CSC361

Computer Networking

Mantis Cheng

Dept of Computer Science

Unit 1

Internet Architecture

Important Concepts

- Internet Architecture
- Packet vs Circuit Switching
- Encapsulation Principles
- Performance of Packet Switch Network
- Bandwidth Delay Product
- TCP/IP Model and Properties
- Internet Applications: End-To-End Principle

History of Networks and The Internet (12:40)

(watch this at home)

The Internet and IP Introduction (4:16)

Summary of Key Concepts

- 4 Layer Model of TCP/IP
- Packet Switch Network
- Internet Applications is primarily end-to-end
- IP (v4) is fundamental to how Internet works
- Layering and Encapsulation

Packet Switching: Principles (16:09)

Summary

- For voice switching, a **circuit** (dedicated path) is setup before communication.
- Circuit switching must **reserve** each link's bandwidth even if the link is **idle**.
- Computer data communication is **bursty**.
- Computer applications require **different** rates.
- Packeting switching splits data into many **packets**.
- Each packet is transmitted **independently**.

The 4 Layer Internet Model (13:22)

Summary

- Link is the lowest layer, which carries data frame from **hop to hop**.
- Network is the IP layer, which provides a logical but unreliable datagram service. All Internet routers support only the **bottom two** layers. The IP protocol glues the whole of Internet together.
- TCP provides a **reliable bi-directional byte-stream** connection service.
- UDP is an **unreliable** datagram service built on top of IP.

Summary (continued)

- HTTP is an application level protocol that uses TCP.
- Routers only provide **network** and **physical** layer services.
- Application level protocols are thus **end-to-end**; they are **not** supported inside the Internet infrastructure, i.e., **switches** and **routers**.

Summary (continued)

- Applications don't care which path the communication takes place between two endpoints.
- Data frame is delievered hop-by-hop by the Link layer.
- Network layer delievers datagrams end-to-end, but **no** guarantee.
- Each router consists of a **network** and a **link** layer; there are **no** transport or application layers!

Summary (continued)

- IP makes **no** attempt to deliver all datagrams reliably or in order.
- It is transport layer, TCP, that delivers data segments **reliably** and **in order**.
- Some applications use UDP to deliver fast datagram without any connection.

[Encapsulation Principle (8:11)]

Summary

- Each layer provides an abstraction.
- The layer above uses the services provided in the lower layers.
- HTTP request/response (headers + data) are sent as TCP payloads; i.e., An TCP segment encapsulates a HTTP request/response.
- TCP segments are transmitted/received as IP payloads, i.e., an IP packet encapsulates TCP segment.

Wireshark Demo

(curl info.cern.ch)

(Open `info.cern.ch.pcap` in Wireshark.)

Summary

- Wireshark is a powerful **packet sniffer**.
- It can captures **all** packets sent/received over one or more network interfaces (e.g., eth0, wifi, ...)
- Packets are captured as **raw binary data**; Wireshark can decode almost all physical (Ethernet) or logical protocol (IP, UDP, TCP).
- The captured packets can be saved for future analysis.
- Wireshark is also a **dangerous** tool!

Packet Switching End-to-end vs Queuing Delays (20:40)

Summary

- **Propogation delay**: $t_d = d/s$, where d is distance between two endpoints, s is almost the speed of light (approx., $2.5 * 10^8$ meter/sec).
- **Tranmission delay**: $t_m = l/r$, where l is the packet length in bits, r is the transmission rate (bits/sec).
- **End-to-end delay**: $\sum_i t_{d_i} + t_{m_i}$, where t_{d_i} is a **fixed** delay, and t_{m_i} is determined by r_i of each link and packet length l.

Summary (continued)

- In a **store-and-forward** packet switching network, many packets may arrive at a router/switch at the same time, and may be delievered to the same outgoing link.
- As a result, packets may be queued at a router due to **congestion**, too many packets arrived at the same time. Thus, routers have buffers, which hold packets waiting to be transmitted.
- These packet buffers introduce an additional delay, called **queueing delay**.

Bandwidth Delay Product

- The *effective* end-to-end throughput is sometimes known as **bandwidth** (bits per second), i.e., how many bits per second can actually be transmitted and received correctly?
- The overall **end-to-end delay** (seconds) varies constantly due to hops, congestion, transmission rates, arrival rates, packet loss, etc.
- What is the quantity **bandwidth * delay**?
- What does it mean?

[A Day In the Life of an Application (12:11)]

Summary

- Most network applications rely on a reliable bi-directional data stream connection.
- HTTP is a ASCII-based protocol for transmitting HTML documents.
- HTTP relies on a **reliable bi-directional data stream**, which is supported by the transport layer.
- **NAT** created problems for network applications
- Network applications can be **client/server** or **peer-to-peer** based

A Day In the Life of a Packet (12:00)

Summary

- Application layer transmits data streams, **end to end**
- Transport layer transmits segments, **end to end**
- Network layer transmits datagrams, **router by router**
- Link layer transmits frames **hop by hop**
- A Connection is establied via a **3-way handshake**, which is identified by a pair of **IP addresses** and **port numbers**.

Summary (continued)

- A router uses a **forwarding table** to determine how to route IP datagrams
- 3-way handshake includes `SYN`, `SYN+ACK` and `ACK` packets.
- `traceroute` is a tool for identifying each router along a path/route to a particular host; the path changes from time to time, and the end-to-end delay various from hop-to-hop

[The End To End Principle (10:33)]

Summary

- Why doesn't the **Network** layer do more?
- All **smart** in network applications are implemented at the endpoints.
- **End-to-End principle** is about making improvements on network communication only at the fringes, i.e., endpoints.

The End