HO CHI MINH CITY UNIVERSITY OF SCIENCE
FACULTY OF ELECTRONICS AND TELECOMMUNICATIONS
DEPARTMENT OF TELECOMMUNICATIONS AND NETWORKS

COURSE

COMPUTER NETWORKS

Chapter 03

Packet

Reference: Peter L Dordal, "An Introduction to Computer Networks," Jul 26, 2019

Lecturer: Nguyen Viet Ha, Ph.D. Email: nvha@fetel.hcmus.edu.vn

1 Packet Delay

1. Packet Delay

❖ **Bandwidth delay**

- ➢ The time needed for a sender to get the packet onto the wire.
- ➢ Ex: 1500-byte packet on 100 Mbps Ethernet,
  - ○ Bandwidth delay $= \frac{12\ 000\ bits}{\frac{100\ bits}{\mu s}} = 120\ \mu s.$

❖ **Propagation delay**

- ➢ The time it takes for the head of the signal to travel from the sender to the receiver.
- ➢ The distance divided by the x speed of light.

1. Packet Delay

Medium	Propagation speed
Thick Coax	0.77c (231,000 km/sec)
Thin Coax	0.65c (195,000 km/sec)
Twisted Pair	0.59c (177,000 km/sec)
Fiber	0.66c (198,000 km/sec)

❖ **Propagation delay**

- ➢ The time it takes for the head of the signal to travel from the sender to the receiver.
- ➢ The distance divided by the x speed of light.
- ➢ Ex: 1,000 m of Ethernet using twisted pair
  - ○ Propagation delay $= \frac{\quad}{\quad} = \quad \mu s.$

1. Packet Delay

❖ Store-and-forward delay

- The time spent reading in the entire packet before any of it can be retransmitted.

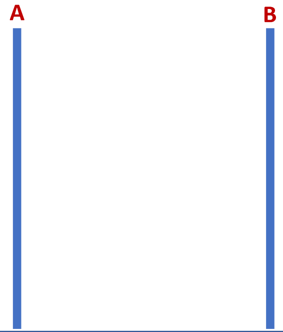
❖ Queuing delay

- Depend on competing traffic and/or the system specifications.
  - Other words, is a delay based on how long the packet has to stay around in the intermediate device (e.g., router).

1. Packet Delay

❖ Delay examples.

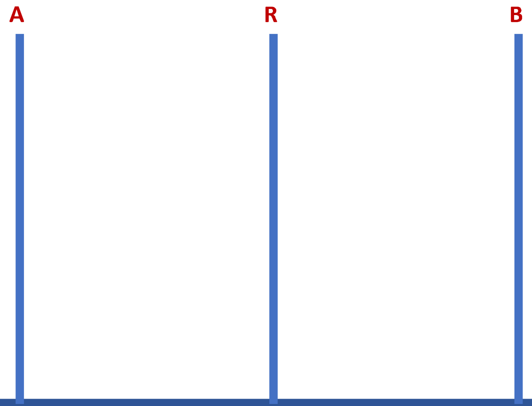
- Case 1: A------------------B
  - Propagation delay is 40 μsec
  - Bandwidth is 1 byte/μsec (1 MB/sec, 8 Mbit/sec)
  - Packet size is 200 bytes (200 μsec bandwidth delay)
  - → One-way transmit time = μsec



1. Packet Delay

❖ Delay examples.

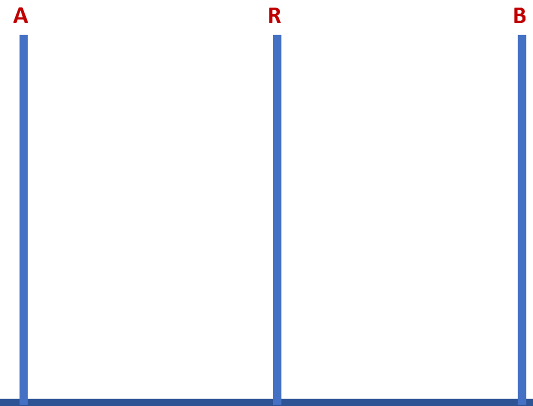
- Case 2: A --------- R ---------- B
  - Two links, each with propagation delay 40 μsec; bandwidth and packet size as in Case 1.
  - → One-way transmit time = μsec



1. Packet Delay

❖ Delay examples.

- Case 3: A --------- R ---------- B
  - The same as 3, but with data sent as two 100-byte packets.
  - → One-way transmit time = μsec



❖**Round-trip time (RTT)**

- At most non-LAN scales, the delay is typically simplified to the RTT.
  - The time between sending a packet and receiving a response.
- Different delay scenarios have implications for protocols:
  - If a network is bandwidth-limited then protocols are easier to design. Extra RTTs do not cost much, so we can build in a considerable amount of back-and-forth exchange.
  - However, if a network is delay-limited, the protocol designer must focus on minimizing extra RTTs.
    - As an extreme case, consider wireless transmission to the moon (0.3 sec RTT), or to Jupiter (1 hour RTT).

❖**Bandwidth x Delay**

- Represents how much we can send before we hear anything back.
- (Delay: usually involving RTT)
- Ex:

RTT	bandwidth	bandwidth × delay
1 ms	10 Mbps	1.2 kB
100 ms	1.5 Mbps	20 kB
100 ms	600 Mbps	8 MB
100 ms	1.5 Gbps	20 MB

❖**Packet Delay Variability**

- **RTT is variable.**
  - On Ethernet and Wi-Fi networks there is an initial "contention period" before transmission actually begins.
  - Different packets are routed via slightly different paths
  - Or are handled differently by different queues of a parallel-processing switch.
  - Mobile nodes are involved, then the distance and thus the propagation delay can change.

❖**Link's bandwidth, too, can vary dynamically.**

- Shared bandwidth
- QoS (Quality of Service)
  - E.g., routers reserve a varying amount of bandwidth for
    - High-priority traffic
    - Depending on demand
    - Best-effort traffic

1. Packet Delay

❖ **Link's bandwidth, too, can vary dynamically.**

➢ ***RTT_{noLoad}*** to be the time it takes to transmit a packet from A to B, and receive an acknowledgment back, with no queuing delay.

➢ We will usually assume that ***RTT_{noLoad}* is fixed and well-defined**.

2 Packet Size

2. Packet Size

❖ **How big should packets be?**

➢ Should they be large (*e.g.,* 64 kB)

➢ or small (*e.g.,* 48 bytes)?

➢ Large packets would not allow other senders timely access to transmit.

➢ Large packets waste a smaller percentage of bandwidth on headers.

➢ In store-and-forward switches, smaller packets have much better throughput.

2. Packet Size

❖ **Packet Size vs. Error Rates**

➢ For relatively high error rates, it turns out to be better to send smaller packet, because when an error does occur then the entire packet containing it is lost.

- Choose the packet size small enough that most packets do not encounter errors.

➢ To be fair, very large packets can be sent reliably on most cable links (e.g., TDM and SONET). Wireless, however, is more of a problem.

2. Packet Size

❖ **Packet Size vs. Real-Time Traffic**

- It is common to commingle bulk traffic on the same links with real-time traffic.
  - Real-time traffic has higher priority than bulk traffic.
    - Router does not begin forwarding a bulk-traffic packet if there are any real-time packets waiting.
    - However, once a bulk-traffic packet has begun transmission, it is impractical to interrupt it.

❖ Limited to the maximum **Ethernet packet size.**

- IPv4: 1500 bytes
- IPv6: 2 MB

3 Error Detection

3. Error Detection

❖ Packet error detection is to add some extra bits (or called error-detection code)

- Allow the receiver to determine if the packet has been corrupted in transit.

❖ A corrupted packet will then be discarded by the receiver.

❖ Reasons:

- Low-frequency bit errors due to things like cosmic rays.
- Interference errors, typically generated by nearby electrical equipment.

3. Error Detection

❖ **Single parity bit**: this will catch all one-bit errors.

- Ex:
  - Data: **1001**
    - → parity bit = 1^0^0^1 = **0**
  - Send: **10010**
  - Receive: 10**1**10
    - → Check: 1^0^1^1^0 = 1 ≠ **0**
    - → Data is corrupted.

3. Error Detection

❖**Internet checksum**: used by IP, TCP and UDP

- Taking the **one's-complement sum** of the 16-bit words of the message.
  - One's-complement sum: take the sum A+B. If there is an overflow bit, add it back in as low-order bit.
- A weakness of any error-detecting code based on sums is that transposing words leads to the same sum, and the error is not detected.
  - In particular, if a message is fragmented and the fragments are reassembled in the wrong order, the ones-complement sum will likely not detect it.

3. Error Detection

❖**Cyclical Redundancy Check (CRC)**

- Ethernet frame.

3. Error Detection

❖**Error-Correcting Codes (also called forward error correction)**

- That allows the receiver in many cases to figure out which bits are corrupted, and fix them.
- Mostly used in the Physical layer.

QA

Lecturer: Nguyen Viet Ha, Ph.D.
Ho Chi Minh City University of Science
Faculty of Electronics and Communications
Department of Telecommunication and Networks
Email: nvha@fetel.hcmus.edu.vn