Fundamentals of Computer Networking

Guevara Noubir
noubir@ccs.neu.edu
CSG150

What is CSG150 about?

- Understand the basic principles of networking:
  - Description of existing networks, and networking mechanisms
  - Understanding of networks modeling and analysis tools
- Covers:
  - Terminology, layering concept, physical layer, data link layer, basics of queuing theory, medium access control (MAC), algorithmic aspects of routing, flow control (window/rate control)
- Very limited coverage of higher layers (applications, compression, TCP/IP programming, etc.)

Course Outline

- Introduction to networking and to 7-layer OSI architecture
- Error detection, retransmissions strategies, and framing
- Introduction to Markov chains and queuing theory
- Multiple access schemes (MAC sublayer):
  - aloha, CSMA, ethernet, token rings, wireless, etc.
- Routing algorithms: Unicast and Multicast
  - Dijkstra, Bellman-Ford algorithms and RIP and OSPF protocols
- Packet Scheduling:
  - Max-Min Fairness, Weighted Fair Queuing.
- Multicast Routing
- Flow Control

Lecture 1 Outline

- Networking Concepts and Terminology
- Protocols Layering
- Internet Architecture
- Reading: Chapter 1.

Basic Terminology

- **Network**: set of nodes interconnected through communication links
- **Node**: host, router, switch
- **Link**: twisted-pair, coaxial cable, optical fiber, wireless
- **Protocol**: set of rules and conventions used between peer entities to communicate
- **Message:** sequence of bits/application level (e.g., email, document)
- **Packet:** messages are broken into packets that can transmitted between network nodes
- **Session:** transaction consisting of a sequence of message exchanges

Types of Networks

- Several taxonomies exist…
- Transmission technology based differentiation:
  - Point-to-point networks
  - Broadcast networks (generally small area: ethernet)
- Scale based differentiation:
  - Local Area Network (LAN): privately owned networks, up to few miles in size (e.g., ethernet)
  - Metropolitan Area Networks (MAN): larger than LANs, may cover a city (e.g., IEEE802.6 DQDB)
  - Wide Area Network (WAN): covers a large geographical area (e.g., country, continent)

Performance Issues

- Packets arrival modeling assumed (simplicity) either:
  - *Poisson Process* (data packets)
  - *ON/OFF flow* model (for digitized voice)
- Common performance metrics:
  - latency and throughput,
  - delay-jitter, and rate-jitter (maximum - minimum)

Latency = *PropagationDelay* + *TransmissionDelay* + *QueuingDelay*
PropagationDelay = *Distance/SpeedOfLight (independent of message size)*
TransmissionDelay = *MessageSize/Bandwidth*
QueuingDelay = delay due to time spent waiting in queues

Sessions Characteristics

- Message arrival rate:
  - Poisson arrivals, deterministic arrivals, uniformly distributed
- Session holding time: duration of the session
- Expected message length and length distribution
- Allowable delay: depends on the application type
  - Interactive, real-time, email, voice, video-on-demand
- Reliability: depends on the application (FER)
  - Voice/video (tolerant), email (strict), sensing etc.
- Message & packet ordering: depends on the application
  - Databases, email, etc.

Types of Applications

- Interactive terminal and computer sessions:
  - Small packet length, small delay, high reliability
- File transfer:
  - High packet length, high delay, high reliability
- High resolution graphics:
  - High packet length, small delay, low arrival rate
- Voice application:
  - Small packet length, small delay, small reliability, high arrival rate

Sessions Transmission Paradigms

- Circuit Switching
- Store-and-Forward Switching

Circuit Switching

- On session establishment a path from source to destination is selected. Resources are allocated over all the links of the path. Route does not change during session life.
- Links can be shared by different sessions through mechanisms such as *time-division multiplexing* (TDM) or *frequency-division multiplexing* (FDM).
- For any link: the sum of rates of sessions using the link is at most the bandwidth of the link.
- Example: telephone networks.
- Guarantees: rate and in order packets delivery.

Store-and-Forward Switching

- Links are shared on a "*demand basis*" vs. fixed allocation
- Packets wait in a queue before being transmitted

Inefficiency of Circuit Switching

- Broadly speaking: when a session is idle, the reserved resources are lost.
- Simple analysis of session *s*:
  - λ: message arrival rate;
  - $1/\lambda$: inter-arrival time between messages
  - $\overline{X}$: expected transmission time
  - $\overline{L}$: expected message length
  - Bit rate allocated to *s*: $r_s = \frac{\overline{L}}{\overline{X}}$
  - Link utilization = $\lambda\overline{X}$
  - T <0.01 for interactive app.

P: propagation delay
Q: queuing delay
T: allowable expected delay

$$\overline{X} + P + Q \le T \Rightarrow \lambda\overline{X} < \lambda T$$

$$if \lambda T << 1 \Rightarrow utilization - is - poor$$

Circuit Switching vs. Store-and-Forward Switching

- Advantage of Store-and-Forward over Circuit Switching:
  - Network utilization is better (each link is utilized when there is some traffic)
  - Lower delays
- Drawbacks of Store-and-Forward:
  - Necessity of having control flow mechanisms to avoid buffer congestion and maintain acceptable delays. This is generally achieved through some feed-back to the senders.

More Taxonomy

- Message store-and-forward
- Packet store-and-forward
- Packet switching = store-and-forward switching

- Virtual circuit switching = packet switching + fixed path
- Dynamic routing => each packet finds its own path

Layering

- Layering: is a form of "*hierarchical modularity*"
- The role of each layer is to provide services for higher layers (abstracting lower layers)
- What is important is the functional relation between Inputs and Outputs
- Each layer has and interface to higher/lower layers (constituted of service access points: SAP)
- Entities at the same layer on different nodes are called *peer-entities*. They communicate through protocols
- A set of consecutive layers is called a *Protocol Stack*
- Drawback of layering: may hide important information (e.g., TCP over wireless links)

OSI Reference Model

- Open systems Interconnect (OSI) defined a reference for a layered architecture of data networks
- Existing protocol stacks (e.g., TCP/IP)) are quite different from the OSI RM but it is still an interesting conceptual model because of its clean structure

Application
Presentation
Session
Transport
Network
Data Link Control
Physical Interface

Physical Layer

- Function: provides a "*virtual bit pipe*"
- How: maps bits into electrical/electromagnetic signals appropriate for the channel
- The physical layer module is called a *modem* (modulator/demodulator)
- Important issues:
  - Timing: synchronous, intermittent synchronous, asynchronous (characters)
  - Interfacing the physical layer and DLC (e.g., RS-232, X.21)

Data Link Control Layer (DLC)

- Receives packets from the network layer and transforms them into bits transmitted by the physical layer. ***Generally*** guarantees order and correctness.
- Mechanisms of the DLC:
  - Framing: header, trailer to separate packets, detect errors…
  - Multiple access schemes: when the link is shared by several nodes there is a need for addressing and controlling the access (this entity is called MAC sublayer)
  - Error detection and retransmission (LLC sublayer)

Network Layer

- Provides naming/addressing, routing, flow control, and scheduling/queuing in a multi-hop network
- Makes decisions based on packet header (e.g., destination address) and module stored information (e.g., routing tables)
- General comment: each layer looks only at its corresponding header (here packet header)
- Routing is different on virtual circuit networks than on datagram networks

Datagram vs. Virtual Circuits

- In virtual circuits after an initialization phase all packets follow the same path. We generally assume that packets are delivered once and only once, and in order.
- In a datagram network packet are routed individually. They may be lost or delivered out-of sequence.
- Sometimes referred to as: *connection-oriented service* and *connectionless service*.

Flow control & Congestion

- Flow control avoids sending data faster than the destination can absorb
- Congestion control avoids sending data faster than the the network can handle.
- Requires:
  - efficient feedback mechanism, buffer management, route load-balancing
- In a connectionless service it is not easy to negotiate an equitable service between users

Transport Layer

- Provides a reliable mean to transmit messages between two end-nodes through:
  - Messages fragmentation into packets
  - Packets reassembly in original order
  - Sessions multiplexing and splitting
  - Retransmission of lost packets
  - end-to-end flow control and congestion control

Session Layer

- Was intended to handle the interaction between two end points in setting up a session:
  - Two connections
  - Service location (e.g., would achieve load sharing)
  - Control of access rights
- In many networks these functionalities are inexistent or spread over other layers

Presentation Layer

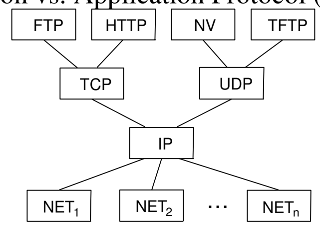
- Was intended to provides data encryption, data compression, and code/types conversion from one architecture to another.

Application Layer

- What's left …
- Examples:WWW, Email, Telnet, …

Internet Architecture

- Defined by Internet Engineering Task Force (IETF)
- IETF requires working implementations for standard adoption
- Application vs. Application Protocol (FTP, HTTP)



Implementing Network Software

- Success of the internet is partially due to:
  - Minimal functionality within the network
  - Most of the functionality running is software over general-purpose computers
- Simple Application Programming Interface
- Efficient Protocol implementation

Application Programming Interface

- Each OS can have a special interface exported by the network to the applications developer
- Most widely used network API is: socket interface
  - Initially developed by the Berkeley Distribution of Unix and today ported to almost all OS
  - ```
    int socket(  int domain, /* PF_INET, PF_UNIX */
                 int type, /* SOCK_STREAM,SOCK_DGRAM */
                 int protocol)
    ```


Client/Server Sockets

- TCP:
  - Client: socket, connect, (send, recv)*, close
  - Server: socket, bind, listen, (accept, (recv, send)*, close)*
- UDP:
  - Client: socket, bind, sendto/recvfrom
  - Server: socket, bind, sendto/recvfrom
- `int connect (int socket, struct sockaddr *address, int addr_len)`
- `int send (int socket, char *message, int msg_len, int flags)`
- `int recv (int socket, char *buffer, int buf_len, int flags)`
- `int bind (int socket, struct sockaddr *address, int addr_len)`
- `Int listen (int socket, int backlog)`
- `Int accept (int socket, struct sockaddr *address, int *addr_len)`
- Other: hostent *gethostbyname(const char *);

Summary

- Networking terminology and basic concepts
- Layering in networking
- 7-layers OSI reference model
- Internet architecture
    ```