Lecture 2 (Packet Switching)

1 Recap

- Each packet is made up of a header, data and (optionally) trailer
- **Routing** is determined dynamically using header information
- At each node, the packet is stored temporarily before it is forwarded to the next node (**store-and-forward**)

1.1 Datagram Packet Switching

- Path for each node is determined individually, choosing the optimal path based on congestion and other factors
- This can lead to packets arriving in different order that they were sent in

2 Application Protocol

The **URL (Uniform Resource Locator)** is used to make it very easy to process the requests in a systematic manner. It involves using **HTTP(S) (Hyper Text Transfer Protocol)** and **TCP (Transmission Control Protocol)**. To facilitate the communication, the following sequence of steps occur:

1. Finding the IP: 6 messages
2. Connection establishment (via TCP): 3 messages
3. HTTP request and and acknowledgement: 2 each for request and acknowledgment, total 4 messages
4. Tearing down the TCP connection: 4 messages

Thus, a total of 17 messages are communicated between the two end points

3 Important Terminology

3.1 Link

Transmission technology such as:

- Coaxial
- Twisted pair: has multiple sub-cables which serve different purposes
- Optical
- Radio

3.1.1 Properties of Links

1. **Latency (delay)**: notion of *length* of the link
   Propagation time for data on the link
2. **Bandwidth (capacity)**: notion of *width* of the link
   Amount of data transmitted per unit time
3. **Bandwidth-Delay Product (BDP)**: notion of *volume* of the link
   The total amount of data that can be present in the link at any point of time

3.1.2 Utilisation of Link

Fraction of time link is busy transmitting, denoted by ρ. It is equal to:

$$A/B,$$

where, A = arrival rate, B = bandwidth.

3.2 Node

The computational device at the end of every link. It is if two types:

1. Host: a "user"
2. Network node: switch/router

3.3 Packets

They contain the payload (body), and also contain a header (the *interface*)

3.4 Propagation Delay

- Delay of reaching the next switch after transmission (for each bit, not packet)
- Corresponds to the latency
- Haven't changed much historically

3.5 Transmission Delay

- Time taken to "transmit" all packets into a link
- Equals the ratio of packet size to the bandwidth
- Has reduced significanty over the decades

3.6   Queuing Delay

- Time spent waiting in queue
- Computed by dividing the packet bits ahead by the bandwidth of the next link
- Depends on stochastic multiplexing but has gotten smaller
- Does not happen if
    - i. packets evenly spaced
    - ii. arrival rate is less than service rate
- Caused by
    - i. *Bursti-ness* of arrival
    - ii. Variation in packet length

3.7   Round Time Trip

- Time taken by a packet to reach the destination and response to reach the source
- Equal to sum of all propagation, transmission, queuing and processing delays over the path

Note: **Content Distribution Networks (CDNs)** *cache* the data closer to the clients ("trending" data can be stored locally). This reduces backbone bandwidth requirements and latency.
The first CDN company was "Akamai", which was started at an MIT competition

3.8   Jitter

- Difference between minimum and maximum delay
- Mostly affected to differences in queuing delay
- Think about why is it relevant?

3.9   Packet Loss

1. **Buffer Overflow**: different strategies can be used such as drop tail, drop head or drop a random packet
2. **Corruption**: can be detected using checksum, low probability to occur because of sophisticated technology

4   Queuing Terminology

- A: arrival rate
- P: peak rate
- W: waiting time
- L: average number of packets waiting in queue (length of queue)

4.1   Little's Law

$$L = A \times W$$

This is an important law in queuing theory and helps in computing W since L and A are easy to obtain practically

5   Statistical Multiplexing

- Enough to have a bandwidth (with a margin) equal to the average peak capacity for bursty data
- Relies on the fact that not all bursts happen at the same time
- Similar implementations are present throughout history and different domains:
    1. Phone network instead of dedicated lines between each pair of end nodes
    2. Cloud computing has shared data centres instead of dedicated ones

6   Interface

Each level of the protocol interacts independently with its own level. The information is encapsulated towards the lowest level at the peer which is then transmitted to the other peer which then "decapsulates" the information. This architecture is known as **Open Systems Interconnection (OSI) Architecture**