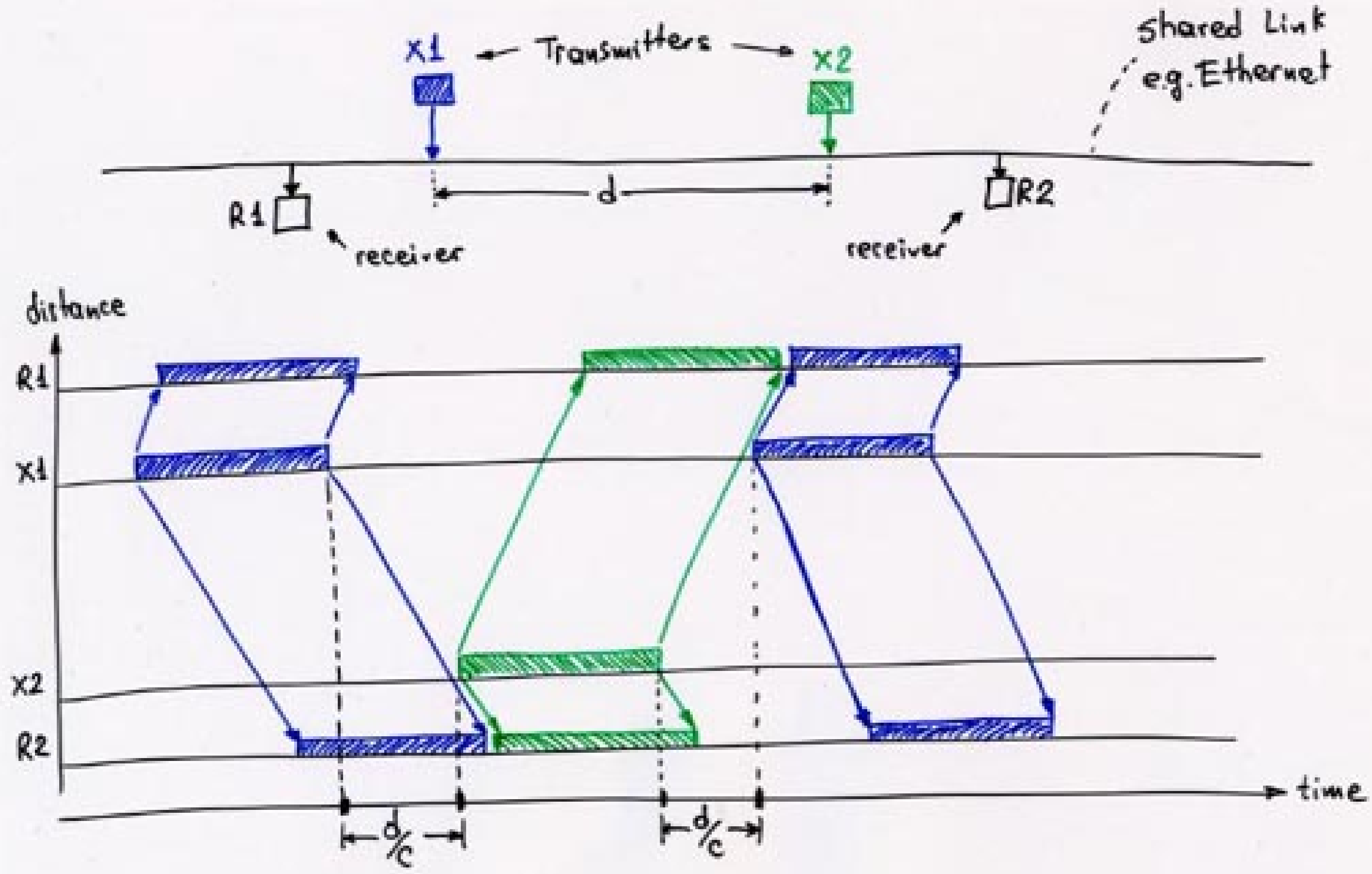


(e.g. wireless links, old ethernet, buses)

⇒ **Lower Cost:**

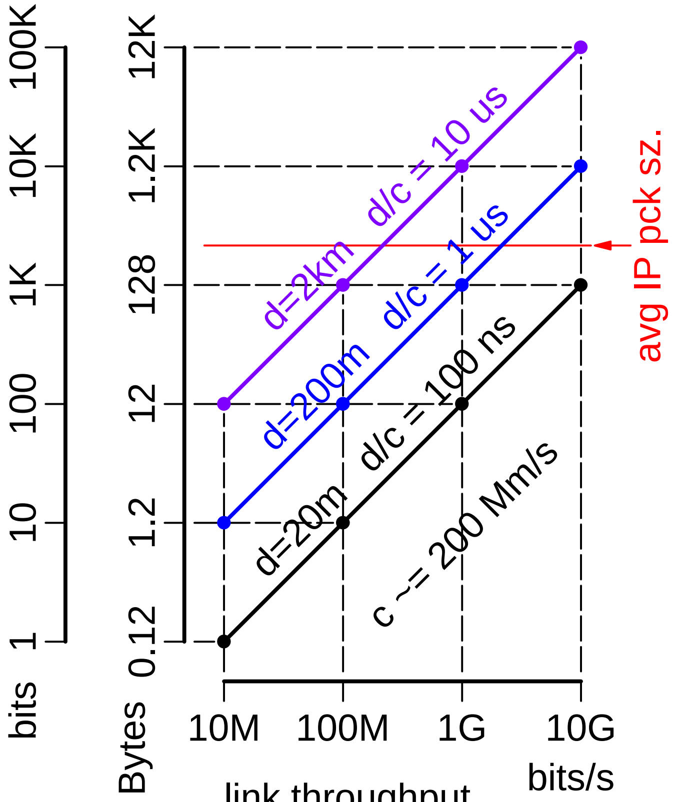
- broadcast & select: switching is inherent in the medium
- natural in some environments --e.g. wireless without directional antenas

Shared Medium Link: Turn-Around Overhead



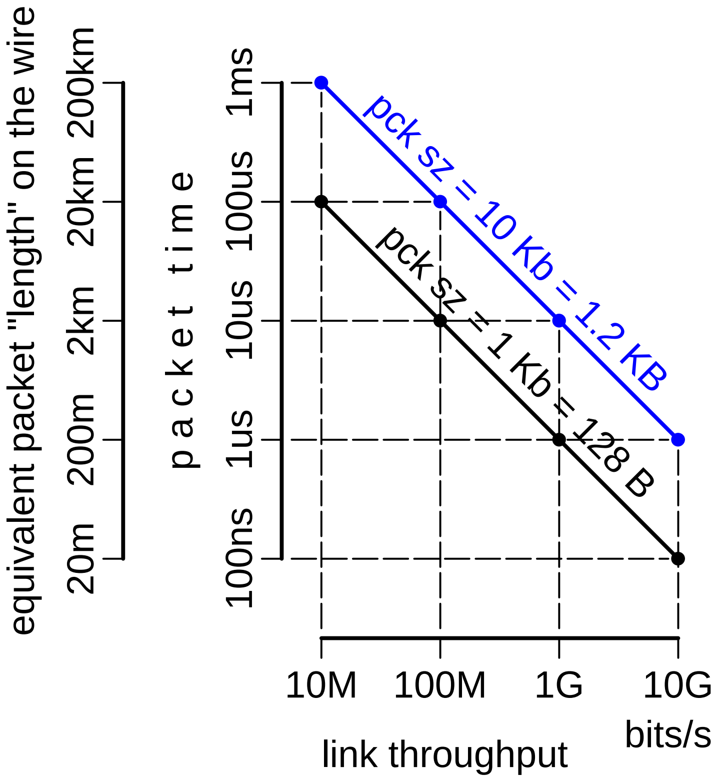
How many bits is the length of the wire?

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



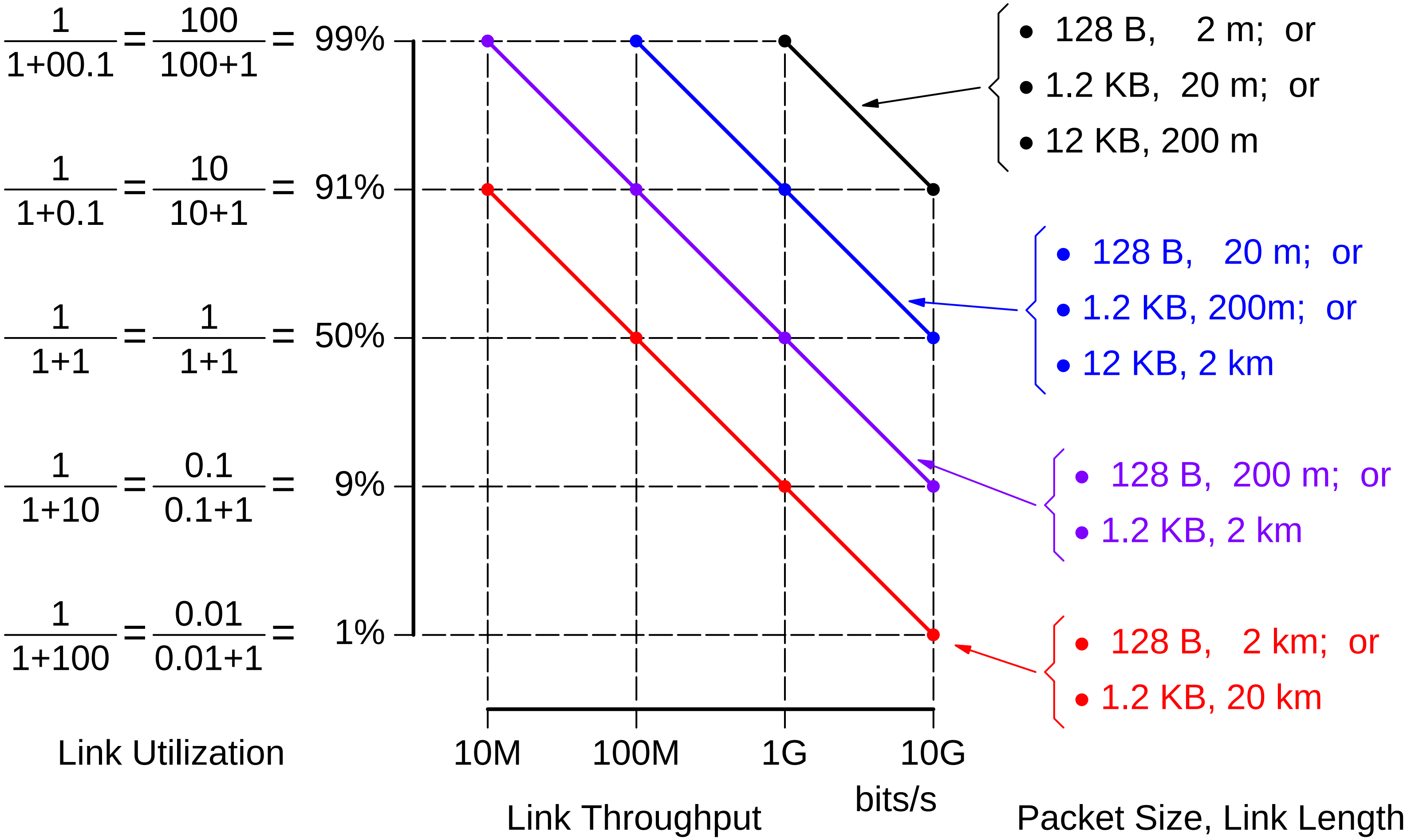
How long is a packet on the wire?

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

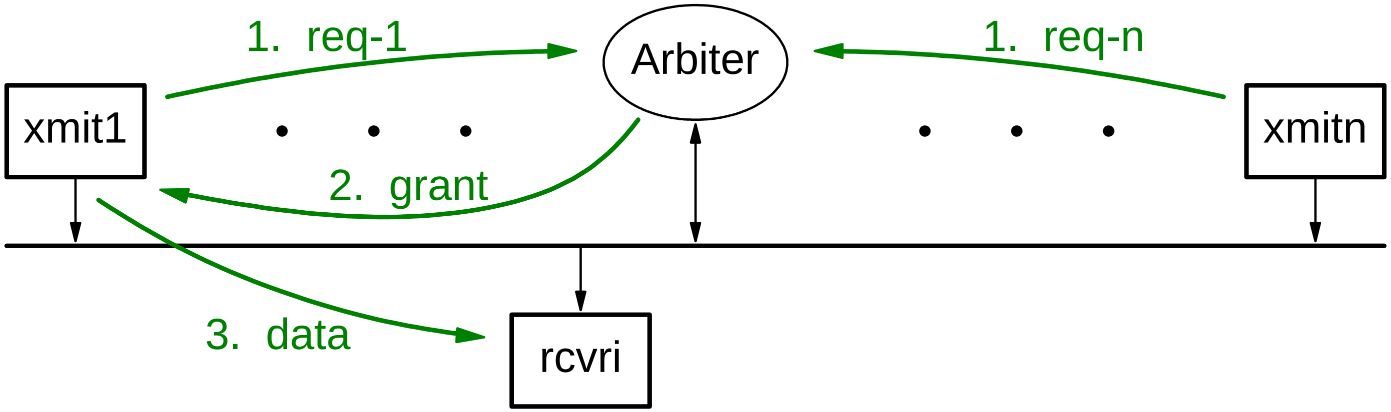


Link Utilization = f(Packet Length, Wire Length, Throughput)

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



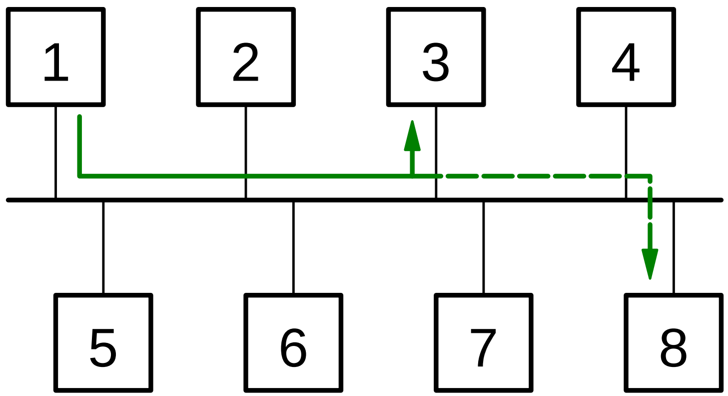
Arbitration Overhead in Shared Media



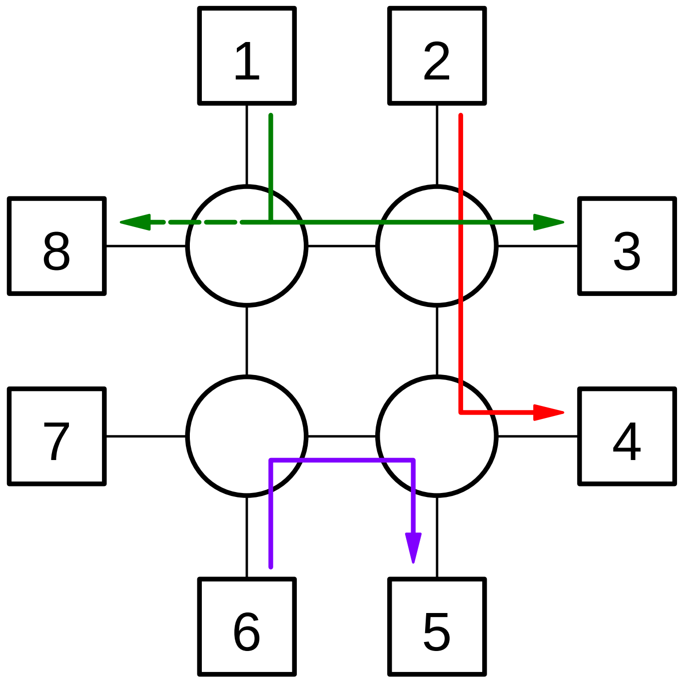
- Separate medium for requests and grants?
  - → increased media cost, increased latency.
- Shared medium for all of request, grant, and data?
  - → reduced throughput, increased latency.
- Optimistic arbitration (CDMA/ethernet style) ?
  - → limited peak throughput, very high latency at high loads.

Sequential versu Parallel Transmissions

Shared Medium:



Point-to-Point Links + Switches:



Single transmission at a time

Multiple transmissions in parallel

This Course: Point-to-Point Links

- Throughput in shared media is rather low
  - time switching (simpler), protocols usually in software, with little, if any, & simple H/W support
- Shared media in modern networking: almost expelled, except in wireless
  - wireless MAC protocols are a major topic in other courses,and differ significantly from the hardware architecture of high-speed switches