Networking Course Notes (Extended)

Md Sadek Hossain Asif

March 17, 2025

Contents

Chapter 1

Computer Networks and the Internet

1.1 Radio Channels & Wireless Communication

1.1.1 Radio Channels: The Basics

Radio channels are part of the electromagnetic spectrum and require no physical wire. They can penetrate walls and enable user mobility. However, environmental factors significantly influence performance. For instance, *path loss* reduces the signal strength with distance, *shadow fading* can attenuate signals behind obstacles, *multipath fading* arises from reflections, and various electromagnetic signals can cause *interference*.

1.1.2 Classification of Terrestrial Radio Channels

- **Short-distance** channels typically reach a few meters, suitable for personal or peripheral devices such as wireless keyboards and headsets.
- **Local-area** channels span tens to a few hundred meters, as found in WiFi networks.
- **Wide-area** channels, exemplified by cellular systems, cover tens of kilometers and support mobile communication across broader regions.

1.1.3 Satellite Communication Channels

- **Geostationary satellites** orbit at about 36,000 km above Earth, remaining fixed above a point on the equator. They offer wide coverage but introduce a propagation delay of up to 280 ms due to the large distance.
- By contrast, **Low-Earth Orbit (LEO) satellites** orbit much closer and exhibit significantly lower delay.

However, because LEO satellites move rapidly relative to Earth, many satellites must be deployed in constellations to ensure continuous global coverage.

1.2 The Network Core

What Is the Network Core?

The core of the Internet is a high-speed, interconnected mesh of routers and links, often called the "backbone." It connects diverse access networks and content providers worldwide, transferring massive volumes of data over fiber, copper, or wireless backbones.

Packet Switching Fundamentals

Data in the network core is forwarded via **packet switching**. A large message is divided into *packets*, each forwarded independently by routers. Under **store-and-forward** operation, a router must first receive an entire packet before transmitting it on the outbound link.

Transmission Delay Example. Suppose each packet has length $L = 10{,}000$ bits, and the link rate is $R = 10^6$ bits/s (1 Mbps). The *transmission delay* to place the packet onto the link is:

$$d_{\text{trans}} = \frac{L}{R} = \frac{10{,}000 \text{ bits}}{10^6 \text{ bits/s}} = 0.01 \text{ s}.$$

Ignoring propagation and queuing, if exactly one router lies between sender and receiver, then a single packet might take about $2 \times 0.01 = 0.02$ s to appear in its entirety at the destination.

Packet switching adapts well to bursty traffic and re-routes around failures. However, if multiple packets arrive concurrently, *queuing* can occur, potentially causing *packet loss* if router buffers are saturated.

1.2.1 Key Delay Components in Packet-Switched Networks

Packets traversing a path of routers experience four main delay components:

Processing Delay (d_{proc})**:** This is the time spent examining packet headers and making forwarding decisions. In modern routers, this delay is typically on the order of microseconds.

Queuing Delay (d_{queue})**:** This is the waiting time in the router's buffer if other packets are still transmitting. Queuing delay is highly variable—it can be nearly zero under light traffic conditions, but may increase dramatically (even up to seconds) under heavy congestion.

Transmission Delay (d_{trans})**:** This is the time required to push all bits of a packet onto the link, computed as $\frac{L}{R}$, where L is the packet size in bits and R is the link bandwidth in bits per second.

Propagation Delay (d_{prop})**:** This is the time for the bits to travel from one node to the next, given by $\frac{d}{s}$, where d is the physical distance and s is the propagation speed (typically close to 2×10^8 m/s in fiber).

Full End-to-End Delay. In the simple example provided earlier, we ignored the queuing delay because under ideal, lightly loaded conditions, it may be negligible. However, in practical scenarios, the queuing delay is often a significant contributor to the total delay. Therefore, a more complete expression for the end-to-end delay, when a packet traverses N routers, is given by:

$$d_{\text{end-to-end}} = \sum_{i=1}^{N} \left(d_{\text{proc}}^{(i)} + d_{\text{queue}}^{(i)} + d_{\text{trans}}^{(i)} + d_{\text{prop}}^{(i)} \right).$$

For a simplified, homogeneous network (i.e., each router has similar delays), this can be approximated as:

$$d_{\text{end-to-end}} \approx N \times \left(d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}} \right).$$

This full equation clearly includes d_{queue}, which is crucial for accurately modeling delay under various traffic conditions.

1.2.2 Circuit Switching and Multiplexing

In contrast to packet switching, **circuit switching** establishes a dedicated end-to-end path before data transfer begins. Once established, this circuit reserves all necessary resources (such as buffer space and a fraction of each link's transmission capacity) for the duration of the connection.

Consider traditional telephone networks. When you make a call, a dedicated connection (or circuit) is established between the caller and receiver. Resources along this path remain reserved even during silent periods, leading to potential inefficiencies.

A circuit on a link is implemented via:

1. **Frequency-Division Multiplexing (FDM):** The available frequency spectrum is divided into bands, and each circuit is allocated a specific frequency band. For example, telephone networks often allocate a 4 kHz band per call. Similarly, FM radio stations share the band between 88 MHz and 108 MHz.

2. **Time-Division Multiplexing (TDM):** Time is divided into frames of fixed duration, and each frame is subdivided into time slots. When a connection is established, one slot per frame is reserved exclusively for that connection. For example, if a link

transmits 8,000 frames per second and each slot carries 8 bits, then each circuit obtains a transmission rate of $8,000 \times 8 = 64$ kbps.

Suppose a file of 640,000 bits is to be sent from Host A to Host B over a circuit-switched network. Assume that:

- Each link uses TDM with 24 time slots.
- Each link has a bit rate of 1.536 Mbps.
- It takes 500 ms to establish the circuit.

Since the link is divided into 24 slots, each circuit receives a rate of:

$$\frac{1.536 \text{ Mbps}}{24} = 64 \text{ kbps}.$$

Thus, the transmission time for 640,000 bits is:

$$\frac{640{,}000 \text{ bits}}{64{,}000 \text{ bits/s}} = 10 \text{ seconds}.$$

Including the 0.5 second circuit establishment time, the total delay becomes 10.5 seconds. Note that this transmission delay is independent of the number of links in the circuit.

1.2.3 Packet Switching vs. Circuit Switching

Circuit switching predates packet switching and was used in early telephone networks. Before sending data, resources (links, bandwidth) are reserved end-to-end, guaranteeing a fixed transmission rate but risking underutilization during silent periods.

In **packet switching**, resources are shared on demand. This approach is more efficient for data and can dynamically accommodate more users. However, without reservations, packet-switched flows may face variable delays or packet drops, especially under heavy traffic.

1.3 Protocol Layers, Service Models, and Encapsulation

1.3.1 Introduction to Layered Architecture

Networking protocols are structured into layers to manage complexity. Each layer offers specific services to the layer above and expects certain functions from the layer below. This design makes individual layers easier to modify or upgrade without requiring changes to the entire system. Analogously, an airline splits passenger travel into separate functions (ticketing, baggage, etc.) that build upon each other to deliver a complete end-to-end service.

1.3.2 Protocol Layering in Computer Networks

A popular model includes five layers:

- **Application Layer**: Hosts user-facing applications such as HTTP for web browsing and SMTP for email.
- **Transport Layer**: Provides logical end-to-end data transfer, using protocols like TCP and UDP.
- **Network Layer**: Responsible for moving packets across networks, typically via IP addressing and routing.
- **Link Layer**: Handles local, hop-by-hop frame transmission over a specific link (e.g., Ethernet, WiFi).
- **Physical Layer**: Manages the transmission of raw bits via electrical or optical signals over the medium.

Encapsulation

Each layer adds its own header (and possibly trailer) around the data from the layer above, in a process called *encapsulation*. For example:

1. The **application** creates a message.
2. The **transport layer** packages this into a segment with ports and reliability info.
3. The **network layer** adds IP addressing to form a datagram.
4. The **link layer** frames it with local delivery headers/trailers.

This nesting of headers is undone at the destination, layer by layer.

1.4 Delay, Loss, and Throughput in Packet-Switched Networks

Delay (the time a packet spends en route), **packet loss** (when packets overflow buffer capacity), and **throughput** (the rate at which data reaches its destination) are fundamental to understanding network performance.

1.4.1 Components of Delay

In packet-switched networks, every packet experiences delays at various stages as it travels from the source to the destination. These delays can be divided into four key components:

Processing Delay (d_{proc})**:** This delay represents the time a router or switch takes to examine the packet's header, check for errors, and decide on the appropriate forwarding action. Modern routers are optimized to perform these tasks very quickly, typically on the order of microseconds. However, the processing delay can vary depending on the complexity of the router's functions and the efficiency of its hardware.

Queuing Delay (d_{queue})**:** Once a packet is processed, it may need to wait in a buffer (queue) if the outgoing link is currently busy transmitting other packets. The duration of this delay depends on the network traffic load. Specifically, it is influenced by:

- The packet arrival rate, denoted by λ (in packets per second).
- The average packet size, L (in bits).
- The transmission rate of the link, R (in bits per second).

A useful metric for understanding congestion is the traffic intensity, calculated as $\lambda \cdot L/R$. When this ratio exceeds 1 for a sustained period, it indicates that packets are arriving faster than they can be transmitted, causing the queue to build up and the delay to increase significantly.

Transmission Delay (d_{trans})**:** This is the time required to push all bits of a packet onto the transmission link. It is calculated using the formula:

$$d_{\text{trans}} = \frac{L}{R},$$

where L is the packet's length in bits, and R is the link's transmission rate in bits per second. Transmission delay directly depends on the packet size and inversely on the link speed.

Propagation Delay (d_{prop})**:** After a packet is transmitted onto the link, the individual bits travel through the physical medium to reach the next node. The propagation delay is given by:

$$d_{\text{prop}} = \frac{d}{s},$$

where d is the physical distance between the nodes, and s is the propagation speed in the medium (typically around 2×10^8 m/s for fiber). This delay is independent of the packet's size and depends solely on the distance and the properties of the transmission medium.

1.4.2 Queuing Delay and Packet Loss

A router's buffer is finite. Once it is full, new packets arriving at that queue are dropped, causing **packet loss**. Higher-layer protocols like TCP may detect such losses and retransmit, which can temporarily increase congestion if a network is already saturated.

1.4.3 Throughput in Packet-Switched Networks

Definition: Instantaneous throughput is measured at any given point in time, while **average throughput** is $\frac{F}{T}$ for a file of size F bits transferred over T seconds.

Factors: A path's throughput is typically capped by the slowest (bottleneck) link. Multiple flows sharing a link must compete for available capacity, leading to lower per-flow throughput.

Goodput vs. Throughput

It is important to distinguish **goodput** from raw throughput. **Goodput** refers to the rate of *useful* data delivered to the application at the destination (i.e., excluding headers, retransmitted packets, and other protocol overhead). For instance, if a 1-Mbps link is running, but 20% of the bits are IP/transport headers, and an additional 10% are retransmitted packets, the effective rate of original application data delivered can be significantly less than 1 Mbps. We can model this simply:

$$\text{Throughput} = \frac{\text{Total bits delivered}}{\text{Time}},$$

but

$$\text{Goodput} = \frac{\text{Original (non-duplicated) app bits delivered}}{\text{Time}}.$$

Hence, goodput is often what matters to the end user, since it measures actual data received (e.g., useful file contents) rather than raw bits on the wire.

Example of Goodput Calculation. Suppose a file of $F = 8 \times 10^6$ bits is transferred in $T = 10$ seconds over a link with a nominal rate of 1 Mbps. Naively, you might say the throughput is $(8 \times 10^6)/10 = 0.8$ Mbps. However, if 1 Mbps includes a 10% overhead for headers or retransmissions, then only 0.9 Mbps is used for original data; thus the **goodput** is effectively 0.9×1 Mbps $= 0.9$ Mbps minus additional overhead. In real scenarios, repeated losses or large protocol headers can reduce goodput further.

1.5 Networks Under Attack and Security Challenges

Modern communication networks are vulnerable to numerous types of malicious attacks, partly because many Internet protocols were designed under early assumptions of mutual trust.

1.5.1 Types of Attacks

Malware Infiltration

Malware, including viruses, worms, and spyware, can spread over networks through drive-by downloads, email attachments, or direct exploitation. Infected hosts may form botnets, collectively controlled by attackers to distribute spam or launch large-scale attacks.

Denial-of-Service (DoS) and DDoS Attacks

DoS attacks aim to make a system or network resource unavailable to legitimate users. Techniques range from sending massive amounts of traffic (bandwidth flooding) to exploiting application vulnerabilities (vulnerability attacks). **DDoS** leverages multiple compromised hosts, making it far more difficult to block the attack by filtering a single source.

1.5.2 Other Security Threats

Packet Sniffing

Attackers can passively capture data traveling through a broadcast medium or wireless channel, collecting private information such as passwords. Encryption (e.g., TLS/SSL) is essential: even if packets are intercepted, the payload remains unintelligible without proper decryption keys.

IP Spoofing and Masquerading

By forging the source IP address, an attacker can masquerade as a trusted host, manipulating routing or misleading victims. Strong endpoint authentication is required to ensure packets truly come from whom they claim.

1.5.3 Why the Internet Is Vulnerable

The Internet's early research-focused design did not anticipate the scale and diversity of today's usage. Vast numbers of IoT devices, aging protocols, and sophisticated cybercriminals create ongoing vulnerabilities. Securing legacy architectures often involves performance trade-offs and incremental adoption of new protocols or patches.

1.5.4 Defense Strategies and Future Directions

Several measures mitigate these threats. **Encryption** (TLS, IPsec) prevents eavesdroppers from reading sensitive data. **Firewalls** and **Intrusion Detection Systems** filter malicious

or abnormal traffic patterns. **Rate limiting** and **traffic shaping** can dampen flooding attacks. **Endpoint authentication** verifies packet origins to curb IP spoofing. Future directions include cryptographically strengthening the core of the Internet via protocols like DNSSEC and secure BGP, as well as adopting zero-trust networking principles.