Unit-1

It's a pleasure to connect, Dr. Obula Raju D! □□

Your credentials are impressive:

- **M.Tech in Computer Networks & Information Security**
- **Assistant Professor, CSE at Andhra Engineering College**
- **Subject Code: 23A05501T**
- **Version :V08012025**

UNIT I:

- **Computer Networks and the Internet** What Is the Internet? Network Edge, The Network Core, Delay, Loss, and Throughput in Packet Switched Networks (Textbook 2), Reference Models, Multimedia Networks, Guided Transmission Media, Wireless Transmission (Textbook 1)
  Lecture: 8 Hrs

What Is the Internet?

1. **Definition:**
   The Internet is a global network interconnecting billions of computing devices (end systems/hosts) via communication links and packet switches.
2. **Components (Nuts-and-Bolts View):**

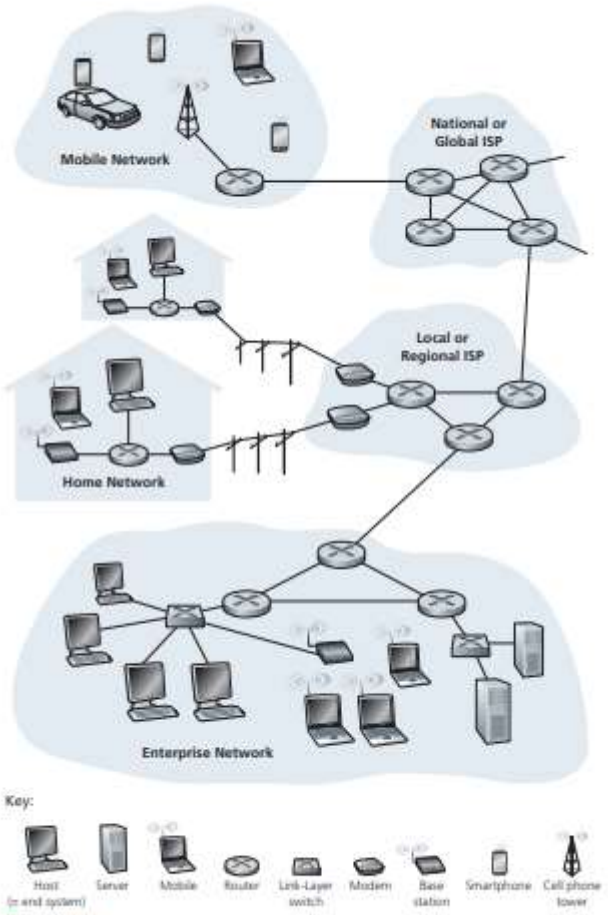

Figure 1.1 • Some pieces of the Internet

- **End Systems/Hosts:** Traditional (PCs, servers) and non-traditional devices (smartphones, sensors, appliances).
- **Communication Links:** Use various media (fiber, copper, radio) with different transmission rates (bits/sec).
- **Packet Switches:** Routers (core network) and link-layer switches (access networks) forward packets.
- **Packets:** Data segmented by the sender, headers added, transmitted, and reassembled at the destination.
- **ISPs:** Provide access (DSL, cable, WiFi, cellular) and interconnect hierarchically (lower-tier to upper-tier).

3. **Function (Services View):**

- Provides an infrastructure for **distributed applications** (email, web, streaming, games) running on end systems.
- Offers services via an **Application Programming Interface (API)** that programs use to send/receive data across the network.

4. **Role of Protocols:**

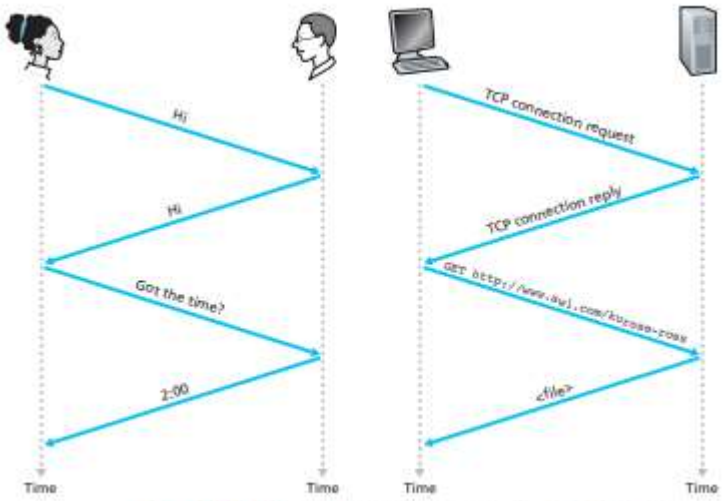

Figure 1.2 ♦ A human protocol and a computer network protocol

- **Definition:** Rules governing communication between network entities, defining:
  - Message format and order.
  - Actions taken on message transmission/receipt or other events.
- **Examples:** TCP, IP, HTTP, SMTP.
- **Standardization:** Developed by IETF as RFCs; ensure interoperability.
- **Analogy:** Like human protocols (e.g., greetings before asking a question).

Key Takeaways for a Summary Answer:

- **Global Network of Devices:** Connects diverse end systems.
- **Packet-Switched:** Data travels in packets via links and switches (routers, switches).
- **ISP Hierarchy:** Access providers interconnected to form the global infrastructure.
- **Service Platform:** Enables distributed applications through APIs.
- **Protocol Governed:** Relies on standardized rules (like TCP/IP) for communication.

Network Edge Components

1. Network Edge Components:

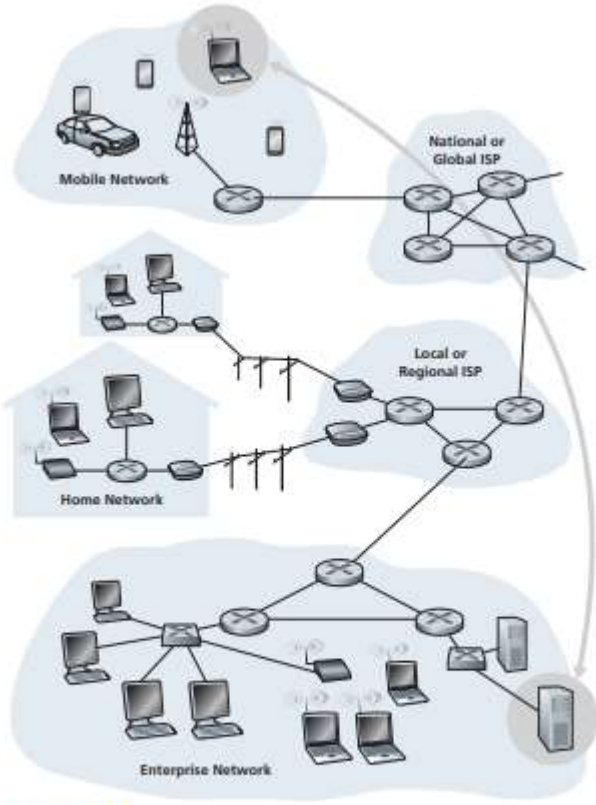

Figure 1.3 • End-system interaction

search results, e-mail, Web pages, and videos reside in large **data centers**. For example, Google has 30–50 data centers, with many having more than one hundred thousand servers.

- **End Systems (Hosts):** Devices at the edge (e.g., PCs, smartphones, servers, IoT devices).

- **Access Networks:** Connect end systems to the **first router** (edge router).

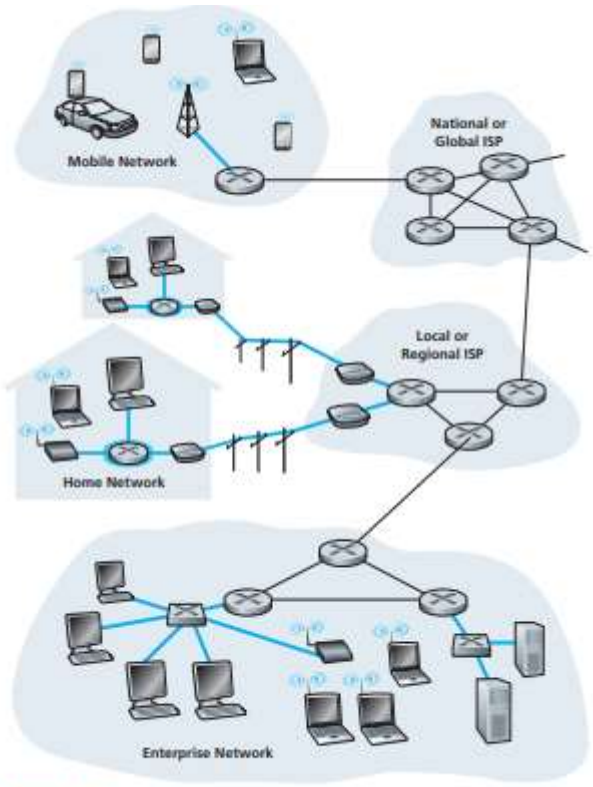

Figure 1.4 ♦ Access networks

-

2. Access Network Types:

a. Home Access:

- **DSL:** Uses phone lines.
  - Asymmetric (e.g., 24 Mbps down / 2.5 Mbps up).
  - Distance-sensitive (< 10 miles from central office).

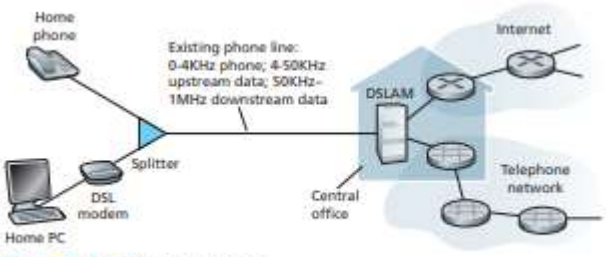

Figure 1.5 ♦ DSL Internet access

- **Cable:** Uses coaxial cable + fiber (HFC).
  - Shared medium (bandwidth drops with more users).
  - Asymmetric (e.g., 42.8 Mbps down / 30.7 Mbps up).

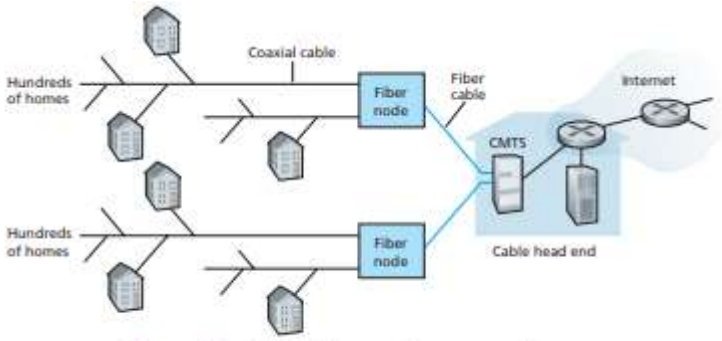

Figure 1.6 • A hybrid fiber-coaxial access network

- **FTTH (Fiber to the Home):**
  - Direct fiber or PON (Passive Optical Network).
  - High-speed (up to gigabits/sec; avg. ~20 Mbps).

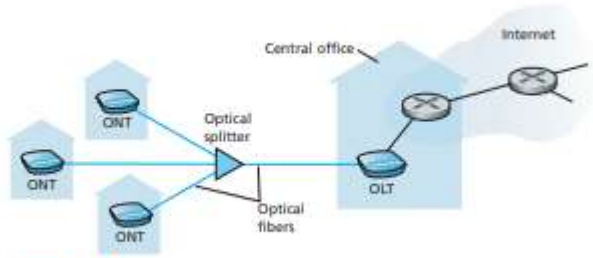

Figure 1.7 • FTTH Internet access

- **Satellite/Dial-up:** For rural areas (slow speeds).

b. Enterprise/Home LANs:

Figure 1.9 ♦ A typical home network

- **Ethernet:** Wired LAN using switches (100 Mbps – 10 Gbps).
- **WiFi (IEEE 802.11):** Wireless LAN (up to 54+ Mbps shared).
  - Short range (~tens of meters).

c. Wide-Area Wireless:

- **3G/4G/LTE:** Cellular networks for mobile devices.
  - Long range (~tens of km).
  - Speeds: 3G (>1 Mbps), LTE (10+ Mbps).

---

3. Physical Media (Guided):

- **Twisted Pair:** Copper wires (UTP for LANs/DSL).
- **Coaxial Cable:** Used in cable internet (HFC).
- **Fiber Optics:** High-speed, long-distance backbone.

4. Key Concepts:

- **Edge Router:** First router connecting an access network to the core.
- **Shared vs. Dedicated Media:**
  - Cable/WiFi: Shared bandwidth.
  - DSL/FTTH: Dedicated links.

Why it matters: The network edge is where end users/devices connect to the Internet via diverse technologies (DSL, cable, WiFi, cellular), each with trade-offs in speed, cost, and scalability. Physical media (copper, fiber, radio) enable these connections.

The Network Core

1. Core Function:

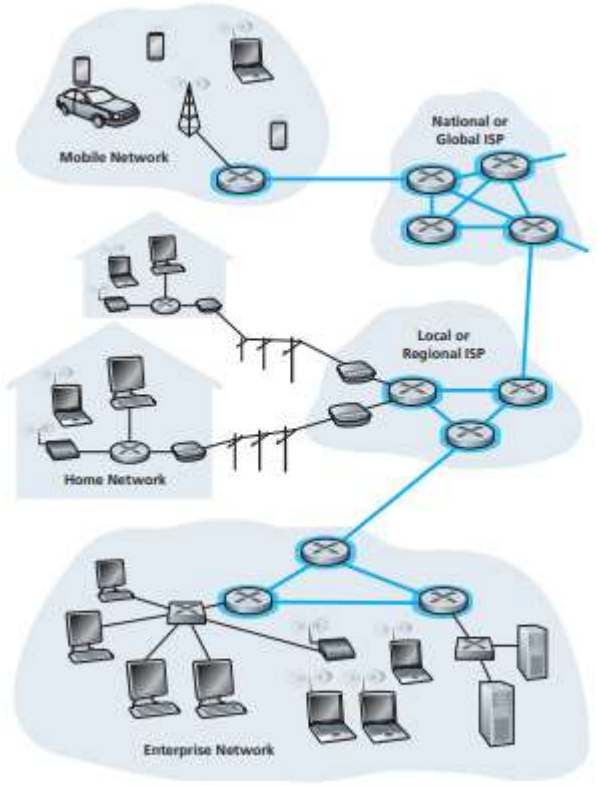

Figure 1.10 • The network core

Interconnects end systems via a mesh of **routers** and **links**, handling data transfer across the Internet.

---

2. Packet Switching:

- **Data Segmentation:** Messages split into **packets** for transmission.
- **Store-and-Forward:** Routers wait for entire packet before forwarding → **Delay** = $\mathbf{N \times (L/R)}$ (for N links, packet size L, link rate R).

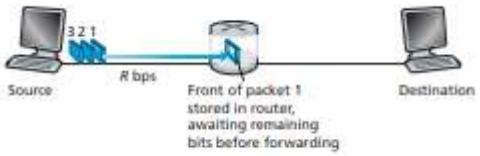

Figure 1.11 ♦ Store-and-forward packet switching

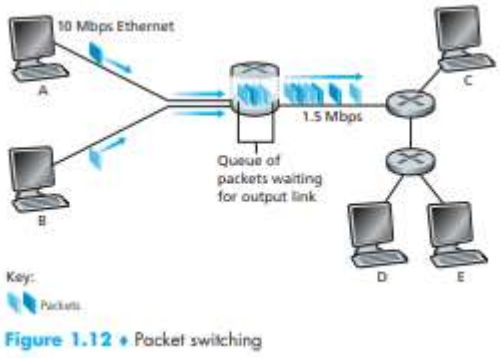

Figure 1.12 ♦ Packet switching

- **Queuing & Loss:**
    - Packets wait in **output buffers** if links are busy.
    - **Packet loss** occurs if buffers overflow.
- **Routing:**
    - **Forwarding Tables:** Routers use destination IP addresses to choose outgoing links.
    - **Routing Protocols:** Automatically configure paths (e.g., shortest path).

---

3. Circuit Switching:

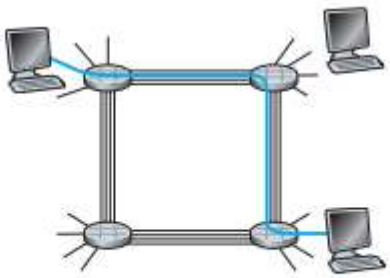

Figure 1.13 ♦ A simple circuit-switched network consisting of four switches and four links

- **Dedicated Path:** Resources (bandwidth) reserved for entire session (e.g., phone call).
- **Multiplexing:**
    - **FDM:** Divides frequency bands.
    - **TDM:** Divides time slots.

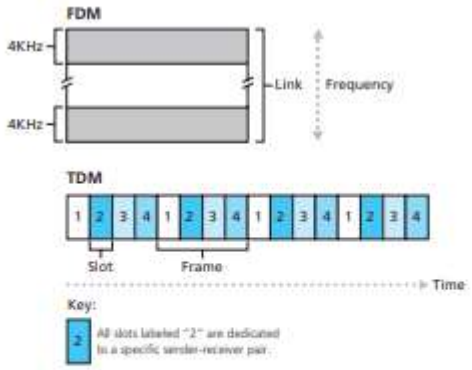

Figure 1.14 ♦ With FDM, each circuit continuously gets a fraction of the bandwidth. With TDM, each circuit gets all of the bandwidth periodically during brief intervals of time (that is, during slots)

- **Inefficiency:** Idle resources during silence (e.g., pauses in calls).
- **Delay Example:** File transfer = Setup time + (File Size/Circuit Rate) Circuit RateFile Size.

---

4. Packet vs. Circuit Switching:

Packet Switching	Circuit Switching

Packet Switching	Circuit Switching
No reservation; shared links	Dedicated path reserved
Variable delay (queuing)	Constant rate/performance
Efficient for bursty traffic	Wastes bandwidth during idle
Dominates modern Internet	Used in traditional telephony

---

5. Network of Networks (Internet Structure):

- **Hierarchy of ISPs:**
    - **Access ISPs** (local) → **Regional ISPs** → **Tier-1 ISPs** (global, e.g., AT&T, NTT).
- **Interconnection:**
    - **Peering:** ISPs connect directly (settlement-free).
    - **IXPs (Internet Exchange Points):** Hubs for ISP peering.
    - **Multi-homing:** ISPs connect to multiple providers for redundancy.
- **Content Providers (e.g., Google):**
    - Deploy private networks, bypass tiers via peering/IXPs.

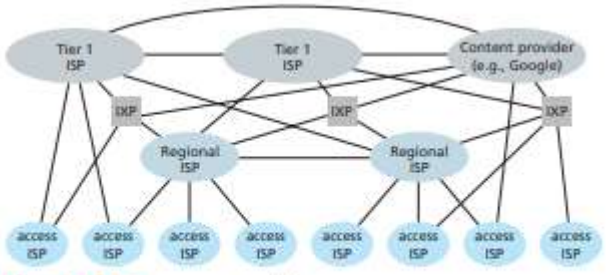

Figure 1.15 ♦ Interconnection of ISPs

Key Takeaways:

- **Internet Core = Packet-Switched.**
- **Routers** forward packets using **forwarding tables**.
- **Circuit switching** inefficient for data traffic.
- **Internet structure:** Multi-tiered ISP hierarchy + peering/IXPs.

Why it matters: The core's packet-switched design enables scalable, efficient data transfer, while the tiered ISP ecosystem ensures global connectivity.

Delay, Loss, and Throughput in Packet-Switched Networks

1. Types of Delay

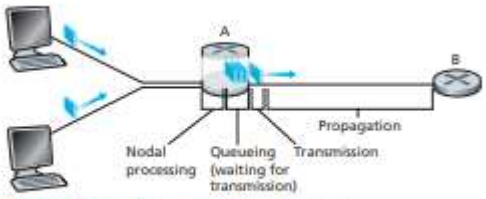

Figure 1.16 • The nodal delay at router A

- **Processing Delay (`d_proc`):**
    - Time to inspect packet headers, check errors, and route packets.
    - Typically microseconds.
- **Queuing Delay (`d_queue`):**
    - Time spent in router buffers waiting for transmission.
    - Depends on traffic intensity (`La/R`); **unbounded if** `La/R > 1`.
    - Highly variable (μs to ms).
- **Transmission Delay (`d_trans`):**

    - Time to push all packet bits onto the link: `L / R` (packet size ÷ link rate).
- **Propagation Delay (`d_prop`):**
    - Time for a bit to travel from sender to receiver: `distance ÷ propagation speed` (≈ speed of light).
- **Total Nodal Delay:**

  `d_nodal = d_proc + d_queue + d_trans + d_prop`

2. Packet Loss

- Occurs when **router buffers overflow** due to congestion.
- Lost packets may be **retransmitted** by end systems (e.g., TCP).
- Loss probability ↑ as traffic intensity (`La/R`) approaches 1.

3. End-to-End Delay

- Sum of delays across all routers and links:

  `d_end-end = Σ (d_proc + d_queue + d_trans + d_prop)` for all nodes.
- **Traceroute** measures round-trip delays to each router.

4. Throughput

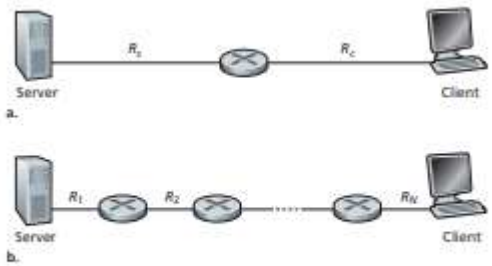

Figure 1.19 • Throughput for a file transfer from server to client

- **Instantaneous throughput:** Current data reception rate (bps).
- **Average throughput:** Total data ÷ transfer time (`F / T`).
- **Bottleneck Principle:**
    - End-to-end throughput = `min{R1, R2, ..., RN}` (slowest link rate).

- In access networks: `min{R_server, R_client}`.
- **Shared Links:** Throughput drops if multiple flows congest a link (e.g., 10 flows sharing 5 Mbps → 500 kbps each).

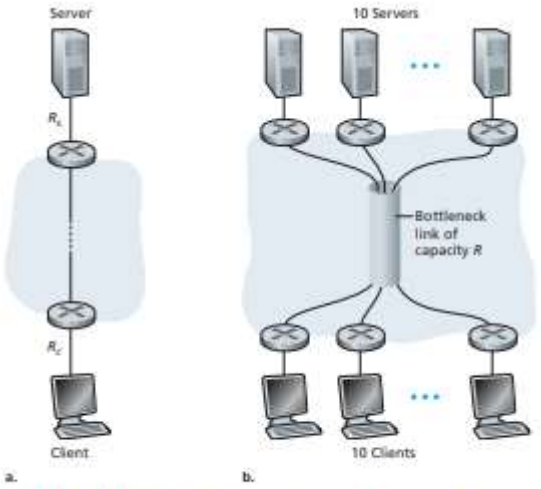

Figure 1.20 ♦ End-to-end throughput: (a) Client downloads a file from server; (b) 10 clients downloading with 10 servers

---

Key Formulas

- **Traffic Intensity:** `La / R` (must be ≤ 1 to avoid infinite queues).
- **Transmission Delay:** `d_trans = L / R`.
- **Propagation Delay:** `d_prop = distance / propagation_speed`.

Why It Matters

- **Delay/Loss** impact real-time apps (VoIP, gaming).
- **Throughput** determines file transfer times.
- **Bottlenecks** often occur at access links (e.g., home WiFi).

Summary: Delays arise from processing, queuing, transmission, and propagation. Packet loss occurs during congestion. Throughput is capped by the slowest link or shared resources. Design for $La/R \leq 1$!

Reference Models,

1. OSI Model (7 Layers)

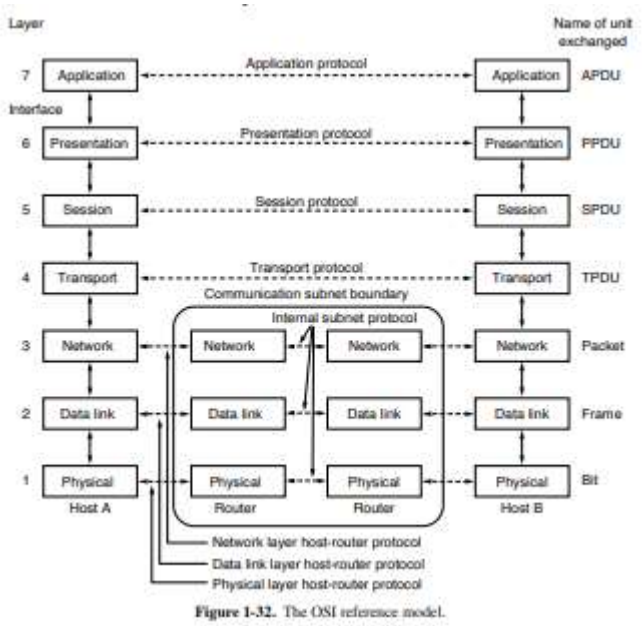

Figure 1-32. The OSI reference model.

Purpose: Theoretical framework for standardization.

Layers:

1. **Physical:** Transmits raw bits over physical media.
2. **Data Link:** Error-free frame transfer between nodes (e.g., MAC addresses).
3. **Network:** Routing packets across networks (e.g., IP addresses).
4. **Transport:** Reliable end-to-end data delivery (e.g., TCP/UDP).
5. **Session:** Manages connections/dialogue control.
6. **Presentation:** Data translation, encryption, compression.
7. **Application:** User-facing services (e.g., HTTP, FTP).

Key Contributions:

- Clear distinction between **services**, **interfaces**, and **protocols**.
- Strict layering minimizes cross-layer dependencies.

---

2. TCP/IP Model (4 Layers)

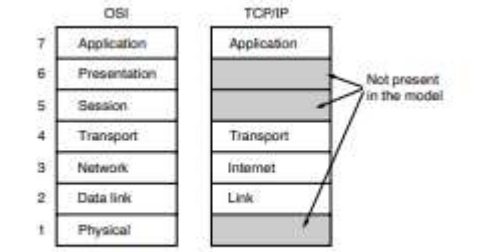

Figure 1-33. The TCP/IP reference model.

Purpose: Practical implementation (basis of the Internet).

Layers:

1. **Link Layer:** Interface to physical networks (e.g., Ethernet, WiFi).
2. **Internet Layer:** Routes packets across networks using **IP** and **ICMP**.
3. **Transport Layer:** End-to-end communication via **TCP** (reliable) or **UDP** (unreliable).
4. **Application Layer:** Combines OSI's application, presentation, and session layers (e.g., DNS, HTTP, SMTP).

Design Goals:

- Network survivability (works even if parts fail).
- Flexibility for diverse applications.
- Seamless inter-network communication.

---

3. Key Differences

Feature	OSI Model	TCP/IP Model
Layers	7 (strict separation)	4 (pragmatic grouping)
Session/Presentation	Dedicated layers	Merged into Application
Approach	Theory-first	Implementation-first
Adoption	Limited (academic)	Dominant (Internet)

4. Hybrid Model (Modern Internet)

- **5 Layers:** Physical → Data Link → Network → Transport → Application.
- **Why:** Combines TCP/IP's practicality with OSI's lower-layer clarity.

Why It Matters:

- OSI provides a **teaching tool** for concepts.
- TCP/IP reflects **real-world Internet architecture**.
- Modern networks use a **blended approach**.

 Key Insight: TCP/IP succeeded due to its simplicity, flexibility, and working protocols. OSI's rigid layers (e.g., separate session/presentation) proved unnecessary in practice.

Guided Transmission Media

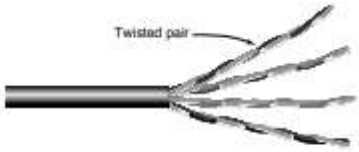

Figure 2-1. Category 5e UTP cable with four twisted pairs. These cables can be used for local area networks.

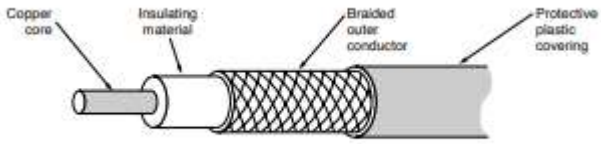

Figure 2-2. A coaxial cable.

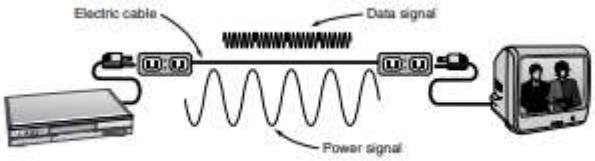

Figure 2-3. A network that uses household electrical wiring.

1. Key Guided Media

Medium	Description	Bandwidth	Use Cases
Twisted Pair	Insulated copper wires twisted in pairs (Cat 5e/6/7/8). UTP (unshielded) vs. STP (shielded).	Up to 10 Gbps (Cat 6/7)	Ethernet LANs, telephone lines.
Coaxial Cable	Copper core + insulator + braided mesh + sheath.	Up to 6 GHz	Cable TV, broadband Internet.
Fiber Optics	Glass core + cladding + protective coating. Light pulses transmit data.	50+ Tbps (theoretical); 100 Gbps (practical)	Backbone networks, FTTH, long-haul.
Power Lines	Uses electrical wiring; data overlaid on power signals.	~500 Mbps	Home LANs, IoT (limited adoption).

---

2. Fiber Optics Deep Dive

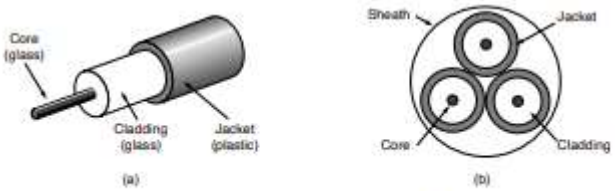

Figure 2-6. (a) Side view of a single fiber. (b) End view of a sheath with three fibers.

- **How it works:**
    - Light source (laser/LED) → Glass fiber → Detector (converts light to electrical signal).
    - **Total Internal Reflection:** Traps light in core (critical angle principle).
- **Types:**
    - **Multimode:** Thicker core (50μm); shorter distances (<15 km); used in LANs.
    - **Single-mode:** Thin core (10μm); longer distances (100+ km); backbone networks.
- **Bands:** 0.85 μm (short-range), 1.30 μm & 1.55 μm (low attenuation, long-haul).
- **Advantages:**
    - Extreme bandwidth, low attenuation (~0.2 dB/km), EMI immunity, security.

---

3. Copper vs. Fiber

Metric	Copper (Twisted Pair/Coax)	Fiber Optics
Bandwidth	Up to 10 Gbps	100+ Gbps
Distance	Km (needs repeaters)	100+ km (no repeaters)
EMI/Security	Vulnerable	Immune
Cost	Low	Higher (installation)

Metric	Copper (Twisted Pair/Coax)	Fiber Optics
Weight/Size	Heavy/bulky	Light/thin

---

4. Key Concepts

- **Bandwidth:** Carrying capacity of a medium (Hz).
- **Full/Half-Duplex:** Simultaneous vs. alternating transmission.
- **Persistent Storage "Bandwidth":**
  - Physically moving tapes/disks (e.g., 100 PB via truck) → Effective bandwidth >> networks.
  - Moral: "*Never underestimate the bandwidth of a station wagon full of tapes!*"

Why It Matters

- **Twisted pair** dominates access networks (cost-effective).
- **Fiber** dominates backbones (speed, distance, future-proofing).
- **Power-line** and **coax** fill niche roles (e.g., home networking).

 Insight: Fiber's theoretical limits dwarf copper, but deployment costs and electronic bottlenecks constrain practical use.

Wireless Transmission

1. Wireless Communication Basics

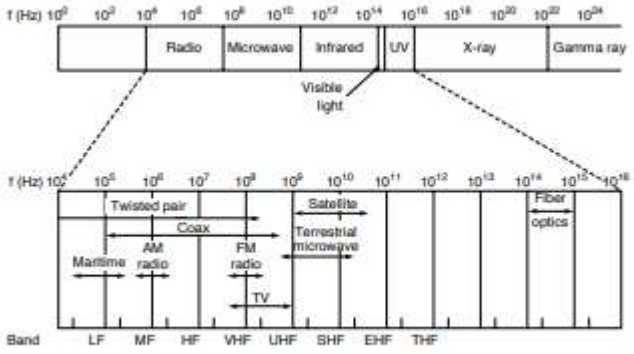

Figure 2-8. The electromagnetic spectrum and its uses for communication.

- **Principle:** Electromagnetic waves generated by electron movement carry data through space/vacuum.
- **Key Equation:** $\lambda f = c$
  - λ (wavelength), f (frequency), c (speed of light $\approx 3 \times 10^8$ m/s).
  - **Rule of thumb:** $\lambda(\text{meters}) \times f(\text{MHz}) \approx 300$.
- **Spectrum Bands** (Fig. 2-8):
  - **Radio → Visible Light:** Used for data transmission (modulation of amplitude/frequency/phase).
  - **Higher Frequencies (UV, X-rays):** Unsuitable (dangerous, poor propagation).

---

2. Spread Spectrum Techniques

Technique	How It Works	Applications	Advantages
Frequency Hopping (FHSS)	Rapidly switches carrier frequency hundreds of times/sec.	Bluetooth, military comms, older 802.11.	Anti-jamming, resists fading/interference.
Direct Sequence (DSSS)	Spreads signal using code sequence over wide band.	3G/CDMA, GPS, older 802.11b.	Efficient spectrum sharing (CDMA), interference tolerance.
Ultra-Wideband	Sends low-energy rapid pulses over very wide	Short-range IoT, radar imaging, location	Minimal interference to narrowband signals, high data

Technique	How It Works	Applications	Advantages
(UWB)	band (≥500 MHz).	tracking.	rates.

 Fun Fact: Hedy Lamarr co-invented frequency hopping during WWII to secure torpedo communications!

---

3. Why Spread Spectrum?

- **Resilience:** Tolerates interference, fading, and jamming.
- **Security:** Hard to detect/intercept (especially FHSS).
- **Efficiency:** Enables multiple signals to share spectrum (e.g., CDMA).

---

4. Key Concepts

- **Bandwidth ∝ Data Rate:** Higher frequency bands (e.g., microwave, fiber) offer massive bandwidth.
- **UWB Definition:** Bandwidth ≥500 MHz or ≥20% of center frequency.
- **Trade-offs:**
  - *FHSS/DSSS:* Robust but complex.
  - *UWB:* High speed/short-range; minimal interference.

Why It Matters: Wireless tech enables mobile networks, IoT, and secure communications. Techniques like FHSS/DSSS/UWB optimize spectrum use while combating real-world challenges like interference and fading.

Unit-II

It's a pleasure to connect, Dr. Obula Raju D! □□

Your credentials are impressive:

- **M.Tech in Computer Networks & Information Security**
- **Assistant Professor, CSE at Andhra Engineering College**
- **Subject Code: 23A05501T**
- **Version :V09072025**

UNIT II:

The Data Link Layer, Access Networks, and LANs

Data Link Layer Design Issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols (Textbook 1) Introduction to the Link Layer, Error-Detection and -Correction Techniques, Multiple Access Links and Protocols, Switched Local Area Networks, Link Virtualization: A Network as a Link Layer, Data Center Networking, Retrospective: A Day in the Life of a Web Page (Packet) (Textbook 2)

Data Link Layer Design Issues

1. **Service Interface:** Provides defined services to the network layer (unacknowledged connectionless, acknowledged connectionless, acknowledged connection-oriented).
2. **Framing:** Breaks the physical layer's bit stream into manageable frames using techniques like:
    - Byte count (prone to synchronization errors).
    - Flag bytes with byte stuffing.
    - Flag bits with bit stuffing.
    - Physical layer coding violations.
    - Often combines preamble + length for robustness (e.g., Ethernet, Wi-Fi).
3. **Error Control:** Ensures reliable delivery over unreliable channels:
    - Uses checksums/error-detecting codes to identify corrupted frames.

- Relies on acknowledgements (ACKs/NAKs), timers, and sequence numbers.
- Retransmits lost or corrupted frames detected by timeouts or NAKs.
- Sequence numbers prevent duplicate delivery.

4. **Flow Control:** Prevents fast senders from overwhelming slow receivers:
    - Primarily uses feedback-based mechanisms (e.g., receiver grants permission/window).
    - Rate-based control is less common here (more typical in transport layer).
    - Ensures the receiver can process incoming frames.

In essence: The Data Link Layer's core responsibility is to transform the physical layer's potentially error-prone bit stream into a reliable, manageable link for the network layer. It achieves this through framing, error detection/correction, retransmission mechanisms, and flow control, often implemented using feedback, timers, and sequence numbers. While principles appear in higher layers, they are foundational and often simplest here.

Error Detection and Correction

3.2 Error Detection and Correction

Core Concepts

1. **Why Needed**:
    - Channels vary in reliability (e.g., fiber optics = low errors; wireless/aging copper = high errors).
    - **Strategies**:
        - **Error-Correcting Codes (FEC)**: Add redundancy to **reconstruct** original data at receiver.
        - **Error-Detecting Codes**: Add redundancy to **detect** errors → trigger retransmission.

2. **Error Models**:
    - **Single-bit errors**: Isolated flips (e.g., thermal noise).
    - **Burst errors**: Consecutive bits corrupted (e.g., wireless fade).
    - **Erasures**: Known error locations (easier to correct).

Error-Correcting Codes (FEC)

Used for noisy channels (e.g., wireless) where retransmission is impractical.

1. Hamming Codes

- **Type**: Linear systematic block code.
- **Mechanism**: Adds parity bits to create codewords. Corrects **single-bit errors**.
- **Example**: (11,7) code: 7 data bits + 4 check bits.
- **Key**: Uses **syndrome** to locate/flip erroneous bit.
- **Limitation**: Distance = 3 → detects 2 errors, corrects 1.

Maths

- **Data**: `1001` (4 bits)
- **Parity Bits**:
    - Position 1: Bits 1,3,5,7 → `? _ 1 _ 0 _ 1` → `0` (even)
    - Position 2: Bits 2,3,6,7 → `_ ? 1 _ _ 0 1` → `0` (even)
    - Position 4: Bits 4,5,6,7 → `_ _ _ ? 0 0 1` → `1` (odd)
- **Codeword**: `0011001`
- **Error Correction**:

- Received: `0011101` (bit 5 flipped)
- Syndrome: Check parities → Positions 1 & 4 fail → Error at bit 5 (`1+4=5`).

2. Convolutional Codes

- **Type**: **Not a block code**; output depends on current + past bits (memory).
- **Mechanism**: Encodes streams (e.g., NASA code: rate=1/2, constraint length=7).
- **Decoding**: **Viterbi algorithm** (finds most likely input sequence).
- **Advantage**: **Soft-decision decoding** handles analog signal uncertainty.

3. Reed-Solomon (RS) Codes

- **Type**: Linear block code operating on **m-bit symbols** (e.g., bytes).
- **Strengths**: Excellent for **burst errors** (e.g., 128-bit burst correctable in RS(255,233)).
- **Usage**: CDs, DVDs, satellite, DSL.

Math

- **Data**: Points (1,1), (2,3) → Line $y=2x-1$ $y=2x-1$
- **Redundancy**: Add points (3,5), (4,7) on the line.
- **Error**: Received (3,5), **(3,6)**, (4,7)
- **Correction**: Fit line → (3,6) deviates; recover as (3,5).

4. Low-Density Parity Check (LDPC)

- **Type**: Linear block code with sparse parity-check matrix.
- **Strengths**: Near-optimal for large blocks; outperforms older codes.
- **Usage**: 10G Ethernet, 802.11, DVB.

Key Insight:

- **Parity/Hamming**: Linear algebra (XOR operations).
- **CRC**: Polynomial division (mod 2).
- **Reed-Solomon**: Polynomial interpolation over finite fields.

Error-Detecting Codes

Used for reliable channels (e.g., fiber) where retransmission is efficient.

1. Parity

- Single parity bit → detects **single-bit errors**.
- **Weakness**: Misses burst errors; detection probability = 0.5 for long bursts.
- **Improvement**: **Interleaving** (spread bits across rows/columns).

Maths:

- **Data**: `1010001` (7 bits, three 1s → odd)
- **Even Parity**: Add `1` → `10100011` (four 1s, even)
- **Error Detection**: If received as `10101011`, parity fails (five 1s → odd).

2. Checksums

- **Internet Checksum (IP)**: 16-bit one's complement sum of words.
- **Weakness**: Fails with swapped data/zero additions.
- **Better**: **Fletcher's checksum** (adds positional sensitivity).

3. Cyclic Redundancy Check (CRC)

- **Strongest & Most Common** (Ethernet, 802.11, USB).
- **Mechanism**: Treats data as polynomial → divides by generator polynomial $G(x)$.
- **Steps**:

1. Append r zeros to data (r=degree of $G(x)$).
2. Divide by $G(x)$ → get remainder.
3. Append remainder to data.

- **Detection Capabilities**:
  - All single-bit errors.
  - All bursts ≤ r bits.
  - Odd-bit errors (if $G(x)$ includes $x+1$).

Standard: IEEE 802 CRC-32 ($G(x)=x32+x26+...+1$).

Maths

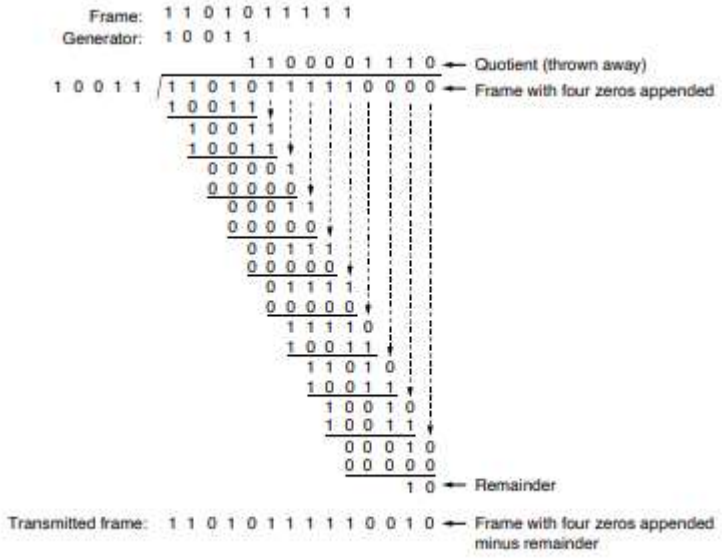

Figure 3-9. Example calculation of the CRC.

Generator: G(x)=x3+x+1$G(x)=x^3+x+1$ → 1011

Data: 11010011101100

- Append 3 zeros → 11010011101100000

Modulo-2 Division:

text

```
11010011101100000 ÷ 1011
Remainder = 010 (use polynomial long division)
```

Transmitted Frame: 11010011101100010

Verification: Receiver divides by 1011; remainder ≠ 0 → error detected.

Key Trade-offs

Approach	Best For	Overhead	Complexity
FEC	Noisy channels	High	High
Detection + Retx	Reliable channels	Low	Low

Design Insight:

- FEC (e.g., LDPC/RS) suits wireless links.
- CRC dominates wired links (efficiency + robustness).

Elementary Data Link Protocols

1. Initial Assumptions

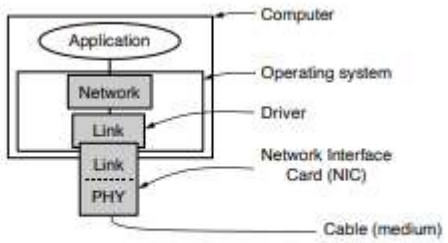

Figure 3-10. Implementation of the physical, data link, and network layers.

- **Independent Layers**: Physical, data link, network layers are separate processes (e.g., NIC handles physical + part of data link).
- **Unidirectional**: Data flows only **A → B** (simplex).
- **Reliable Machines**: No crashes; only communication errors handled.
- **Network Layer Packets**: Treated as pure data by data link layer.

---

2. Basic Transmission & Receipt

- **Framing**: Data link layer encapsulates packets into frames (header + payload + trailer).
- **Checksum**: Validated on receipt (e.g., CRC).
- **Events**:
  - wait_for_event(&event) handles:
    - frame_arrival (valid frame) → Process and pass to network layer.
    - cksum_err (corrupted frame) → Discard.
- **No frame headers** exposed to network layer (ensures layer independence).

---

3. Simplex Protocols

Protocol 1: Utopia (No Flow/Error Control)

- **Assumptions**: Perfect channel, infinite buffers, always-ready network layers.
- **Sender**:

c

```
while (true) {
   from_network_layer(&packet);  // Get packet
   s.info = packet;              // Build frame
   to_physical_layer(&s);        // Send frame
}
```

- **Receiver**:

c

```
while (true) {
   wait_for_event(&event);       // Only event: frame_arrival
   from_physical_layer(&r);      // Receive frame
   to_network_layer(&r.info);    // Pass packet to network layer
}
```

- **Flaw**: No handling of overflow or errors.

Protocol 2: Stop-and-Wait (Flow Control)

- **Solves**: Receiver overflow.
- **Mechanism**: Sender waits for **ACK** after each frame.
- **Sender**:

c

```
while (true) {
   from_network_layer(&packet);
   s.info = packet;
   to_physical_layer(&s);
   wait_for_event(&event);       // Wait for ACK
}
```

- **Receiver**:

c

```
while (true) {
   wait_for_event(&event);       // Frame arrives
   from_physical_layer(&r);
   to_network_layer(&r.info);
   to_physical_layer(&ack_frame);  // Send ACK
}
```

- **Channel**: Half-duplex suffices (alternating frames/ACKs).

Protocol 3: ARQ (Error Correction + Flow Control)

- **Solves**: Lost/damaged frames and duplicates.
- **Tools**:
    - **1-bit sequence numbers** (0/1).
    - **Timer** for retransmission.
- **Sender**:
  c

```
next_frame_to_send = 0;
while (true) {
   from_network_layer(&packet);
   s.seq = next_frame_to_send;    // Set sequence number
   s.info = packet;
   to_physical_layer(&s);
   start_timer();              // Start/reset timer
   wait_for_event(&event);
   if (event == frame_arrival) {  // Valid ACK
      from_physical_layer(&r);
      next_frame_to_send = inc(next_frame_to_send); // Toggle 0↔1
   }
   // Timeout → retransmit without toggling seq
}
```

- **Receiver**:
  c

```
frame_expected = 0;
while (true) {
   wait_for_event(&event);
   if (event == frame_arrival) {
      from_physical_layer(&r);
      if (r.seq == frame_expected) { // New frame
         to_network_layer(&r.info);
         frame_expected = inc(frame_expected);
      }
      s.ack = 1 - frame_expected; // ACK last valid frame
      to_physical_layer(&s);      // Send ACK (even for dupes)
   }
}
```

- **Key Behaviors**:

- **Receiver** discards duplicates (wrong seq) but **re-ACKs** last valid frame.
- **Sender** retransmits on timeout (lost frame/ACK).

Why These Protocols Matter

- **Utopia**: Baseline for ideal conditions.
- **Stop-and-Wait**: Introduces **flow control** via ACKs.
- **ARQ (Protocol 3)**: Adds **error recovery** (sequence numbers + timers).
- **Evolution**: Layered improvements addressing real-world issues (noise, delays).

Sliding Window Protocols:

Core Concept

- **Bidirectional Communication**: Allows simultaneous data flow in both directions (full-duplex).
- **Sequence Numbers**: Frames use *n*-bit sequence numbers (0 to 2ⁿ - 1).
- **Windows**:
  - **Sender Window**: Frames sent but not yet acknowledged.
    - Advances when ACKs arrive; buffers unacknowledged frames.
    - If full, blocks network layer until space frees.
  - **Receiver Window**: Frames eligible for acceptance.
    - Delivers frames to network layer **in order** (lower-edge-first).
    - Discards out-of-window frames but sends ACKs.

Key Mechanisms

- **Sender Operation**:
    - Assigns sequence numbers to outgoing frames.
    - Buffers frames until ACKed; retransmits on timeout.
    - Window slides (advances) as ACKs arrive.
- **Receiver Operation**:
    - Accepts frames within window; buffers out-of-order frames.
    - Delivers **in-order** frames to network layer.
    - Always ACKs received frames (guides sender).

Window Size & Efficiency

- **Size = 1** → Stop-and-Wait (inefficient for high-latency links).
- **Larger Windows**:
    - Allow multiple in-flight frames → better link utilization.
    - Require more buffers at sender/receiver.
- **Sequence Number Space**:
    - Max seq = 2ⁿ - 1 (e.g., 3 bits → seq 0-7).
    - Prevents ambiguity between new vs. retransmitted frames.

Visual Example (Fig 3-15)

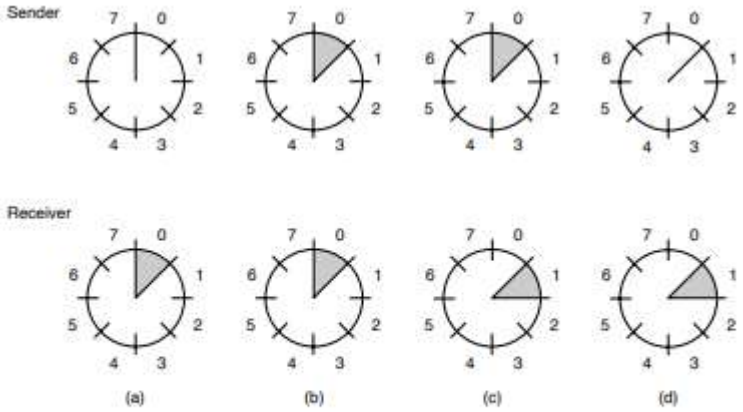

Figure 3-15. A sliding window of size 1, with a 3-bit sequence number. (a) Initially. (b) After the first frame has been sent. (c) After the first frame has been received. (d) After the first acknowledgement has been received.

- **Initial State**: Sender and receiver windows aligned (no frames in transit).
- **After Frame Sent**: Sender window expands; receiver window holds expected frame.
- **After Frame Received**: Receiver delivers in-order frame, slides window.
- **After ACK**: Sender slides window, frees buffer.

Why It Matters

- **Efficiency**: Larger windows minimize idle time (vs. stop-and-wait).
- **Reliability**: ACKs + sequence numbers ensure ordered, lossless delivery.
- **Flexibility**: Adapts to channel conditions via window size.

Introduction to the Link Layer

5.1 Introduction to the Link Layer

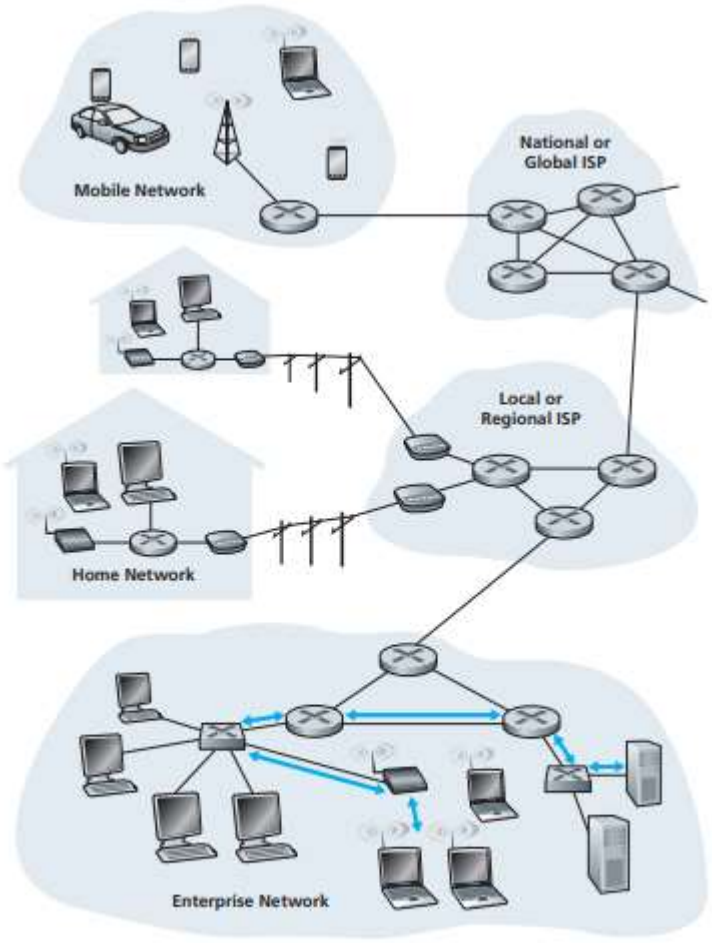

Figure 5.1 ♦ Six link-layer hops between wireless host and server

Key Terminology

- **Node**: Any device running link-layer protocols (hosts, routers, switches, APs).
- **Link**: Communication channel connecting **adjacent nodes** along a path.
- **Datagram Transfer**: Requires traversal over multiple links (e.g., WiFi → Ethernet → router links).

Transport Analogy

- **Tourist** = Datagram
- **Transport Segments** (limo/plane/train) = Links

- **Transport Companies** = Link-layer protocols
- **Travel Agent** = Routing protocol

5.1.1 Link-Layer Services

1. **Framing**:
    - Encapsulates network-layer datagrams into frames.
    - Structure: **Header** + *Data (payload)* + **Trailer**.
    - Frame formats vary by protocol (e.g., Ethernet, WiFi).
2. **Link Access**:
    - **MAC (Medium Access Control)**: Rules for frame transmission.
    - **Point-to-Point Links**: Simple (sender transmits when idle).
    - **Broadcast Links**: Handles *multiple access* (e.g., WiFi, Ethernet).
3. **Reliable Delivery (Optional)**:
    - Uses **acknowledgments (ACKs)** and **retransmissions**.
    - Common for **error-prone links** (e.g., wireless).
    - Often omitted for reliable wired links (fiber, copper).
4. **Error Detection & Correction**:
    - **Detection**: Frame headers include error-detection bits (e.g., CRC).
    - **Correction**: Receiver identifies and fixes bit errors (less common).
    - Implemented in **hardware** for efficiency.

5.1.2 Link-Layer Implementation

Network Adapter (NIC)

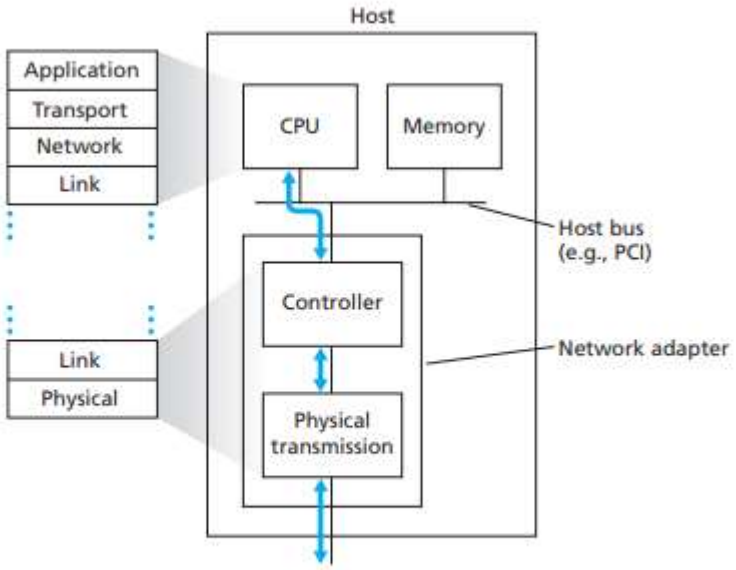

Figure 5.2 ♦ Network adapter: its relationship to other host components and to protocol stack functionality

- **Link-Layer Controller**: Specialized hardware chip handling:
    - Framing, error detection, MAC protocols.
    - Examples: Intel 8254x (Ethernet), Atheros AR5006 (WiFi).
- **Evolution**:
    - Historically plug-in cards (PCI, PCMCIA).
    - Now integrated as **LAN-on-motherboard**.

Hardware-Software Interaction

- **Transmitting**:
    1. Software: Prepares datagram in memory.
    2. Hardware: Encapsulates into frame, transmits via MAC protocol.
- **Receiving**:
    1. Hardware: Receives frame, checks errors, extracts datagram.
    2. Software: Handles interrupts, passes datagram to network layer.

- **Key Insight**: Link layer = **hardware (controller)** + **software (drivers)**.

Why This Matters

- **Local Scope**: Manages data transfer between **adjacent nodes**.
- **Network Layer Reliance**: Depends on link layer for **per-hop delivery**.
- **Efficiency**: Hardware offload boosts speed (e.g., CRC checks).

Error Detection & Correction:

Error Detection & Correction: Student Summary

1. Core Concepts

- **Goal**: Detect/correct bit errors in frames during transmission.
- **EDC Bits**: Appended to data (D) for protection; sent with D.
- **Receiver Challenge**: Decides if received D' matches original D using D' and EDC'.
- **Imperfection**: No technique catches 100% of errors (undetected errors possible).

---

2. Key Techniques

a. Parity Checks

- **Single Parity Bit**:

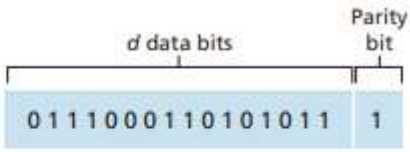

Figure 5.4 ♦ One-bit even parity

- **Even Parity**: Add bit so total 1s = even.
- **Detects**: Odd-bit errors (e.g., 1, 3 flips).
- **Fails**: Even-bit errors (e.g., 2 flips).

- **2D Parity**:

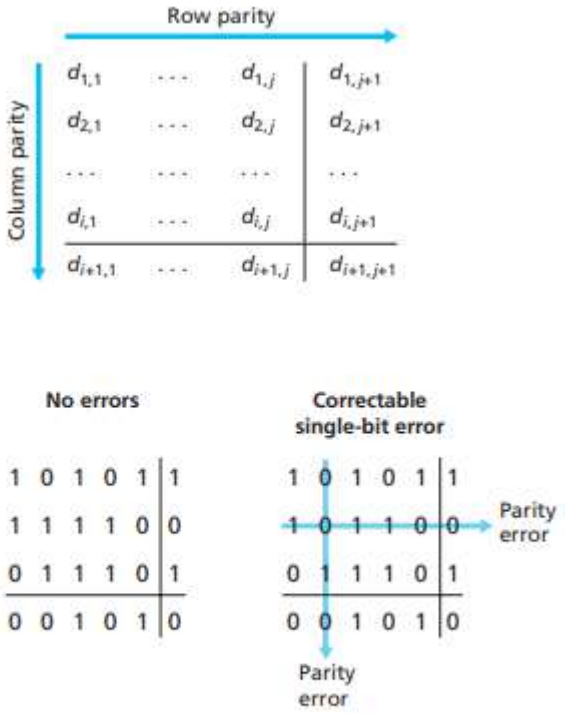

Figure 5.5 ♦ Two-dimensional even parity

- Arrange D in matrix; add row/column parity bits.
- **Detects & Corrects**: Single-bit errors (locate row/col).
- **Detects**: Two errors (but not correct).
- **FEC (Forward Error Correction)**: Enables on-the-spot correction.

b. Checksumming

- **Method**: Treat D as k-bit integers; sum and take complement (e.g., 1s complement).
- **Used In**: Transport layer (TCP/UDP checksum).
- **Pros**: Low overhead (e.g., 16 bits in TCP).
- **Cons**: Weaker error detection vs. CRC.
- **Why Transport Layer?**: Simple/fast for software implementation.

c. Cyclic Redundancy Check (CRC)

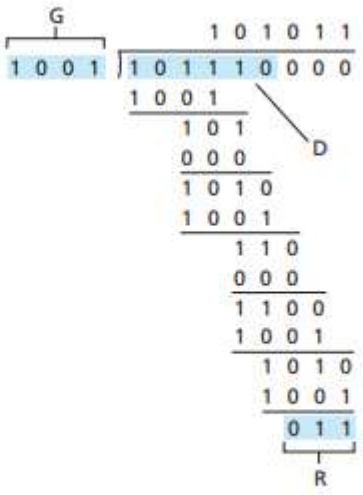

Figure 5.7 ♦ A sample CRC calculation

- **Method**:
  - Treat D as polynomial.
  - Agree on generator polynomial G (r+1 bits, e.g., CRC-32).
  - Compute r-bit remainder R: `R = D * 2^r mod G` (mod-2 arithmetic).
  - Append R to D → transmit `D || R`.
- **Receiver**: Divides received bits by G; non-zero remainder = error.
- **Strengths**:

    - Detects all burst errors < r+1 bits.
    - Detects any odd-bit errors.
    - High detection probability for longer bursts (1 - 0.5^r).
- **Used In**: Link layer (Ethernet, WiFi) due to hardware efficiency.

---

3. Key Trade-offs

Technique	Layer	Strengths	Weaknesses
Single Parity	Link	Simple, low overhead	Misses even-bit errors
2D Parity	Link	Corrects single errors	Limited to small bursts
Checksum	Transport	Fast in software	Less robust than CRC
CRC	Link (hardware)	Robust, efficient in hardware	Higher compute overhead

---

Multiple Access Protocols

Multiple Access Protocols: Student Notes

1. Core Problem: Shared Channel Access

- **Broadcast Links**: Multiple nodes share a single channel (e.g., Ethernet, WiFi).

- **Challenge**: Avoid **collisions** when ≥2 nodes transmit simultaneously → wasted bandwidth.
- **Goal**: Efficient, fair, decentralized protocols for channel sharing.

2. Three Protocol Categories

a. Channel Partitioning

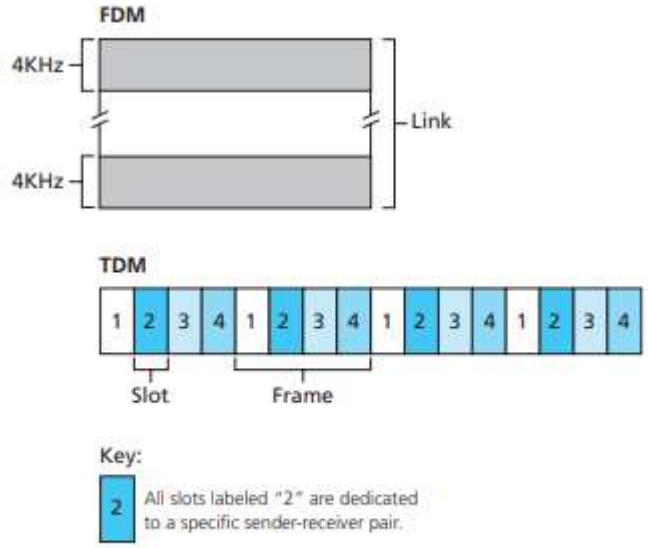

Figure 5.9 ♦ A four-node TDM and FDM example

- **TDM (Time Division Multiplexing)**:
  - Divide time into **slots**; each node gets fixed slot.
  - *Pros*: No collisions, fair.
  - *Cons*: Wastes bandwidth if node idle; delays even when alone.
- **FDM (Frequency Division Multiplexing)**:
  - Split channel into **frequency bands**; assign bands to nodes.
  - Same pros/cons as TDM.
- **CDMA (Code Division Multiple Access)**:
  - Assign unique **codes**; nodes transmit simultaneously.
  - Used in cellular/WiFi (details in Chap 6).

b. Random Access

- Nodes transmit at full rate; retry after collisions with random delays.
    - **Slotted ALOHA**:
        - Time divided into slots; transmit at slot start.
        - Collision → retry with probability `p`.
        - **Max efficiency: 37%** (37% slots successful).
    - **Pure ALOHA**:
        - No slots; transmit immediately.
        - Collision → retry after random delay.
        - **Max efficiency: 18.5%** (half of slotted).
    - **CSMA (Carrier Sense Multiple Access)**:
        - "**Listen before talk**": Sense channel idle before transmitting.
        - *Problem*: **Propagation delay** → hidden collisions (Fig 5.12).

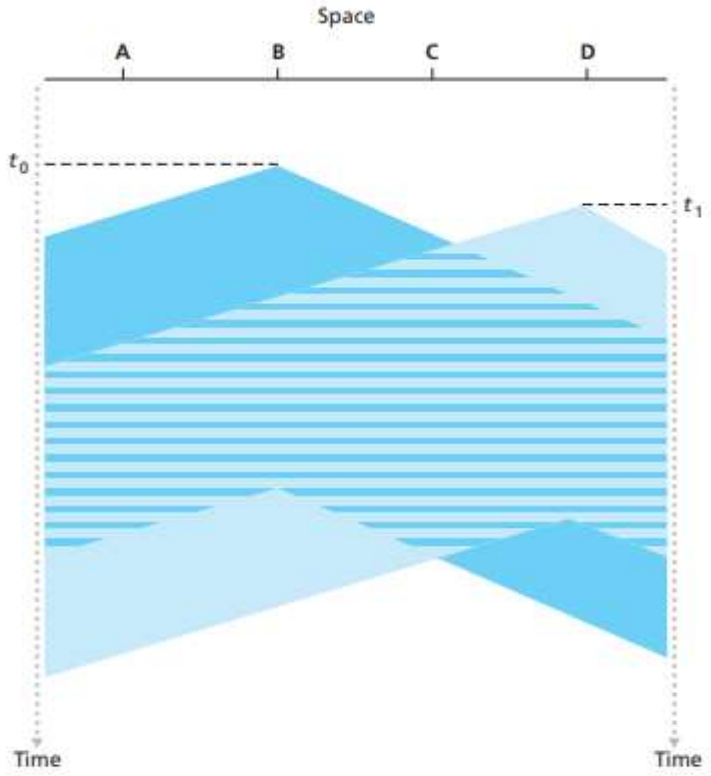

Figure 5.12 ♦ Space-time diagram of two CSMA nodes with colliding transmissions

- **CSMA/CD (Collision Detection)**:
  - "**Listen while talking**": Abort on collision detection.
  - **Binary Exponential Backoff**:
    - After `n` collisions, wait random time ∈ `[0, 2ⁿ−1]` × slot time.
    - Example: Ethernet uses `K × 512` bit times.
  - **Efficiency**: ≈ `1/(1 + 5d_prop/d_trans)` (depends on delay/frame size).

c. Taking Turns

- **Polling**:
  - **Master node** polls slaves in round-robin.
  - *Pros*: No collisions, high efficiency.

- *Cons*: Master failure → system crash; polling delay.
- Used in Bluetooth.

- **Token Passing**:
  - **Token** circulates; node with token transmits.
  - *Pros*: Decentralized, efficient.
  - *Cons*: Token loss → system crash.
  - Examples: FDDI, Token Ring.

3. DOCSIS Case Study (Cable Internet)

- **Hybrid Protocol** combining all three categories:
  - **FDM**: Separate upstream/downstream channels.
  - **TDM-like**: Upstream divided into **mini-slots** allocated by CMTS.
  - **Random Access**: Modems request slots via contention mini-slots (collisions → binary backoff).
- **CMTS Control**:
  - Sends **MAP messages** to grant slots.
  - Ensures collision-free data transmission.

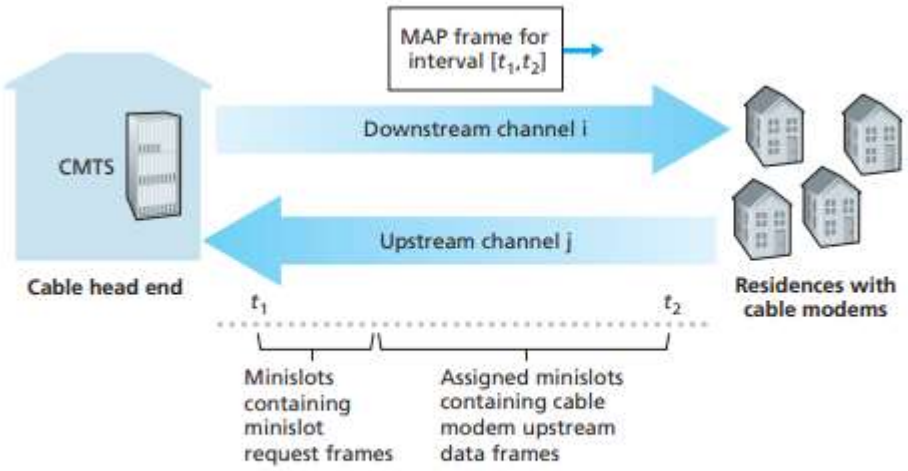

Figure 5.14 ♦ Upstream and downstream channels between CMTS and cable modems

Key Insights

Protocol Type	Examples	Best For	Limitations
Partitioning	TDM, FDM, CDMA	Predictable traffic	Low utilization
Random Access	ALOHA, CSMA/CD	Bursty traffic	Collisions reduce efficiency
Taking Turns	Polling, Token	Balanced load	Single point of failure

- □ **Takeaway**: No "best" protocol; choice depends on network scale, traffic, and fault tolerance. Ethernet (CSMA/CD) dominates wired LANs; DOCSIS illustrates real-world hybrid design.

Switched Local Area Networks

Switched LANs: Student Notes

5.4.1 Link-Layer Addressing & ARP

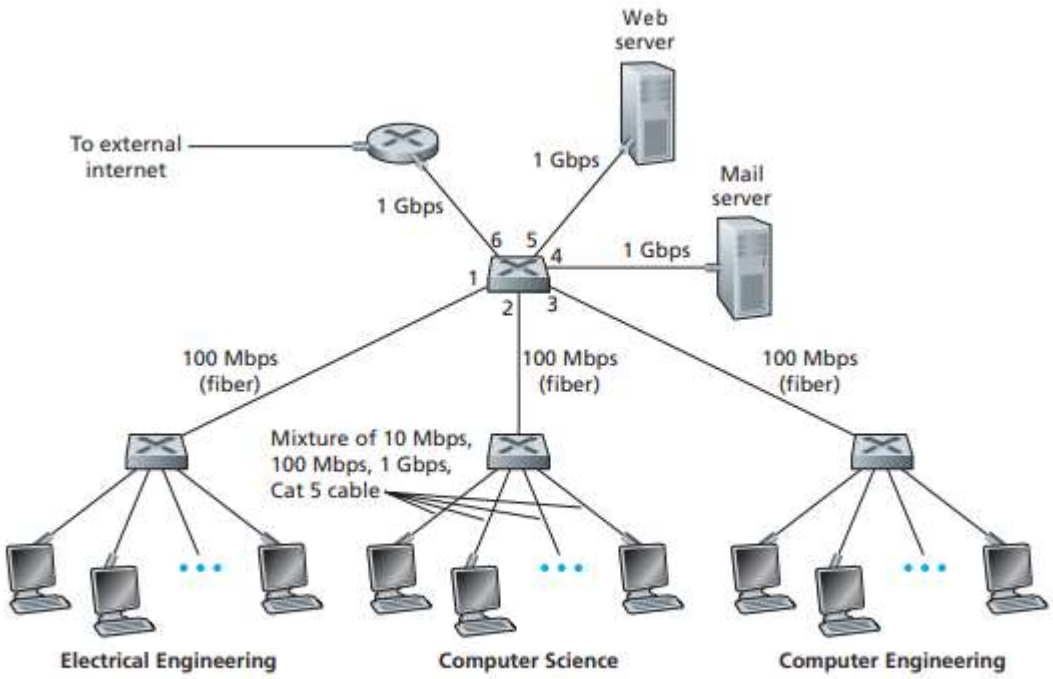

Figure 5.15 ♦ An institutional network connected together by four switches

- **MAC Addresses**:
  - **48-bit** (6-byte) unique identifier for network adapters (e.g., `AA-AA-AA-AA-AA-AA`).
  - **Flat structure**: Never changes (vs. hierarchical IP addresses).
  - **Assigned by IEEE** (OUI prefix + vendor suffix).

- **ARP (Address Resolution Protocol)**:
  - Resolves **IP → MAC address** on the **same subnet**.
  - **ARP Table**: Stores `(IP, MAC, TTL)` mappings.

- **Process**:
    1. Host A broadcasts ARP request: *"Who has IP X?"*
    2. Host X replies: *"MAC of X is Y."*
    3. Host A caches mapping.
- **Off-subnet communication**:
    - Host sends frame to **router's MAC** (not destination MAC).
    - Router forwards datagram using its own ARP for next hop.

5.4.2 Ethernet

- **Dominant wired LAN technology** (simple, cheap, scalable).
- **Frame Structure** (Fig 5.20):

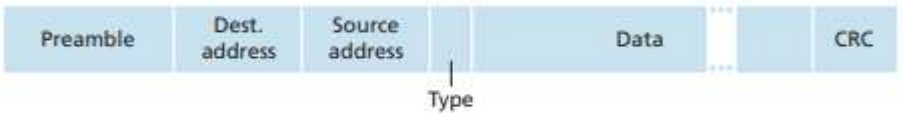

Figure 5.20 ♦ Ethernet frame structure

Field	Size	Purpose
Preamble	8 bytes	Synchronization (ends with `10101011`).
Destination MAC	6 bytes	Target adapter MAC.
Source MAC	6 bytes	Sender adapter MAC.
Type	2 bytes	Demux to network layer (e.g., IP: `0800`).
Data	46–1500 B	Payload (e.g., IP datagram).

Field	Size	Purpose
CRC	4 bytes	Error detection.

- **Services**:
  - **Connectionless**: No handshaking.
  - **Unreliable**: Discards corrupted frames (no ACKs).
- **Evolution**:
  - **Bus/hub topologies** (broadcast, collisions) → **switched star** (collision-free).
  - **Speeds**: 10 Mbps → 100 Mbps (Fast) → 1 Gbps (Gigabit) → 10 Gbps.

5.4.3 Link-Layer Switches

- **Functions**:
  - **Filtering**: Drop frames not addressed to connected segments.
  - **Forwarding**: Move frames to correct outgoing interface.
- **Self-Learning**:
  1. Initially empty **MAC table** `(MAC, interface, TTL)`.
  2. **Learn source MAC** from incoming frames.
  3. **Age out** unused entries (e.g., 60 mins).
- **Advantages**:
  - **Eliminates collisions** (full-duplex, buffered interfaces).
  - **Heterogeneous links** (mixed speeds/media).
  - **Plug-and-play** (no config needed).

Switch vs. Router:

Switch (L2)	Router (L3)
Forwards by **MAC**	Forwards by **IP**

Switch (L2)	Router (L3)
Plug-and-play	Requires IP config
Susceptible to broadcast storms	Blocks broadcasts
Spanning tree limits topology	Flexible routing

5.4.4 VLANs (Virtual LANs)

- **Problem**: Traditional LANs lack:
  - Traffic isolation (broadcasts flood entire network).
  - Efficient switch use (many small groups).
  - Flexible user management (physical rewiring).
- **Solution**: **VLANs** partition single switch into multiple **logical LANs**.
    - **Port-based VLAN**: Assign ports to VLANs (e.g., ports 1–8: Engineering).
    - **Inter-VLAN communication**: Requires **router** (or layer-3 switch).

- **VLAN Trunking** (Fig 5.26):

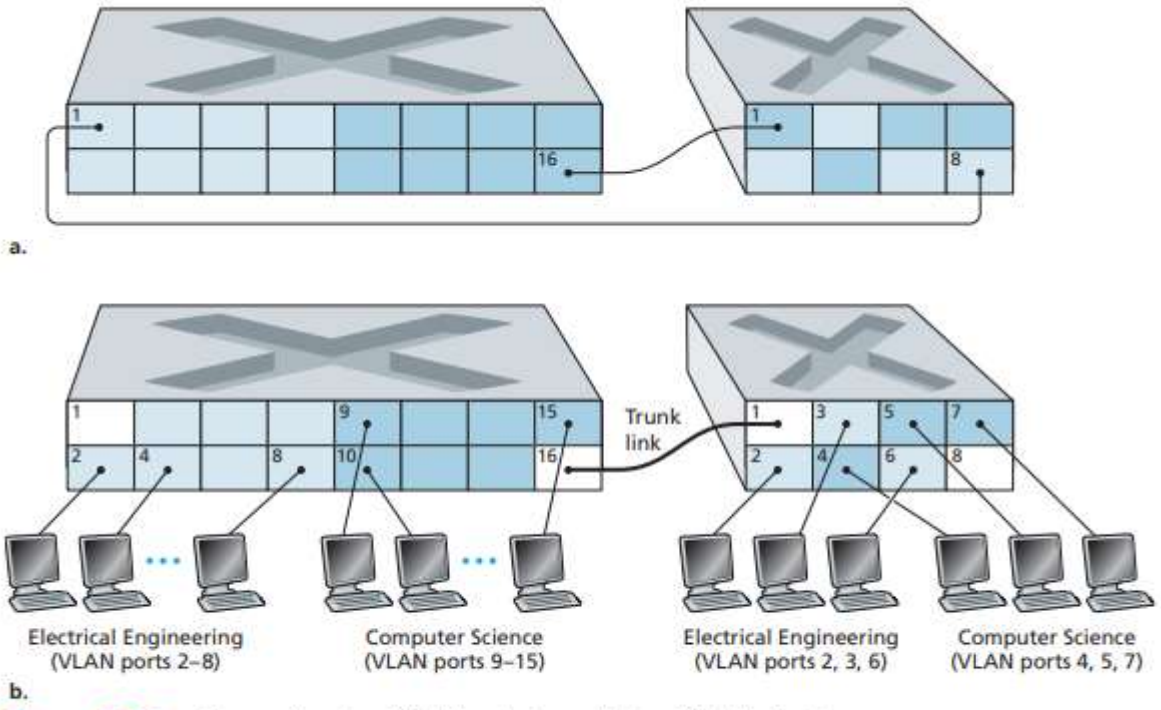

Figure 5.26 ♦ Connecting two VLAN switches with two VLANs: (a) two cables (b) trunked

- **Problem**: Scaling inter-switch VLAN links.
- **Solution**: **802.1Q tag** (4 bytes) added to frames:
  - Includes **VLAN ID** (12 bits).
  - Carried over **trunk port** (belongs to all VLANs).

Key Takeaways

- **MAC & ARP** enable local delivery; **routers** handle inter-subnet traffic.
- **Ethernet** dominates via simplicity, evolution (switches > hubs).
- **Switches** > Hubs: Intelligence (filtering/forwarding), collision elimination.
- **VLANs** solve scalability/management issues in large LANs.

 Real-World: Modern LANs combine Ethernet switches, VLANs, and routers for scalable, secure networks.

Link Virtualization: A Network as a Link Layer

Link Virtualization & MPLS: Student Notes

5.5 Link Virtualization: Network as Link Layer

- **Concept**: Treat complex networks (e.g., telephone, MPLS) as a single "link" for IP layer.
  - *Example*: Dial-up modem → Telephone network = virtual "wire" for IP.
- **MPLS (Multiprotocol Label Switching)**:
  - Blends **virtual-circuit (VC)** techniques with IP routing.
  - Enhances speed/traffic management without replacing IP.

5.5.1 MPLS Key Mechanics

1. **Label Switching**:

Figure 5.28 ♦ MPLS header: Located between link- and network-layer headers

- Adds a 32-bit **MPLS header** between L2 (e.g., Ethernet) and L3 (IP) headers.
- **Label** (20 bits): Acts as VC identifier for forwarding.

- **TTL** (8 bits): Prevents looping (like IP TTL).

plaintext

```
| Eth Header | MPLS Header (Label, TTL, ...) | IP Header | Payload |
```

2. **Label-Switched Routers (LSRs)**:
   - Forward frames based on **labels**, not IP addresses.
   - Non-MPLS routers ignore/forward as standard IP packets.

3.Label Distribution:

- Routers exchange labels via signaling protocols (e.g., **RSVP-TE**).
- *Example*:
  1. R1 advertises: "Route to A via label=6."
  2. R4 learns two paths to A: label=10 (via R3) and label=8 (via R2).

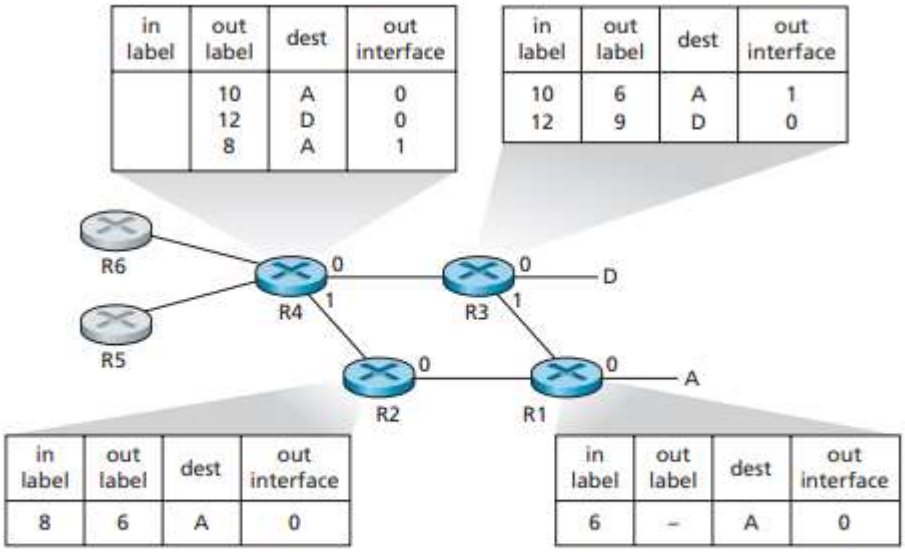

Figure 5.29 ♦ MPLS-enhanced forwarding

MPLS vs. IP Routing

Feature	Traditional IP Routing	MPLS
Forwarding	Longest prefix match (slow)	Label lookup (fast)
Path Control	Single least-cost path (OSPF)	**Traffic engineering**: Multiple paths
Flexibility	Limited by routing protocols	Override paths for QoS, redundancy

MPLS Applications

- **Traffic Engineering**:
  - Force traffic along specific paths (e.g., avoid congestion).
- **Fast Restoration**:
  - Precomputed backup paths for link failures.
- **VPNs (Virtual Private Networks)**:
  - Isolate customer traffic over shared ISP infrastructure.

Why MPLS Matters

- **Efficiency**: Faster forwarding than IP lookup.
- **Control**: Enables ISPs to manage bandwidth, prioritize traffic.
- **Reliability**: Fast failover (<50 ms).
- **Evolution**: Foundation for SDN/segment routing.

Data Center Networking

Data Center Networking: Student Notes

5.6.1 Core Components & Challenges

- **Scale**: Tens to hundreds of thousands of commodity hosts ("blades") in racks.
- **Cost Breakdown**:
    - 45% hosts, 25% power/cooling, 15% networking (switches, links), 15% power.
- **Topology**:
    - **TOR (Top of Rack) switch**: Connects 20–40 hosts per rack.
    - **Hierarchy**: TOR → Tier-2 switches → Tier-1 switches → Border routers (Fig 5.30).
- **Traffic Types**:
    - **External**: Client requests via border routers.
    - **Internet**: Host-to-host (e.g., distributed computations).

5.6.2 Key Mechanisms

- **Load Balancers ("Layer-4 switches")**:

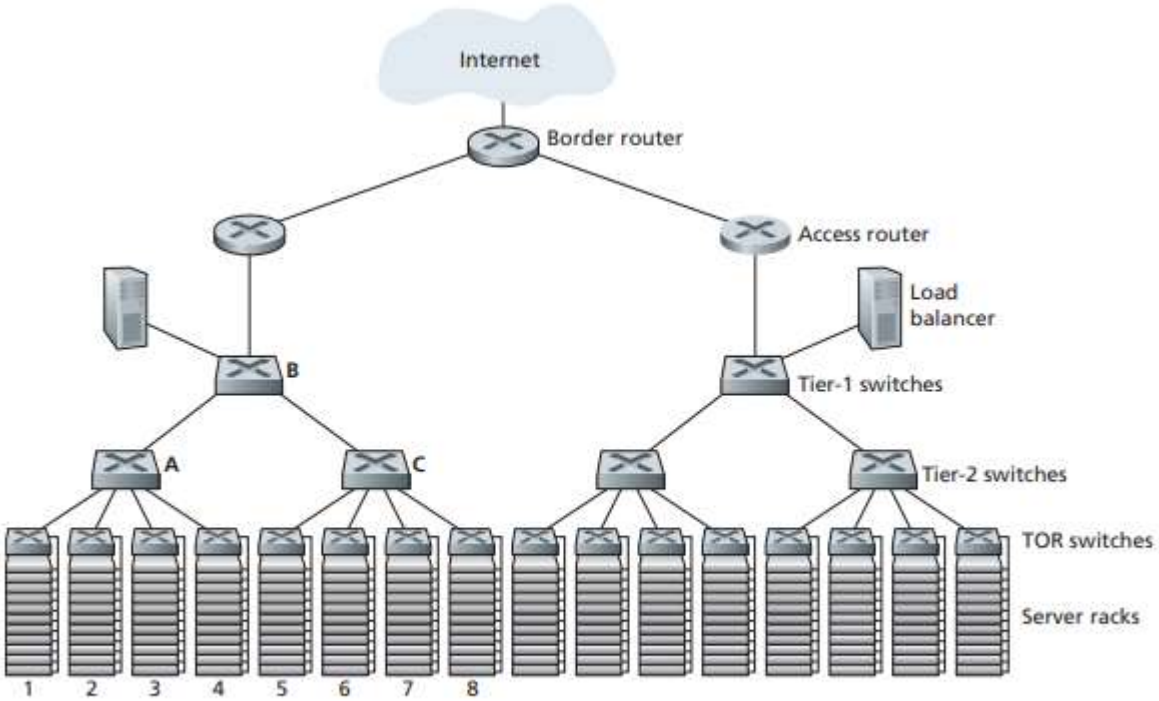

Figure 5.30 ♦ A data center network with a hierarchical topology

    - Distribute external requests to internal hosts.

- Perform **NAT**: Hide internal IPs for security.

- **Hierarchical Limits**:

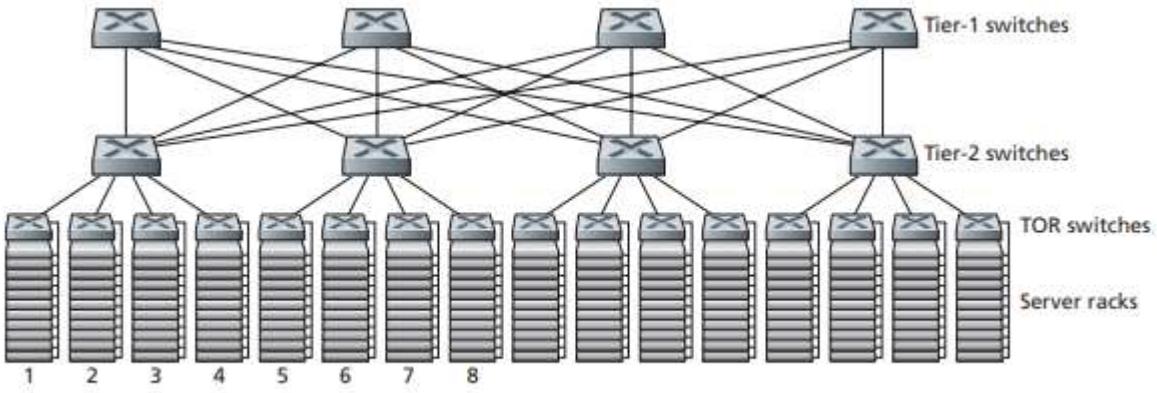

Figure 5.31 ♦ Highly-interconnected data network topology

- **Oversubscription**: Low host-to-host bandwidth across racks.
- *Example*: 40 flows over a 10 Gbps link → each flow gets only 250 Mbps.
- **VLANs**: Segment subnets to limit ARP broadcasts.

5.6.3 Modern Trends

- **Fully Connected Topologies** (Fig 5.31):
  - Tier-1 switches connect to *all* tier-2 switches.
  - **Benefits**:
    - Higher bisection bandwidth (e.g., 40 Gbps for 40 flows).
    - Multiple disjoint paths (fault tolerance).
- **Modular Data Centers (MDCs)**:
  - Prefab shipping containers with thousands of hosts.
  - **Challenges**:
    - Internal container network (commodity Ethernet).
    - Scalable core network between containers (optical/hybrid solutions).
- **Innovations**:
  - Host-based routing (multiple NICs + smart traffic distribution).

- Software-Defined Networking (SDN) for dynamic control.

Why It Matters

- **Scalability**: Hierarchical designs struggle with east-west traffic; new topologies optimize internal bandwidth.
- **Cost Efficiency**: Commodity switches + innovative topologies reduce capex.
- **Flexibility**: VM/container placement across racks requires high, predictable bandwidth.
- **Reliability**: MDCs designed for graceful degradation; redundant paths in fat-tree topologies.

 Takeaway: Data centers evolve via *topology innovation* (fat trees, optical core) and *software-defined management* to handle scale, performance, and cost

Retrospective: A Day in the Life of a Web Page (Packet) (Textbook 2)

A Web Page Request: Protocol Journey

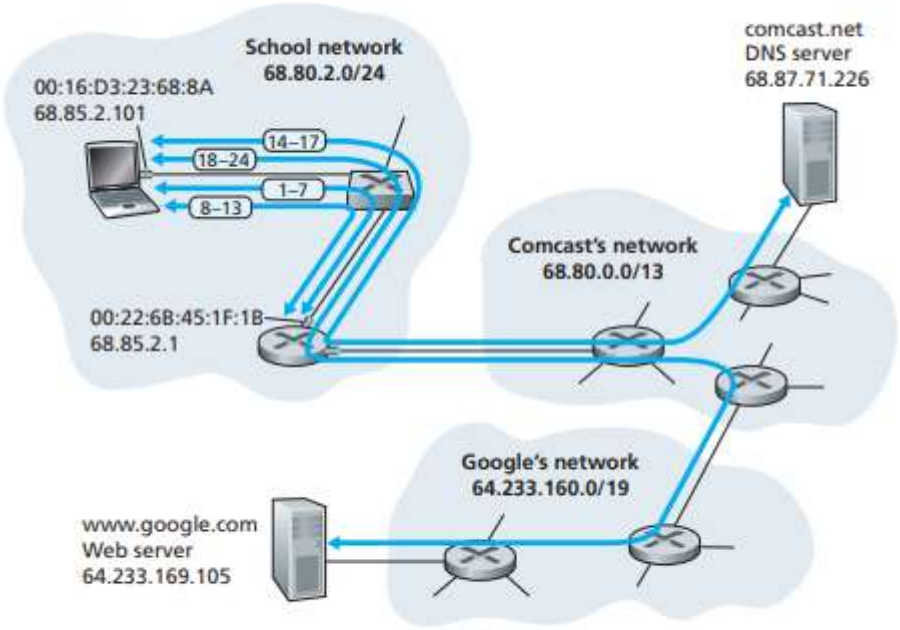

Figure 5.32 ♦ A day in the life of a Web page request: network setting and actions

1. DHCP: Getting an IP Address

- **DHCP Request** (Broadcast):
  - UDP (src:68, dst:67) → IP (src:0.0.0.0, dst:255.255.255.255) → Ethernet (src: *Bob_MAC*, dst: FF:FF:FF:FF:FF:FF).
- **DHCP ACK** (Unicast):
  - Router assigns Bob:
    - IP: `68.85.2.101`
    - DNS: `68.87.71.226`
    - Gateway: `68.85.2.1`

2. DNS: Resolving `www.google.com`

- **DNS Query**:

    - UDP (dst:53) → IP (src: `68.85.2.101`, dst: `68.87.71.226`).
- **ARP for Gateway MAC**:
    - Broadcast ARP: *"Who has 68.85.2.1?"* → Gateway replies: *"MAC: 00:22:6B:45:1F:1B"*.
- **DNS Reply**:
    - Google IP: `64.233.169.105`.

3. TCP: HTTP Connection Setup

- **TCP 3-Way Handshake**:
    1. Bob → Google: `SYN` (seq=x).
    2. Google → Bob: `SYN-ACK` (seq=y, ack=x+1).
    3. Bob → Google: `ACK` (ack=y+1).

- *Routers forward packets using **IP tables** (OSPF/BGP).*

4. HTTP: Fetching the Web Page

- **HTTP GET**:
    - Sent over TCP socket.
- **HTTP Response**:
    - Google sends HTML content.
- **Rendering**:
    - Bob's browser displays the page.

Key Protocols Involved

Layer	Protocols	Function
Application	HTTP, DNS	Web page fetch, name resolution.
Transport	TCP, UDP	Reliable data transfer (HTTP), lightweight (DNS/DHCP).
Network	IP, DHCP, ARP, OSPF/BGP	Addressing, routing, path selection.
Link	Ethernet (MAC), Switching	Local frame delivery, switching.

Takeaways

1. **Bootstrapping**:
    - **DHCP** assigns IP, gateway, DNS.
    - **ARP** resolves local MAC addresses.
2. **Name → IP**:
    - **DNS** translates domain names.
3. **Reliable Transport**:
    - **TCP** ensures error-free HTTP transfer.
4. **Routing**:
    - **OSPF/BGP** guide packets across networks.
5. **Efficiency**:
    - Link-layer switching avoids collisions; IP routing scales globally.

 Insight: This seamless process hides immense complexity across 4+ protocols—showcasing the Internet's layered design!

Unit-3

It's a pleasure to connect, Dr. Obula Raju D!

Your credentials are impressive:

- **M.Tech in Computer Networks & Information Security**
- **Assistant Professor, CSE at Andhra Engineering College**
- **Subject Code: 23A05501T**
- **Version :V03Aug2025**
- **Note: Only reference required. Please refer exclusively to** JNTUA-recommended textbooks**.**

The Network Layer Routing Algorithms, Internetworking, The Network Layer in The Internet (Textbook 1)

Routing Algorithms

1. Routing vs. Forwarding

- **Routing**: Decision-making process to determine optimal paths (handled by routing algorithms).
- **Forwarding**: Actual movement of packets using precomputed routing tables.
- *Analogy*: Routing = planning a route on a map; Forwarding = driving the route.

---

2. Desirable Properties of Routing Algorithms

- **Correctness**: Ensures packets reach the correct destination.
- **Simplicity**: Low computational overhead.
- **Robustness**: Tolerates hardware/software failures without restarting the network.

- **Stability**: Converges quickly to optimal paths after topology changes.
- **Fairness & Efficiency**: Trade-off between equal bandwidth allocation and total throughput optimization (e.g., Fig. 5-5).

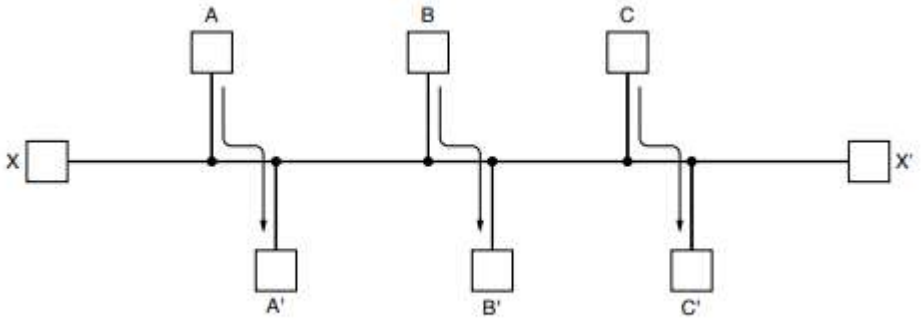

Figure 5-5. Network with a conflict between fairness and efficiency.

---

3. Algorithm Classification

A.Non-Adaptive (Static Routing)

- Routes precomputed offline; no response to failures.
- *Use Case*: Simple topologies with fixed paths (e.g., router F in Fig. 5-3 always sends to E).

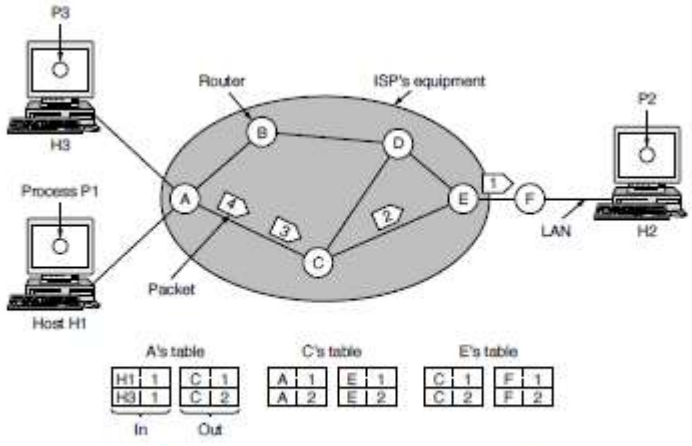

Figure 5-3. Routing within a virtual-circuit network.

B.Adaptive (Dynamic Routing)

- Routes adjust to topology/traffic changes.
- *Subtypes*:
    - **Distance Vector Routing** (e.g., RIP)
    - **Link State Routing** (e.g., OSPF, IS-IS)

---

4. Key Algorithms

A. Optimality Principle & Sink Trees

- **Principle**: If J is on the optimal path from I to K, the optimal path from J to K is the same.
- **Sink Tree**: Tree of optimal routes to a destination (loop-free, finite hops).

B. Shortest Path Algorithm (Dijkstra's)

- **Goal**: Find least-cost paths from a source to all nodes.
- **Steps**:
    1. Assign tentative distances (∞ initially).
    2. Make the node with smallest distance permanent.
    3. Update neighbors' distances.
    4. Repeat until all nodes are permanent.
- **Metrics**: Hops, delay, bandwidth, or custom combinations.

C. Flooding

- **Method**: Send packets on all outgoing links except the incoming one.
- **Optimizations**:
    - Hop counters (TTL).

- Sequence numbers to track duplicates.

- **Use Cases**: Robust broadcasting, wireless networks.

D. Distance Vector Routing (e.g., RIP)

- **Mechanism**: Routers share distance vectors (cost to destinations) with neighbors.
- **Problem**: *Count-to-Infinity* (slow convergence after failures, Fig. 5-10).

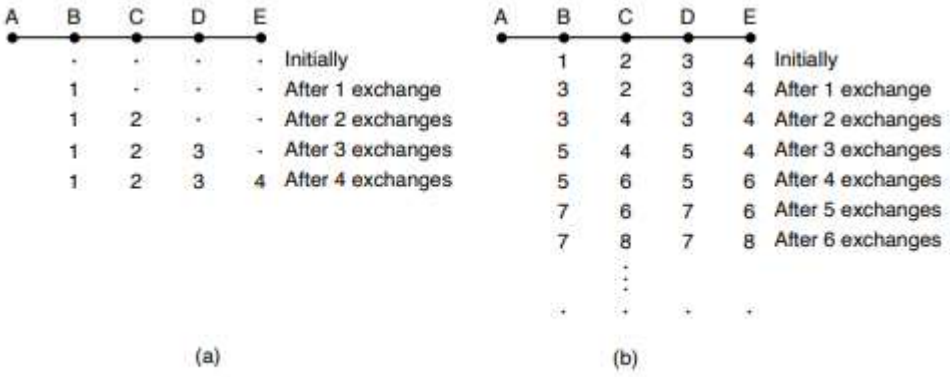

Figure 5-10. The count-to-infinity problem.

- **Fix**: Set "infinity" = longest path + 1. Split horizon/poison reverse heuristics (partial fixes).

E. Link State Routing (e.g., OSPF, IS-IS)

- **Steps**:
  1. Discover neighbors (HELLO packets).
  2. Measure link costs (delay/bandwidth).
  3. Build link-state packets (LSPs).
  4. Flood LSPs to all routers.
  5. Compute shortest paths (Dijkstra's).

- **Advantages**: Fast convergence, no count-to-infinity.

F. Hierarchical Routing

- **Goal**: Reduce table size by grouping routers into regions.

- **Example**: 720 routers → 24 regions → 53 entries/router (vs. 720 without hierarchy).
- **Optimal Levels**: ln N for N routers (minimizes entries).

---

5. Delivery Models

A. Broadcast Routing

- **Methods**:
    - Flooding (inefficient but robust).
    - Reverse Path Forwarding (RPF): Forward only if packet arrived via shortest path to source.
    - Spanning Trees: Minimal packets (requires tree knowledge).

B. Multicast Routing

- **Dense Groups (e.g., LANs)**:
    - *Prune broadcast trees* (e.g., MOSPF).
- **Sparse Groups (e.g., WANs)**:
    - *Core-Based Trees (CBT)*: Single shared tree via rendezvous point (e.g., PIM).

C. Anycast Routing

- **Goal**: Deliver to nearest group member.
- **Method**: Use standard routing (distance/link state) with shared destination address (Fig. 5-18).

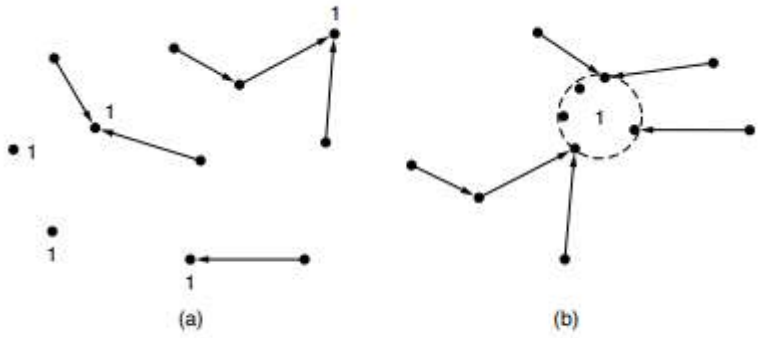

Figure 5-18. (a) Anycast routes to group 1. (b) Topology seen by the routing protocol.

Key Trade-offs

Algorithm	Pros	Cons
Distance Vector	Simple, low overhead	Slow convergence (count-to-∞)
Link State	Fast convergence, no loops	High CPU/memory for large nets
Flooding	Highly robust, no setup	Bandwidth-intensive
Core-Based Tree	Efficient for sparse groups	Suboptimal paths for some senders

---

Summary

- **Static vs. Dynamic**: Non-adaptive for simplicity; adaptive for scalability.
- **DV vs. LS**: DV for small nets; LS for large, dynamic nets.
- **Delivery Models**: Optimize based on group density (broadcast/multicast) or proximity (anycast).
- **Hierarchy**: Essential for large networks to limit table sizes.

For implementation details (e.g., Dijkstra's pseudocode, LSP formats), refer to Figs. 5-7 to 5-13 in the source material.

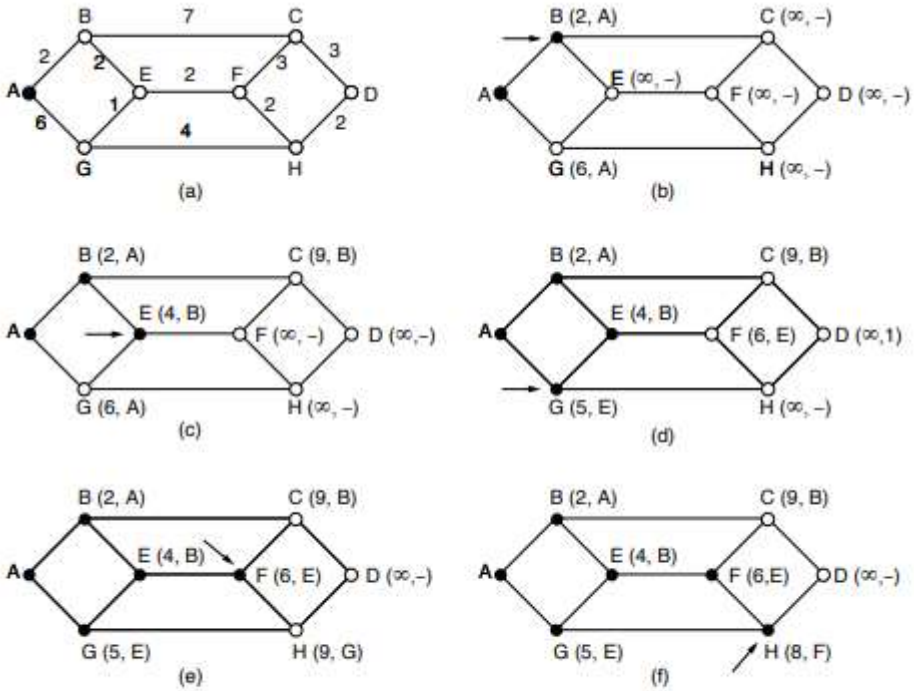

Figure 5-7. The first six steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

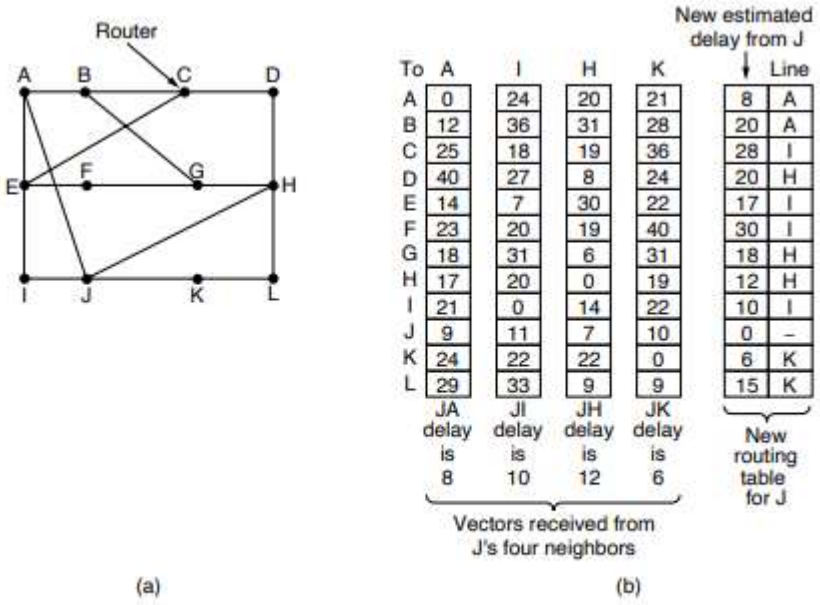

Figure 5-9. (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

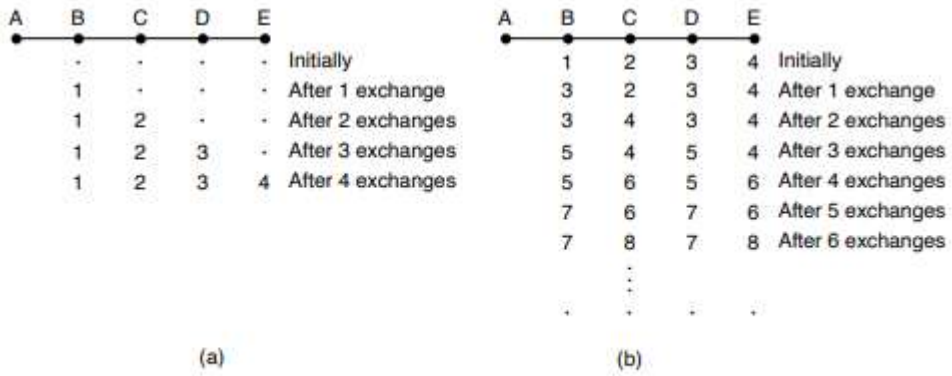

Figure 5-10. The count-to-infinity problem.

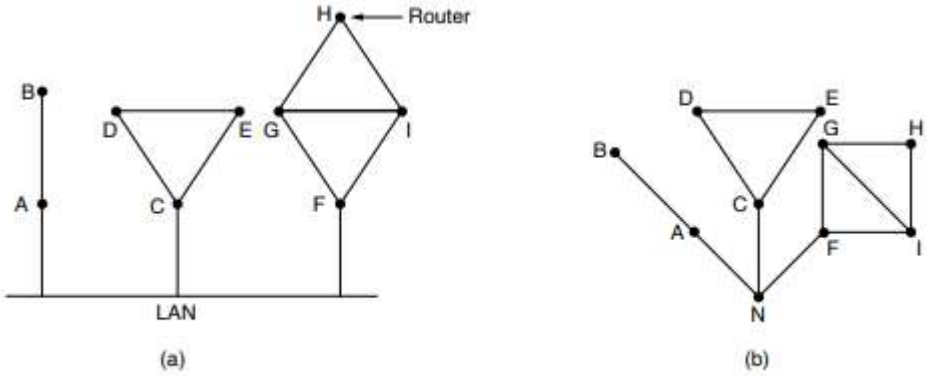

Figure 5-11. (a) Nine routers and a broadcast LAN. (b) A graph model of (a).

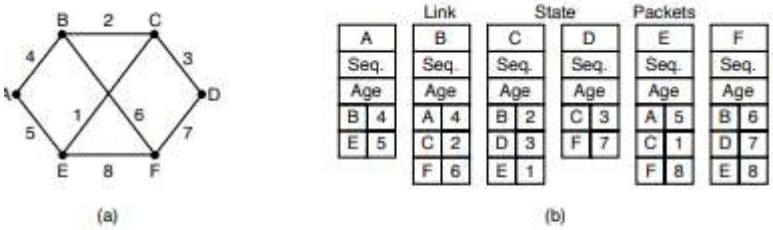

Figure 5-12. (a) A network. (b) The link state packets for this network.

Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Figure 5-13. The packet buffer for router *B* in Fig. 5-12(a).

Internetworking

5.5 Internetworking: Key Concepts

1. The Challenge:

- **Heterogeneity**: Networks differ in protocols, addressing, packet sizes, QoS, security, etc. (e.g., Ethernet vs. cellular networks).
- **Scalability**: Connecting networks exponentially increases value (Metcalfe's Law: N^2 connections for N nodes), but routing/compatibility issues arise.
- **Goal**: Enable communication across dissimilar networks (e.g., the Internet).

2. Solutions:

- **Common Network Layer (IP)**:
    - **IP** acts as a universal "lowest common denominator" protocol.

    - Routers strip incoming frame headers, route packets based on IP addresses, and re-encapsulate for outgoing networks.
    - *Limitation*: Only supports best-effort service (no QoS guarantees).
- **Tunneling**:
    - Encapsulate packets from one network protocol inside another (e.g., IPv6 inside IPv4).
    - Used when source/destination use the same protocol but transit networks differ (e.g., IPv6 over IPv4 Internet).
    - **Overlay Networks**: Created by tunneling (e.g., VPNs for security).
- **Multiprotocol Routers**:
    - Translate between protocols (e.g., IPv4 ↔ IPv6) or tunnel traffic.

3. Internetwork Routing:

- **Autonomous Systems (AS)**: Independently operated networks (e.g., ISP networks).
- **Two-Level Routing**:
    - **Intradomain Routing**: *Within* an AS (e.g., OSPF, RIP).
    - **Interdomain Routing**: *Between* ASes using **BGP (Border Gateway Protocol)**.
    - **Routing Policies**: Business agreements/laws influence path selection (e.g., cost, latency, legal boundaries).

4. Handling Packet Size Differences (Fragmentation):

- **Problem**: Networks have different **Maximum Transmission Units (MTUs)** (e.g., Ethernet: 1500B, IP: ~65KB).
- **Solutions**:
    1. **Path MTU Discovery**:
        - Hosts send packets with "Don’t Fragment" flag.

- Routers drop oversized packets + send error messages to source.
- Source retransmits with smaller fragments.
- *Modern standard*: Shifts fragmentation burden to hosts.

2. **Fragmentation at Routers**:
   - Routers split packets into fragments.
   - **Nontransparent Fragmentation (IP's method)**:
     - Fragments routed independently.
     - Reassembled only at destination using packet ID + offset + "end" flag.
   - *Disadvantages*: Overhead, inefficiency (losing one fragment drops whole packet).

Key Diagrams Simplified

- **Fig. 5-37**: IP packet traversing 802.11 → MPLS → Ethernet:
  - Encapsulation/de-encapsulation at each network boundary.
  - Fragmentation if packet > Ethernet MTU.

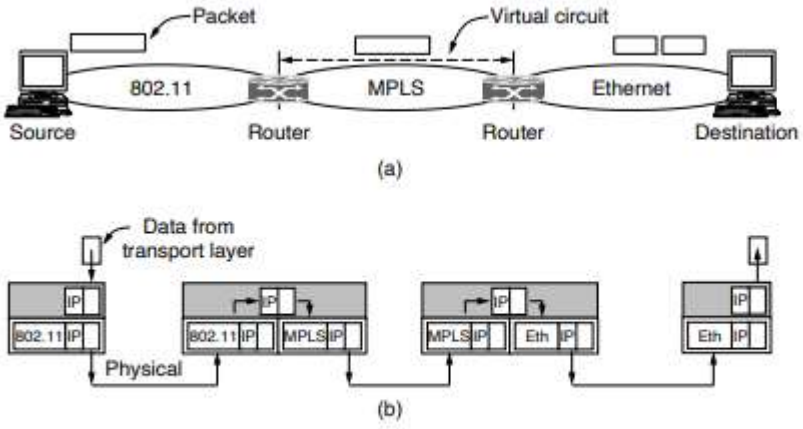

Figure 5-37. (a) A packet crossing different networks. (b) Network and link layer protocol processing.

- **Fig. 5-38 (Tunneling)**: IPv6 packet encapsulated in IPv4 header to cross IPv4 network.

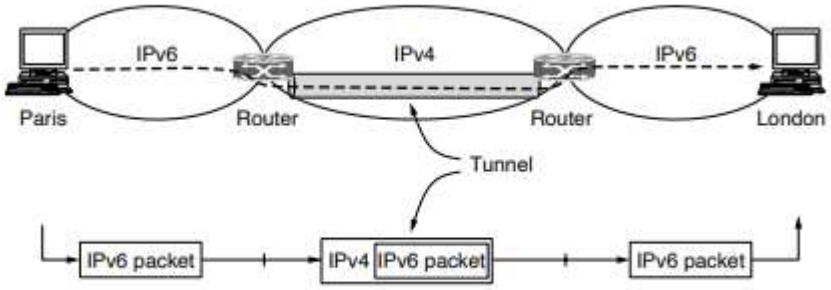

Figure 5-38. Tunneling a packet from Paris to London.

- **Fig. 5-41 (Fragmentation)**:
  - **(a)** Original packet (10B payload).
  - **(b)** Fragmented into 8B chunks.
  - **(c)** Further fragmented if next network's MTU is smaller (e.g., 5B).

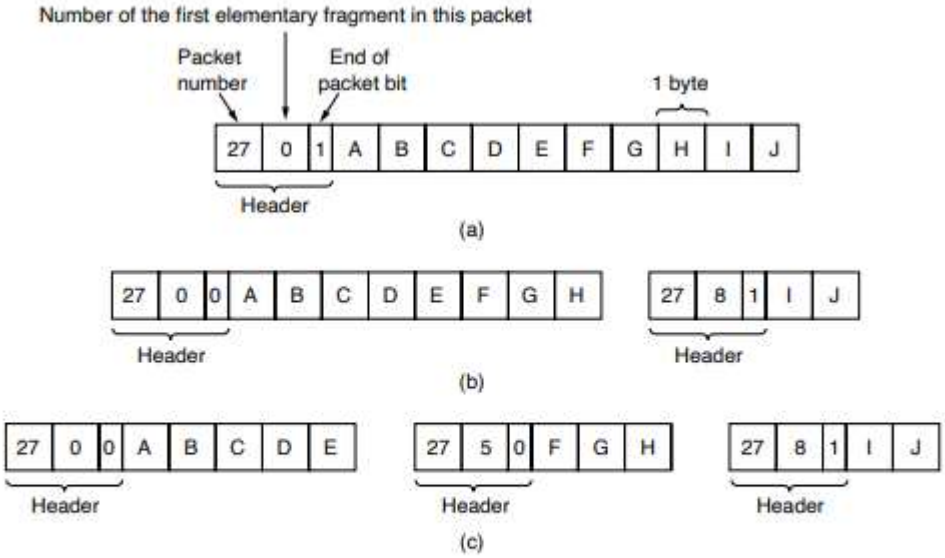

Figure 5-41. Fragmentation when the elementary data size is 1 byte. (a) Original packet, containing 10 data bytes. (b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header. (c) Fragments after passing through a size 5 gateway.

Why This Matters

- **IP's Success**: Simplicity + adaptability ("thin waist" of Internet architecture).
- **Trade-offs**:
    - *Tunneling*: Enables protocol transitions (e.g., IPv6 adoption) but limits mid-path access.
    - *Path MTU Discovery*: Avoids router overhead but adds startup delay.
    - *Fragmentation*: Necessary evil; avoided where possible.

Takeaway: Internetworking hinges on **standardization (IP)**, **tunneling**, and **adaptive routing/scaling (AS/BGP)** to bridge heterogeneity. Fragmentation is a last resort.

The Network Layer in The Internet

5.5 Internetworking: Key Concepts

1. Challenges of Heterogeneity

- **Network Differences**: Addressing, connection models (connectionless vs. connection-oriented), packet size limits (MTU), QoS, security, accounting.
- **Scalability**: Metcalfe's Law (value $\propto N2N2$ connections) drives interconnection but complicates routing.

2. Solutions

- **Common Network Layer (IP)**:
    - Acts as a universal "thin waist" between lower & upper layers.
    - Routers strip/replace frame headers at network boundaries.
    - Supports best-effort service only (no QoS guarantees).
- **Tunneling**:

- Encapsulates packets from one protocol inside another (e.g., IPv6 in IPv4 headers).
- Used when source/destination use same protocol but transit networks differ.
- Enables **overlay networks** (e.g., VPNs).

- **Multiprotocol Routers**: Translate protocols (e.g., IPv4 ↔ IPv6) or tunnel traffic.

3. Handling Packet Size Differences (Fragmentation)

- **Problem**: Networks have different **MTUs** (e.g., Ethernet: 1500B, IP: ~65KB).
- **Solutions**:
  1. **Path MTU Discovery (Modern Standard)**:
     - Hosts send packets with "Don't Fragment" flag.
     - Routers drop oversized packets + send ICMP error to source.
     - Source retransmits with smaller fragments.
  2. **Router Fragmentation (Legacy)**:
     - Routers split packets; fragments routed independently.
     - Reassembled only at destination using packet ID + offset + "end" flag.
     - *Drawbacks*: Overhead, inefficiency (losing one fragment drops whole packet).

4. Key Diagrams

- **Fig. 5-37**: IP packet traversal across dissimilar networks (e.g., 802.11 → MPLS → Ethernet) via encapsulation/de-encapsulation.
- **Fig. 5-38 (Tunneling)**: IPv6 packet encapsulated in IPv4 header to cross IPv4 network.
- **Fig. 5-41 (Fragmentation)**:
  - **(a)** Original packet (10B payload).
  - **(b)** Fragmented into 8B chunks.
  - **(c)** Further fragmented if next network's MTU is smaller (e.g., 5B).

5.7 Network Layer in the Internet

1. Design Principles (RFC 1958)

1. **Make it work** (test prototypes first).
2. **Keep it simple** (avoid non-essential features).
3. **Make clear choices** (one solution per problem).
4. **Exploit modularity** (independent layers).
5. **Expect heterogeneity** (diverse hardware/applications).
6. **Avoid static parameters** (negotiate dynamically).
7. **Prioritize good over perfect design**.
8. **Be strict when sending, tolerant when receiving**.
9. **Ensure scalability** (no centralized databases).
10. **Optimize performance/cost**.

2. IPv4 Protocol

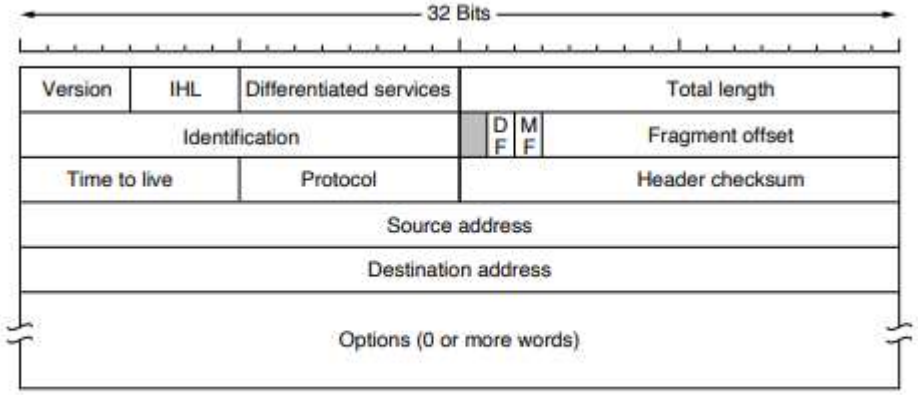

Figure 5-47. The IPv4 (Internet Protocol version 4) header.

- **Header Fields**:
  - **Version** (4), **IHL** (header length), **Total Length** (max 65,535B).
  - **TTL** (prevents looping), **Protocol** (e.g., TCP=6, UDP=17).
  - **Source/Dest Addresses** (32-bit).
  - **Checksum** (header integrity).

- **Fragmentation**: Handled via **Identification**, **Flags** (DF, MF), **Fragment Offset**.
- **Options** (rarely used): Security, strict/loose source routing, record route.

3. IP Addressing

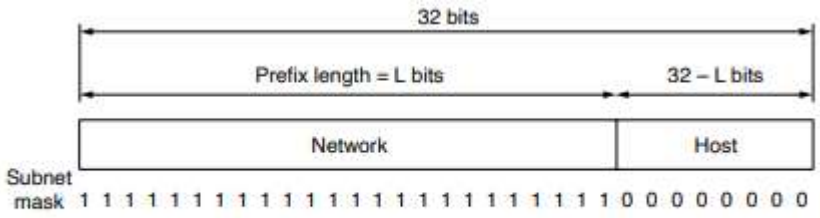

Figure 5-49. An IP prefix and a subnet mask.

- **Hierarchical Structure**: Network prefix + host portion.
- **Subnetting**: Splits a network into smaller subnets (e.g., `/24` → `/26` subnets).
- **CIDR (Classless Inter-Domain Routing)**:
  - Replaces classful addressing (A/B/C).
  - Aggregates routes (e.g., merging `/24` prefixes into `/22` supernet).
  - Reduces global routing tables (~300k entries).

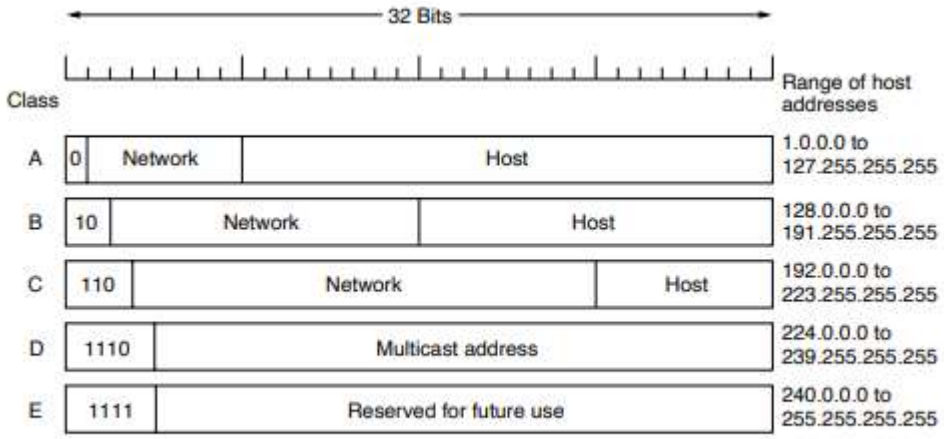

Figure 5-54. IP address formats.

- **NAT (Network Address Translation)**:
  - Maps private IPs (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) to public IP.
  - Uses TCP/UDP ports to track connections.

- *Drawbacks*: Breaks end-to-end connectivity, complicates protocols (e.g., FTP).

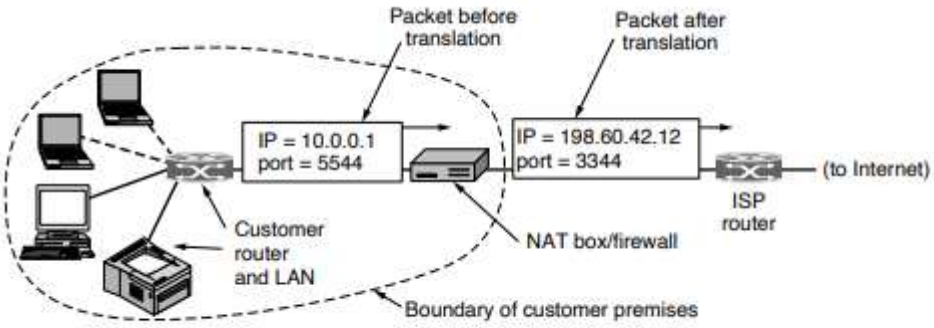

Figure 5-56. Placement and operation of a NAT box.

4. IPv6 Protocol

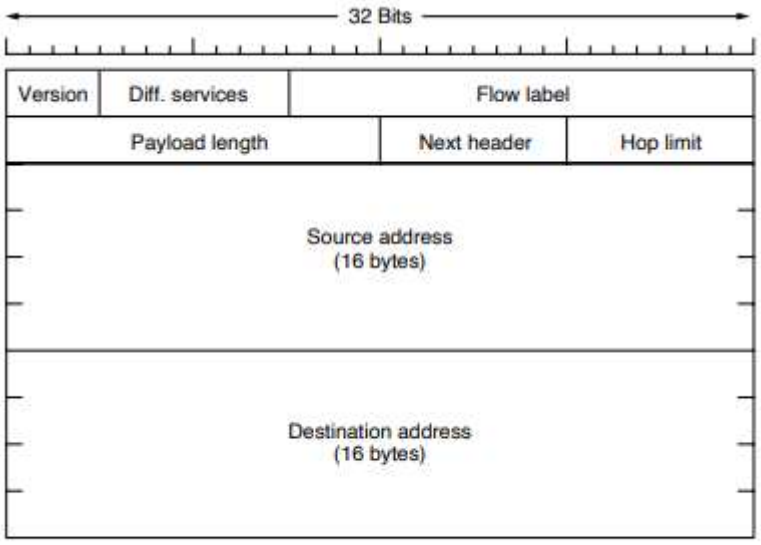

Figure 5-57. The IPv6 fixed header (required).

- **Addresses**: 128-bit (e.g., `2001:0db8::1`), supports auto-configuration.
- **Header Simplification**: Fixed 40B header, no checksum, no fragmentation (uses Path MTU Discovery).
- **Key Features**:
  - **Flow Label** (QoS support).
  - **Extension Headers** (options: hop-by-hop, routing, fragmentation, authentication).

- **Deployment Challenge**: Slow adoption despite IPv4 address exhaustion.

5. Control Protocols

- **ICMP**: Reports errors (e.g., Destination Unreachable, Time Exceeded), used in `ping`/`traceroute`.
- **ARP**: Maps IP → MAC address on LANs via broadcast queries.
- **DHCP**: Dynamically assigns IP addresses, subnet masks, gateways.

6. Routing Protocols

- **OSPF (Intradomain)**:
  - Link-state protocol, divides AS into **areas** (backbone = area 0).
  - Routers flood **LSA** (Link State Advertisements), compute shortest paths.
- **BGP (Interdomain)**:
  - Path-vector protocol; routes include **AS-path** (e.g., AS3 → AS2 → AS1).
  - Policies based on **business relationships**:
    - **Transit**: Customer pays ISP for Internet access.
    - **Peering**: Settlement-free exchange between ISPs.
  - **Hot-Potato Routing**: Exit traffic via closest router.

7. Advanced Technologies

- **MPLS ("Layer 2.5")**:
  - Uses **labels** for fast forwarding (not IP addresses).
  - Supports **traffic engineering** and QoS.

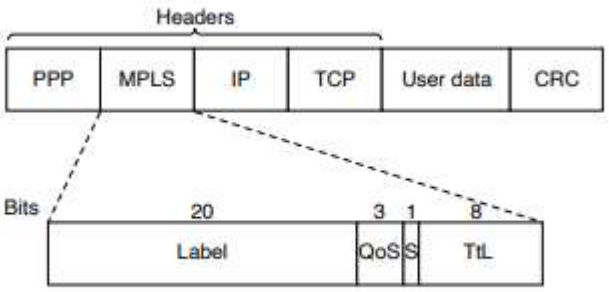

Figure 5-63. Transmitting a TCP segment using IP, MPLS, and PPP.

- **Multicast**:
    - **IGMP**: Manages group membership on LANs.
    - **PIM**: Routing protocol (Dense/Sparse Mode) for multicast trees.

Key Takeaways

- **IP's Success**: Simplicity, hierarchical addressing (CIDR), and adaptability.
- **IPv4 → IPv6**: Driven by address exhaustion; IPv6 simplifies headers but faces slow adoption.
- **Routing**:
    - **Intradomain (OSPF)**: Fast convergence, hierarchical areas.
    - **Interdomain (BGP)**: Policy-driven (economics > shortest path).
- **Real-World Constraints**: NAT breaks end-to-end model but is a necessary workaround; BGP policies reflect business deals.