Understanding QoS

On the physical layer

No configuration examples
No vendors

What is QoS

- Quality of Service is the control of:
  - Delay
  - Jitter
  - Bandwidth
  - Packet Loss

- To understand QoS we first have to understand these parameters

- That is what I will talk about today

Uses of QoS and their goals

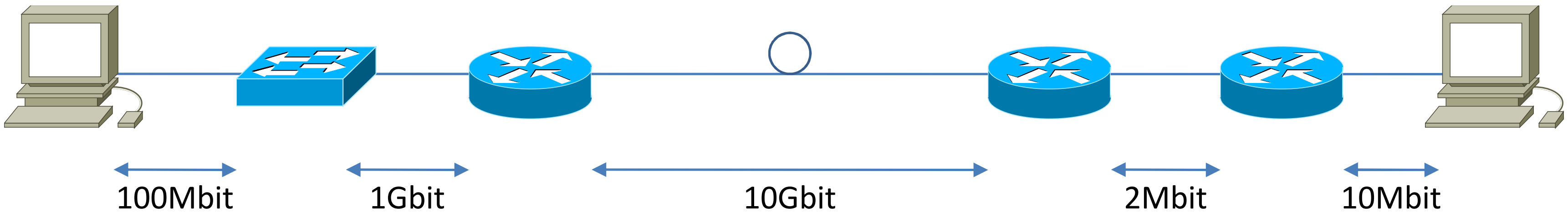
- Depending on the application of QoS we want to control different parameters

- Common examples:
  - Voice over IP → Jitter & Delay
  - Video Conferencing → Jitter & Bandwidth
  - SAP Session → Bandwidth
  - Remote Desktop → Bandwidth
  - Multiplayer Online Games → Delay & Jitter
  - …

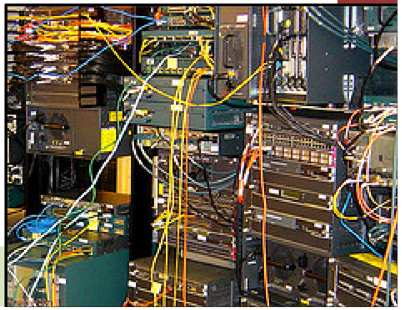
The Parameters

- Delay
  - Is the time it takes for a packet from source to destination
- Jitter
  - Is the variance in delay between successive packets
- Bandwidth
  - Is the amount of network resources allocated (min/max) for a certain application
- Packet Loss
  - Packets that are dropped by the network

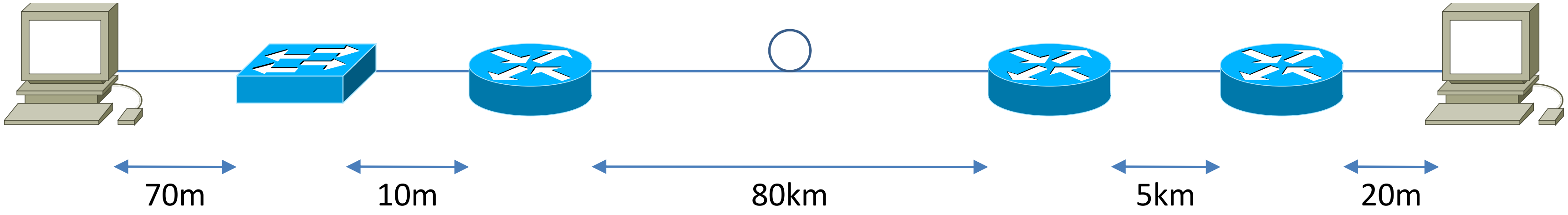
Example Network



- We will use this network as our example:
  - We have a source (left) and destination (right)
  - Physical bandwidth from 2Mbit to 10Gbit
  - One Ethernet Switch
  - Three Routers
  - Five Links

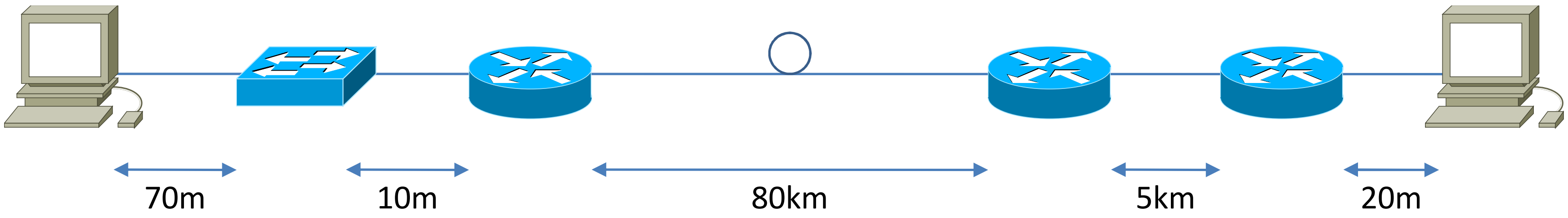


Delay (1): Propagation



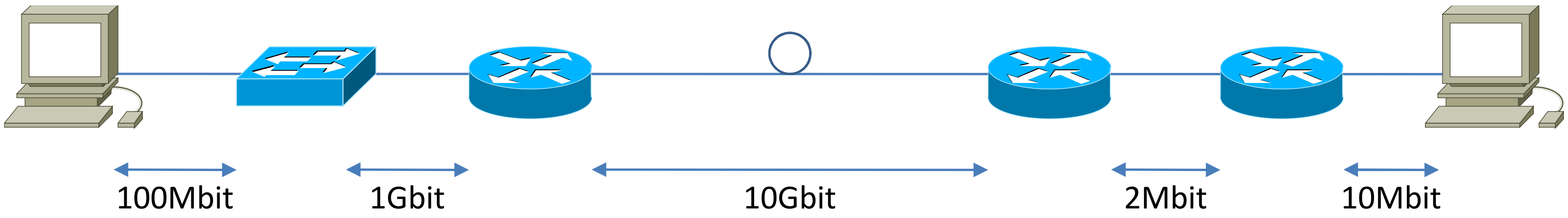
- The first part is propagation delay:
  - We can‘t be faster than speed of light (3.334ns/m)
  - Copper is about 4.7ns/m
  - Fiber is about 5ns/m (yes, it‘s a bit slower)

Delay (1): Propagation



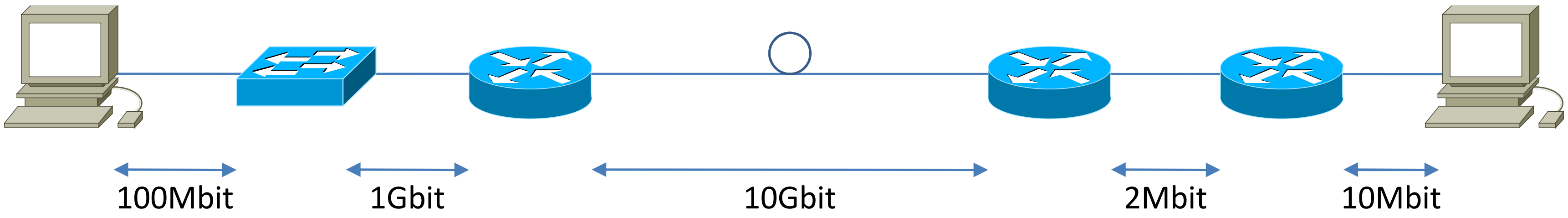
- Lets put this together
  - Copper:
    - 70m + 10m + 5km + 20m = 5‘100m
    - 5‘100m * 4.7ns/m = 23.97µs = *0.024ms*
  - Fiber:
    - 80‘000m * 5ns/m = 400µs = *0.4ms*
- Total: *0.424ms* Propagation Delay

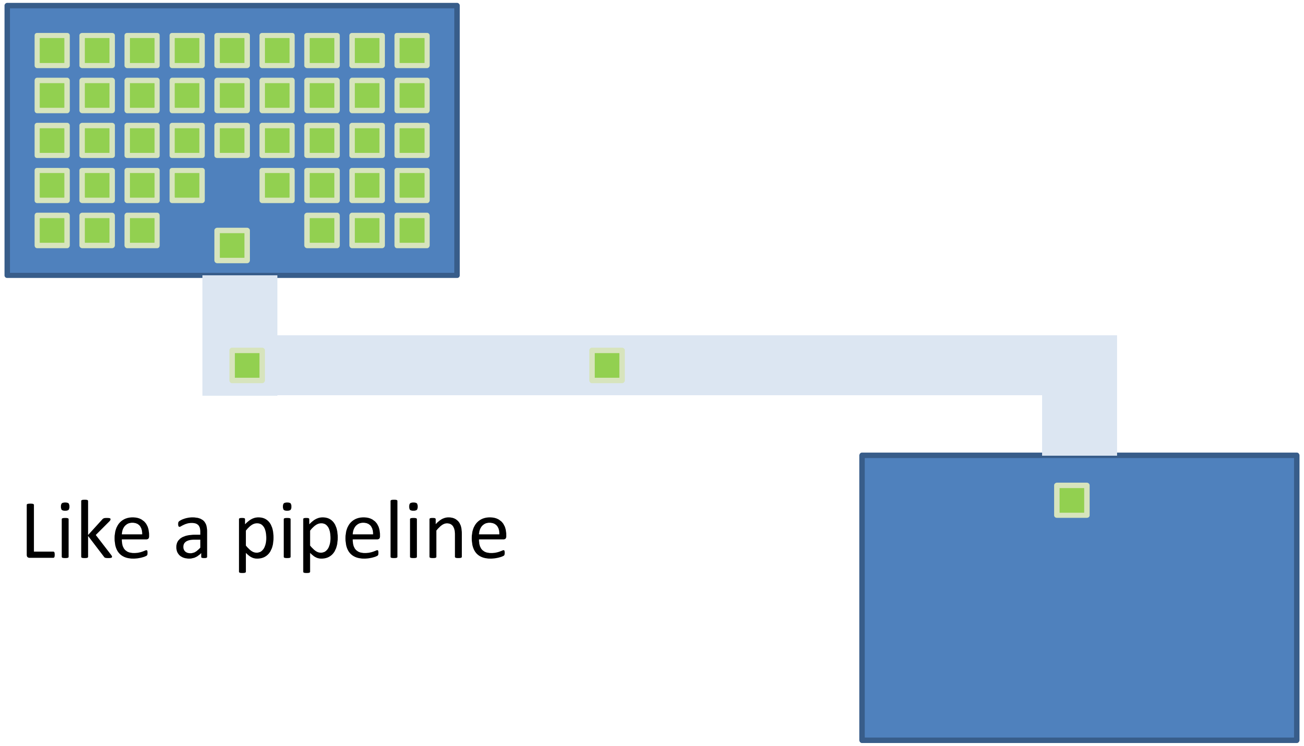
Delay (2): Serialization



- The second part is serialization delay:
  - The time it takes to put the packet bit for bit onto the wire from memory
  - And back again…
  - Formula: Packet size / Link speed

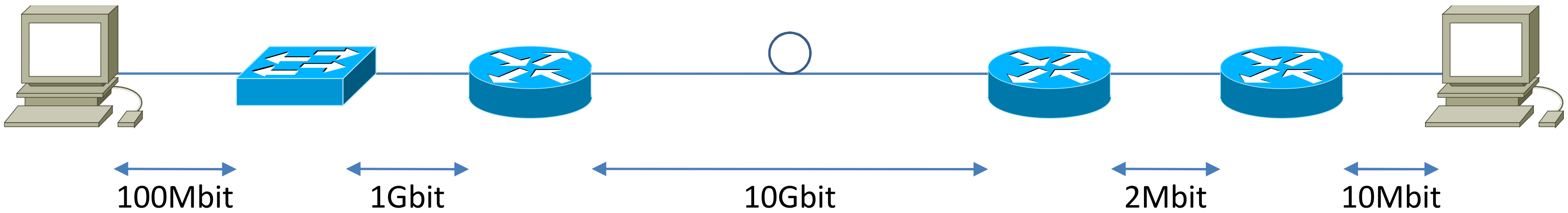
Delay (2): Serialization



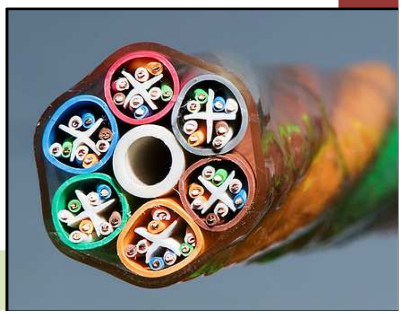


- Like a pipeline

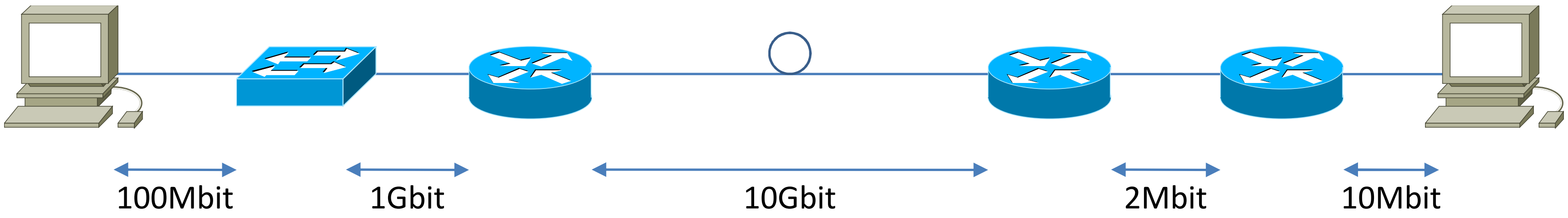
Delay (2): Serialization



- Lets put this together for a 1‘500 Byte Packet:
  - 2 Mbit/s ⇒ 250KB/s ⇒ 1.46KB / 250KB/s = *5.86ms*
  - 10Mbit ⇒ 1.192MB/s … *1.2ms*
  - 100Mbit ⇒ 11.92MB/s … *0.12ms*
  - 1Gbit ⇒ 119.2MB/s … *0.012ms*
  - 10Gbit ⇒ 1‘192MB/s … *0.0012ms*
- Total: *7.1932ms*

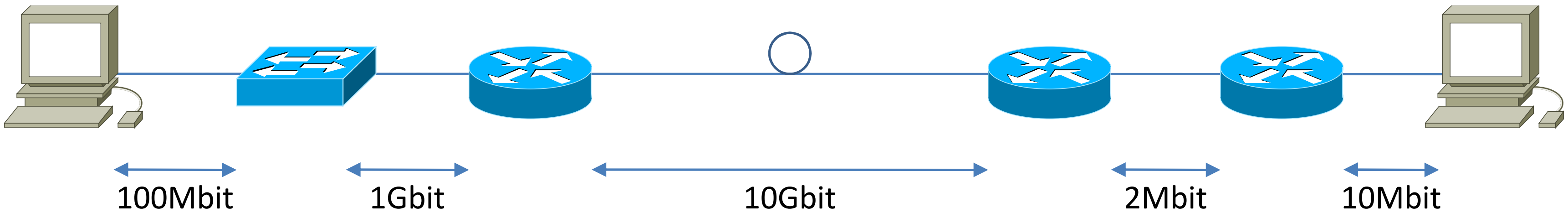


Delay (2): Serialization



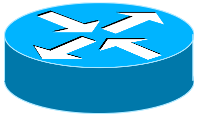
- Some additional details:
  - L2 header overhead (ethernet headers, L2TP, …)
  - Framing overhead (HDLC, ATM, …)
  - Line Encoding overhead
    - ADSL interleaving, fast-path
- May reduce effective bandwidth
- May add significant delay

Delay (3): Processing



- The third part is the processing delay in a network device
  - A switch has to look up the Layer2 MAC address to find the output interface
  - A router has to look up the Layer3 IP address to find the output interface (plus ARP table)
  - Both take some amount of time…

Delay (3): Processing



```
0024.14da.a9d4  g0/9
0019.2f40.ca9b  g0/2
0023.5e20.4588  g0/5
0023.5e53.eb52  g0/5
001e.7a3e.13c0  g0/3
0000.0c0f.4c4c  g0/1
00a0.c5db.2673  f1/1
..............  ....
```

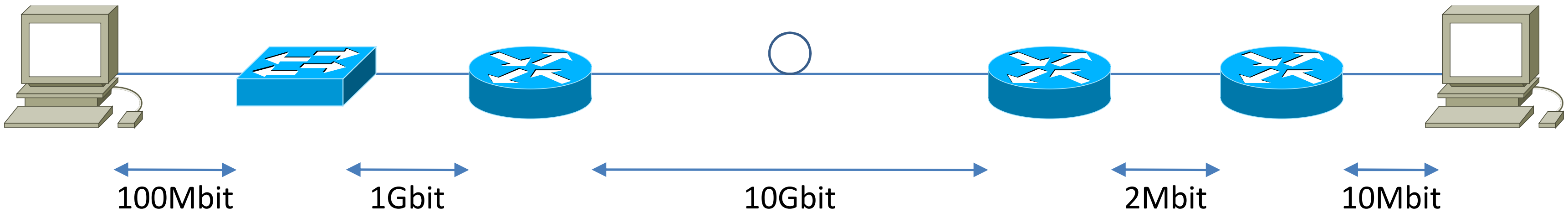
```
62.45.64.0/19   192.168.1.2 g0/1
62.45.128.0/17  192.168.1.3 g0/1
62.48.0.0/19    192.168.1.1 g0/1
62.48.32.0/19   192.168.1.4 g0/1
62.48.43.0/24   192.168.1.3 g0/1
62.48.58.0/23   192.168.1.3 g0/1
62.48.64.0/19   192.168.1.2 g0/1
..............  ........... ....
```

```
0000.0c0f.4c4c  192.168.1.1 g0/1
001e.7a3e.13c0  192.168.1.2 g0/1
0023.5e53.eb52  192.168.1.3 g0/1
00a0.c5db.2673  192.168.1.4 g0/1
..............  ........... ....
```

Perfect match lookup

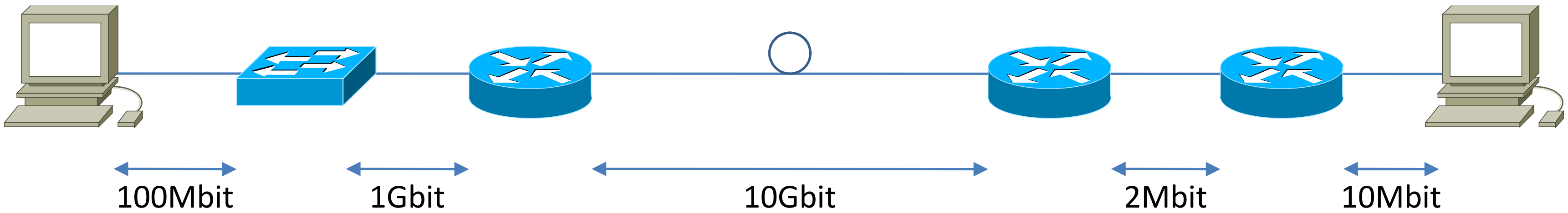
Longest prefix lookup + ARP

Delay (3): Processing



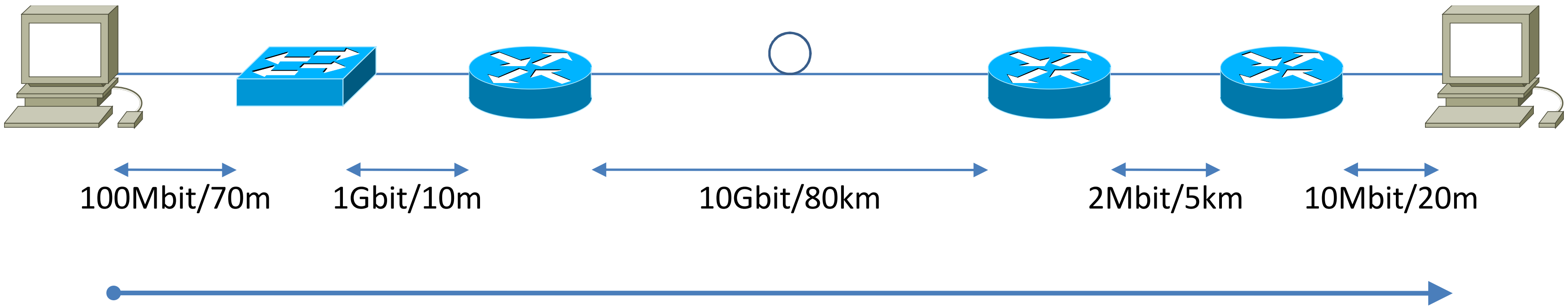
- 0.5-20µs for hardware switching/routing
- 1-100µs for software routing (variable)
- More features mean more delay
  - ACL on Layer 2-4
  - uRPF
  - Encapsulation (L2TP, PPP, MPLS, …)
  - Firewall

Delay: Summary



- Propagation delay is constant and always the same
- Serialization delay depends on the packet size but is constant for a given size
- Processing delay is almost constant and depends on the configured features

Delay: Summary

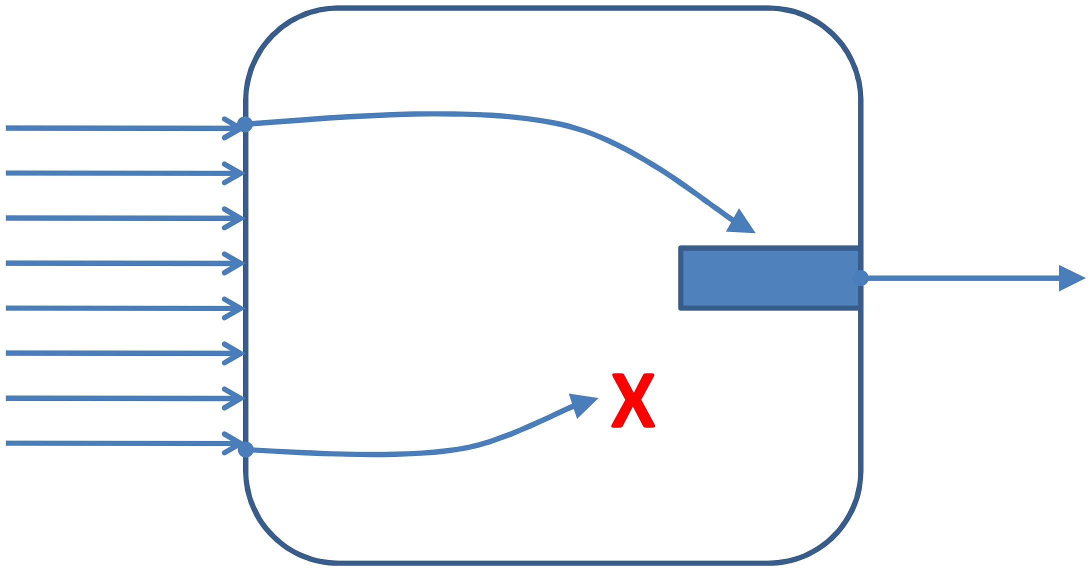


- Propagation delay 0.424ms
- Serialization delay 7.193ms (1‘500B)
- Processing delay <u>0.080ms</u> (20µs*4)

- Total delay one-way *7.697ms*
  - Total delay 64Byte *0.804ms*

Jitter (1): Buffers

- If more than one packet arrives at the same time (or one is still in serialization)…

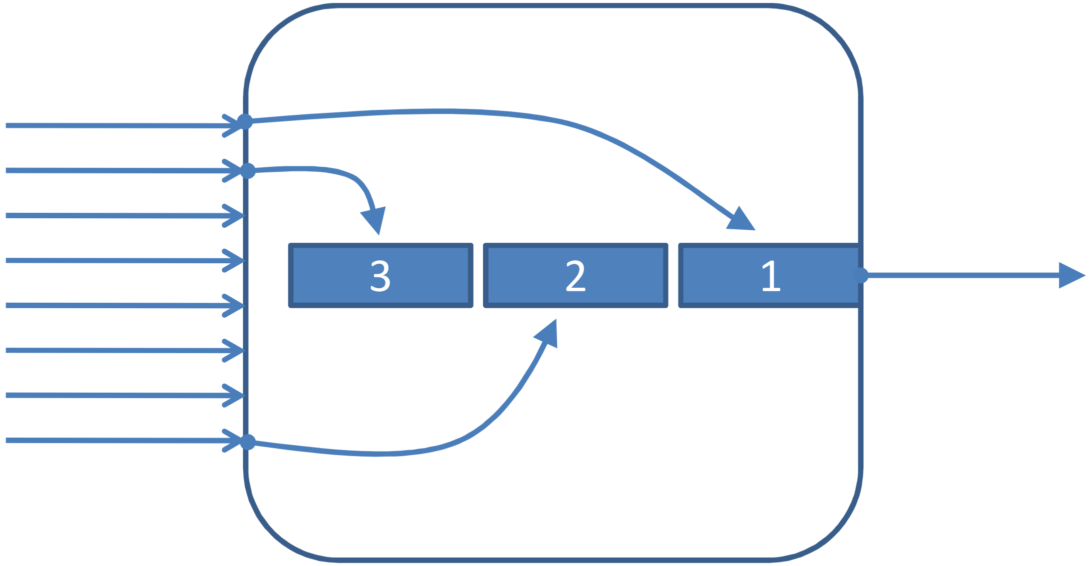


… we have a problem…

- Used to be called a collision in old Ethernet

Jitter (1): Buffers

- Switches and Routers must have buffers to temporarily store multiple packets for the same output interface



- The buffer is organized as a FIFO queue

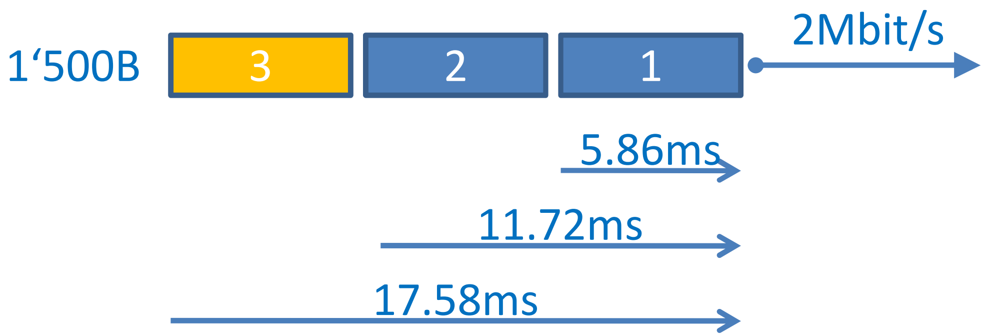
Jitter (2): Buffers & Queues

- Switches and Routers must have buffers and queues!
- Buffers store packets that can‘t be immediately sent out again when the output interface is already busy
- Queues are per output interface and organize the packets in the buffer

FIFO = First In First Out

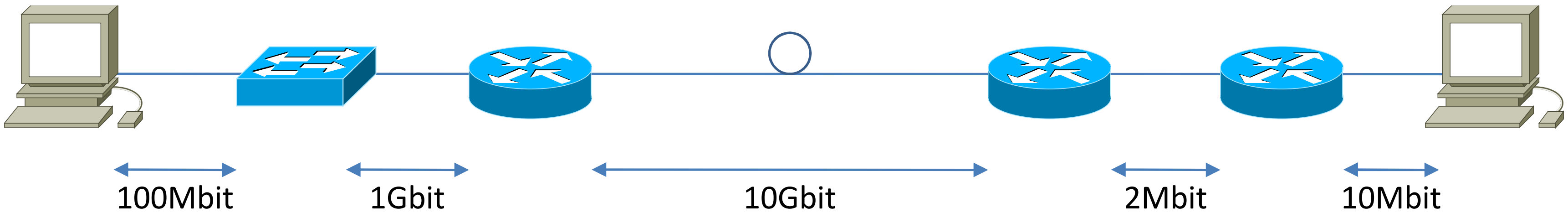
Jitter (2): Buffers & Queues

- We have a Problem again
  - The position of any packet in the queue is completely random
  - There may be a number of other packets before the important one



  - From the queuing we get serialization delay, also called *jitter*

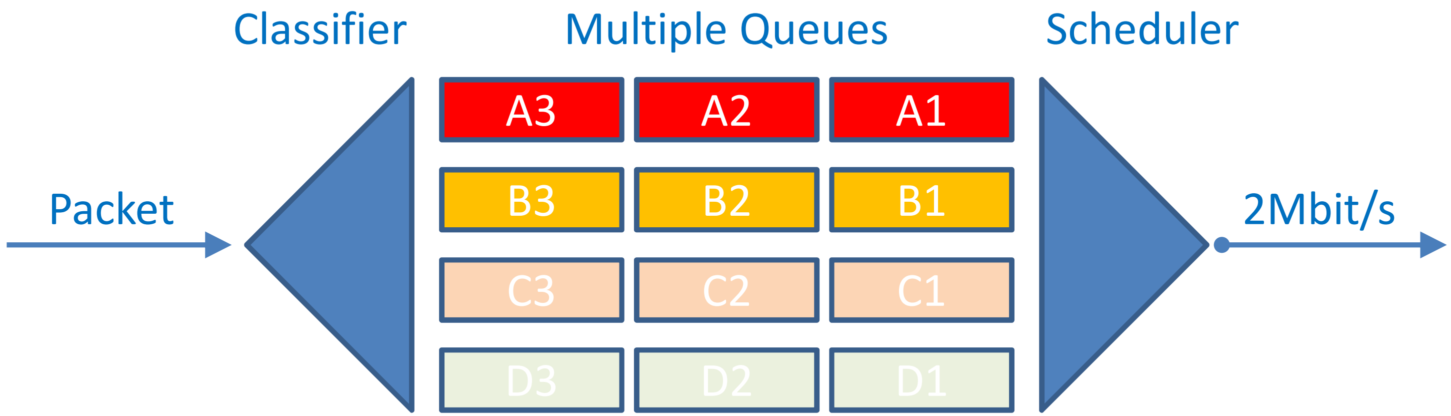
Jitter (2): Buffers & Queues



- Two types queuing effects exist:
  - Micro-peaks – Primary source is serialization delay
  - Congestion – Simply too much traffic for link speed
- Depending on traffic load we get undeterministic queueing delays
- Question: What is the main difference between Switches and Routers?
  - (Ignoring Layer2 vs. Layer3)

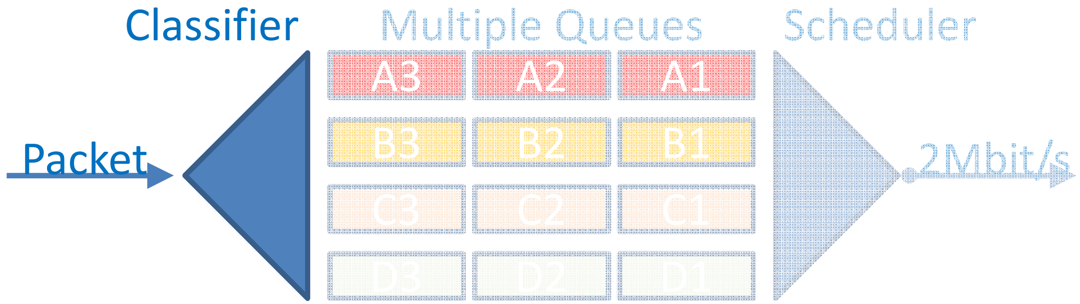
Jitter (3): Multiple Queues

- To treat packets differently based on priority we can use multiple queues



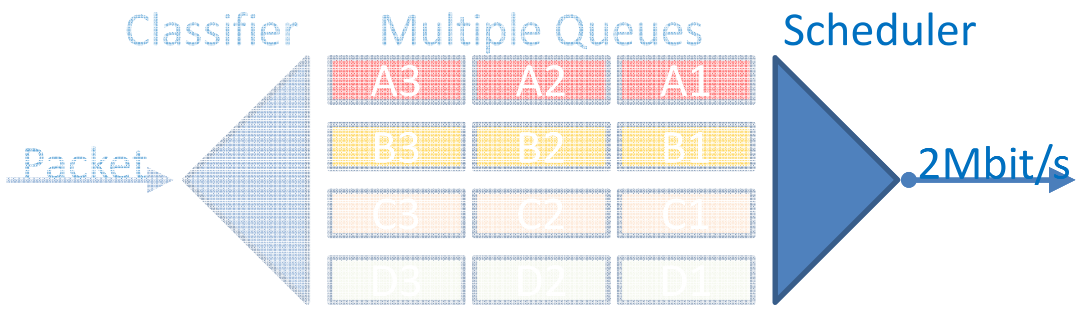
- Within each queue it is still FIFO

Jitter (3): Multiple Queues



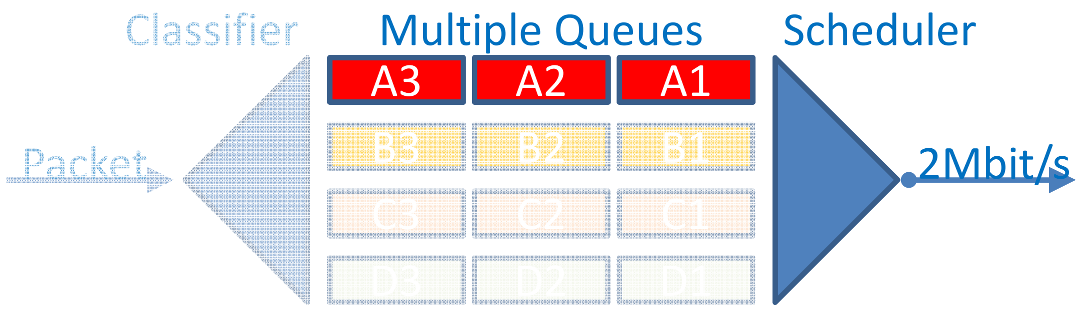
- The Classifier decides which queue a packet belongs to
- It does this based on the packet header information
  - Layer 2: 802.1p Priority bits, MPLS EXP field
  - Layer 3: IP TOS bits, IP Protocol Type, …
  - Layer 4: UDP, TCP Port numbers, …
  - Any complexity is possible (in theory)

Jitter (4): Queue Schedulers



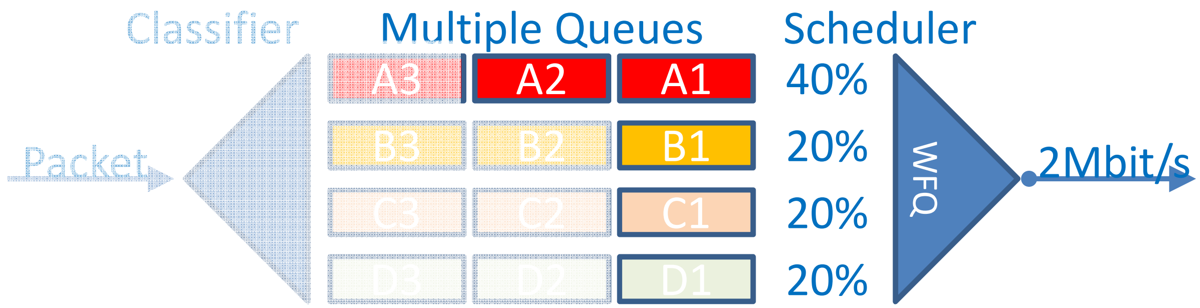
- The scheduler decides from which queue the next packet is sent
- Always send a packet from the highest priority queue if one is waiting?

Jitter (4): Queue Schedulers



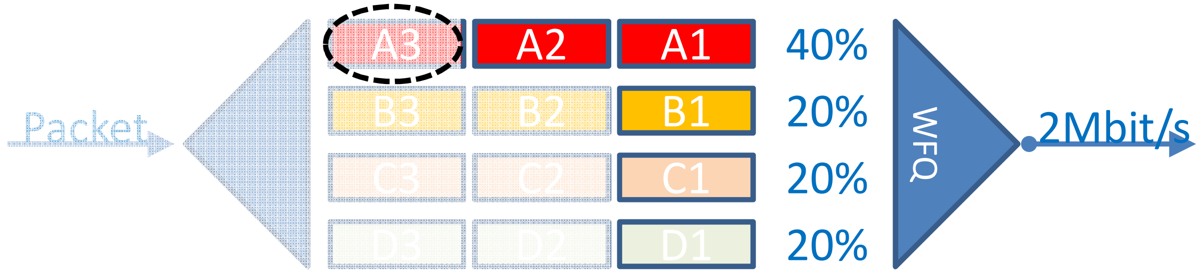
- When always the highest priority queue is served we get „head of line blocking“
  - No lower queue gets a packet out if the next higher priority queue has packets waiting
  - This a problem because lower priority traffic is starved to death
  - The link is monopolized

Jitter (4): Queue Schedulers



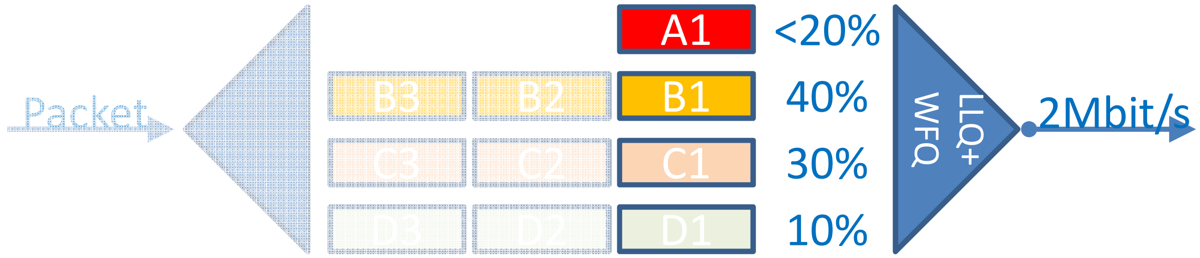
- All queues must be served
  - But not equally
- WFQ is much better (Weighted Fair Queuing)
  - Each queue gets a priority assigned
  - Normally a percentage of the link speed
  - No queue is starved to death
  - Unused bandwidth is shared up and down

Jitter (4): Queue Schedulers



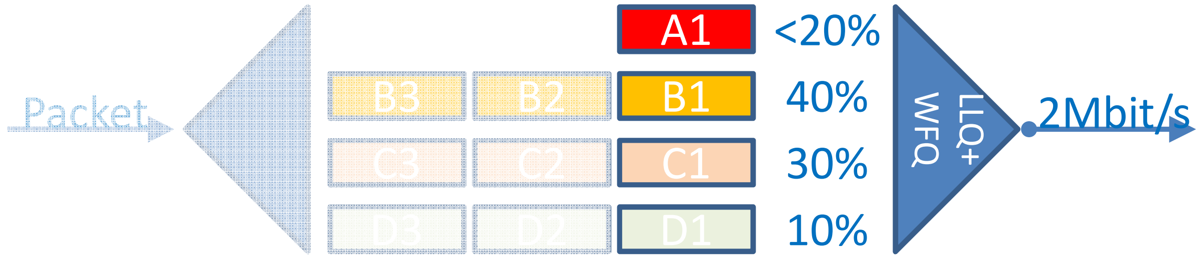
- WFQ is good
  - WFQ ensures weighted fair sharing between queues
- But not good enough for jitter sensitive real-time traffic (VoIP)
  - Even in the highest priority queue a packet may have to wait for lower priority queues to get their fair share

Jitter (4): Queue Schedulers



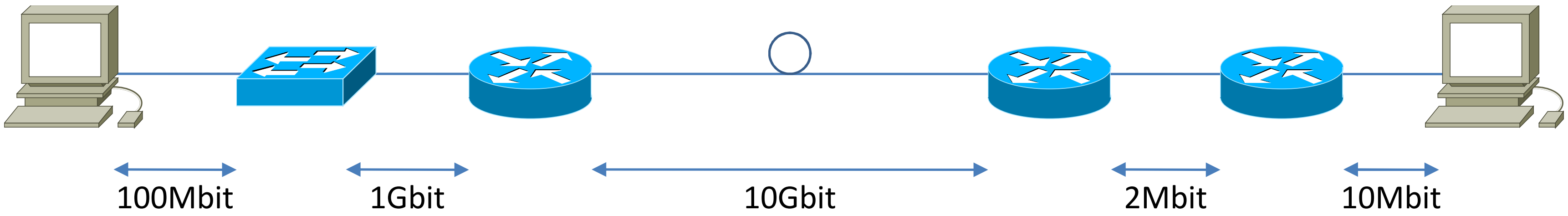
- Low Latency Queuing solves this problem
  - A packet in this queue is always sent first
    - Head-of-line blocking problem again
    - Configure an upper limit of link usage
  - At most one MTU sized packet serialization delay before LLQ packet is sent
  - Also called SP for Strict Priority

Jitter (4): Queue Schedulers



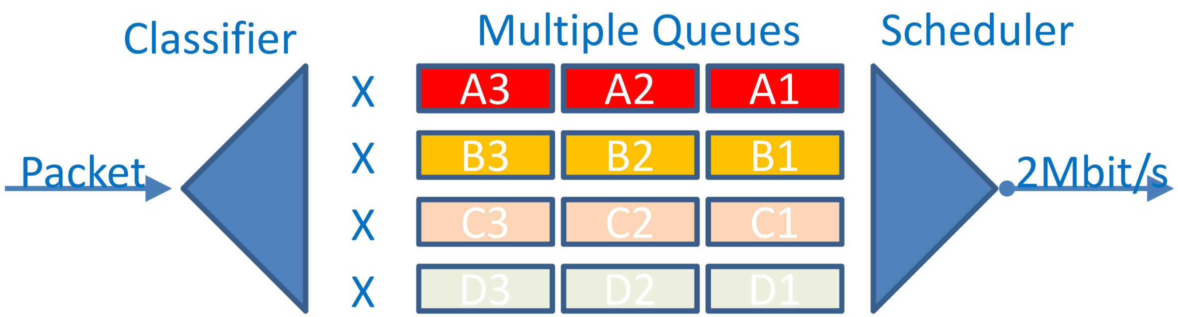
- LLQ is only for real-time traffic
- Never mix real-time (VoIP) and bursty traffic (anything TCP) in a Low Latency Queue!
- Limit the LLQ share to some sane amount (<50%)
- If a link is only used for real-time traffic no special queuing is necessary

Jitter (5): Summary



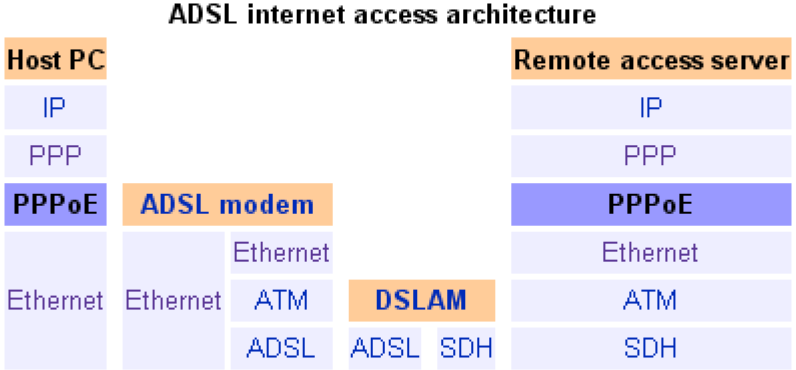
- Real-time traffic (VoIP) is very sensitive to jitter
  - Total constant delay from propagation and processing is not a problem
- Lets calculate worst case jitter spread
  - Calculate serialization delay for the whole path
    - For maximum MTU sized packets
    - Normally 1‘500 Bytes for Ethernet (overhead!)
  - *0ms to 7.193ms base jitter you always have*
  - VoIP doesn‘t care about average jitter
  - Maximal jitter is important for jitter buffers

Packet Loss



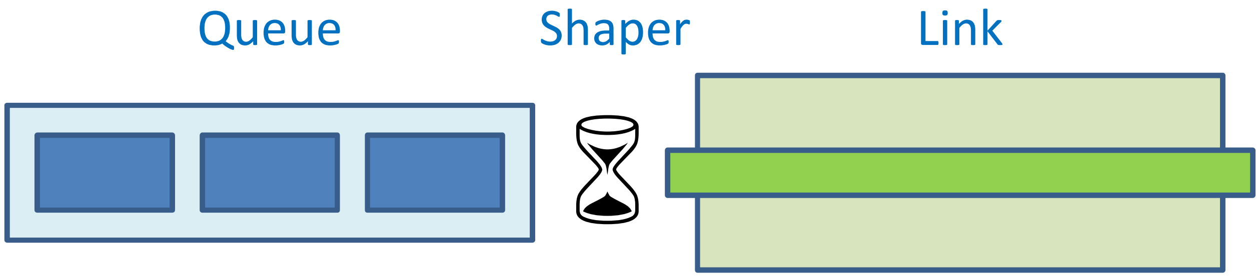
- Packet loss happens when the queue overflows during congestion
  - Buffers are a limited resource
- TCP uses packet loss as primary signal to slow down
  - Some algorithms use delay too
  - Active queue management to prevent simple tail drop behavior
    - RED (Random Early Detection) drops packets before the queue is full to signal TCP to slow down and prevent a tail mass-drop
    - RED has a couple of optimized variants

Bandwidth (1): Calculating



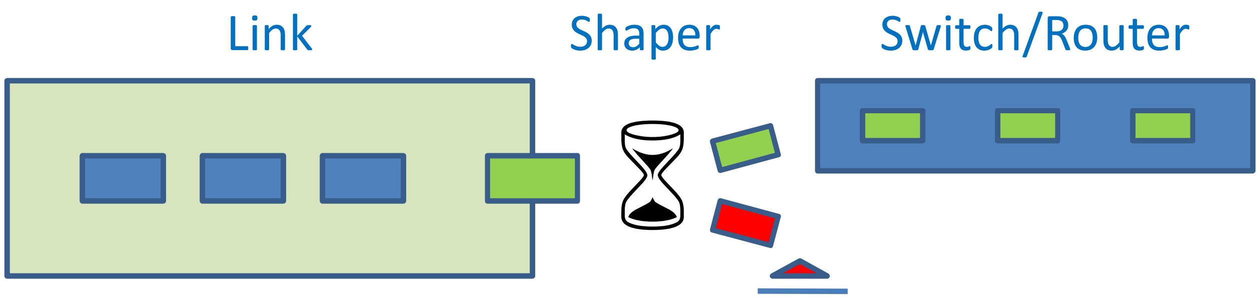
- Calculating the effective net bandwidth is not trivial
- Don‘t forget all the headers
  - Ethernet: MAC header + CRC + IFG
  - HDLC/PPP: Frame header + Escaping (7F)
  - MPLS
  - ATM cell overhead
  - And so on…
- Some overheads are non-linear
  - Packet size distribution is important too

Bandwidth (2): Shaping



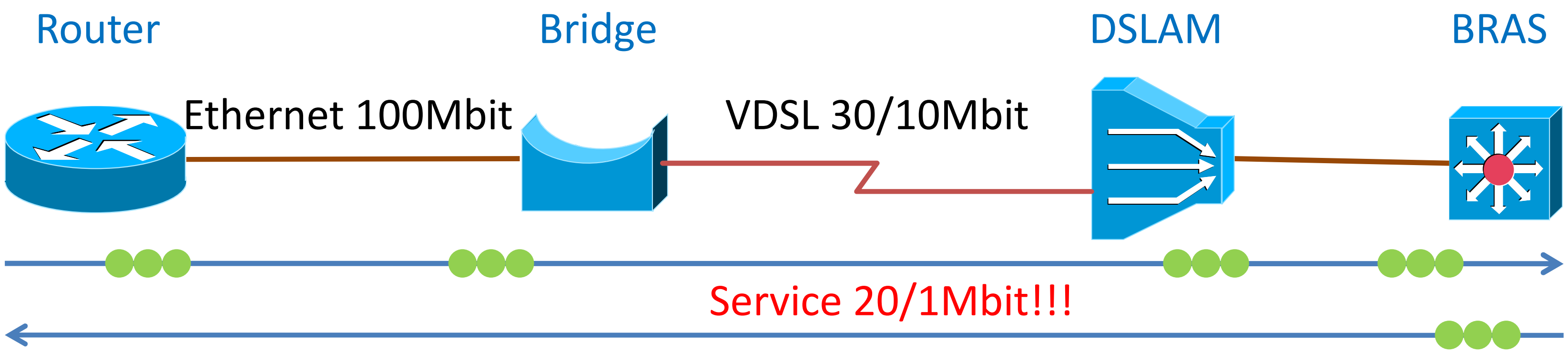
- Bandwidth is reduced to less than physical link speed
  - Serialization delay is still based on packet size divided by link speed
- The number of bytes is limited per time interval
  - Token bucket system
    - granularity
    - burstiness (leaky bucket)
- A queue is formed in front of the limiter
  - Packets wait for the next transmission interval

Bandwidth (3): Limiting



- Normally shaping works only on interface output
- On interface input many devices only support rate limiting
  - All packets that exceed the limit per interval are dropped!
  - No queueing supported
  - Instant packet loss
  - Must shape bandwidth on sending device!

Bandwidth (4): Example

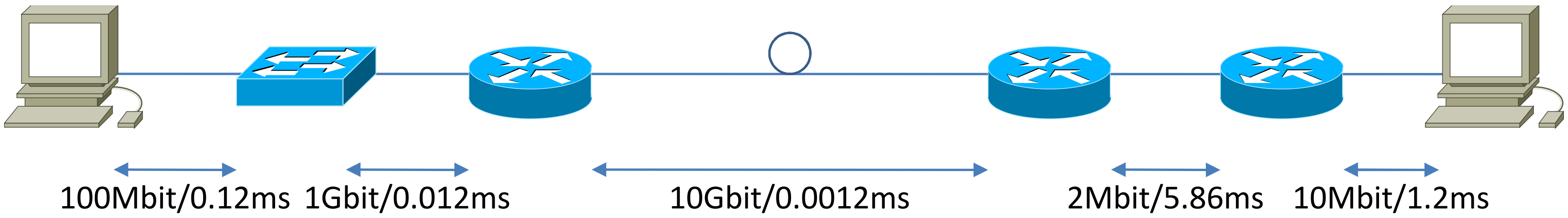


- Popular VDSL Service in Switzerland
  - Link Speed is „fixed“ at 30/10Mbit
  - Service Speed is shaped to 20/1Mbit on BRAS
- What to do?
  - Shape router interface down to 1Mbit
  - Configure classifier for your priority traffic
  - Configure LLQ and WFQ

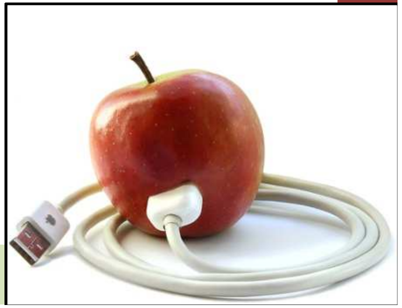
Summary (1)

- Making QoS work in a packet environment is a bit of work
- Constant delay
  - Propagation ~5ns/m or 0.5ms/100km
  - Processing ~20µs per L2/L3 device
  - Encoding up to 20ms with A+VDSL
- Variable delay = Jitter
  - Serialization link speed & packet size
  - Queueing micro-peaks & congestion
- Jitter can‘t be eliminated in a mixed-use network path
- Lower link speed means more jitter

Summary (2)



- Every device must participate in QoS management
- All devices must have the same classifier rules
- Make sure to prevent non-QoS-managed bandwidth reductions



Questions?

- Don‘t hesitate to contact me!
- Thank you for your attention
- I‘m available as a consultant and network engineer who can look at your situation in detail
  - Email: oppermann@networx.ch