T-401-ICYB
Introduction to Computer Networks (continued)

Stephan Schiffel
stephans@ru.is
Reykjavik University, Iceland

02.12.2025

Outline

Review

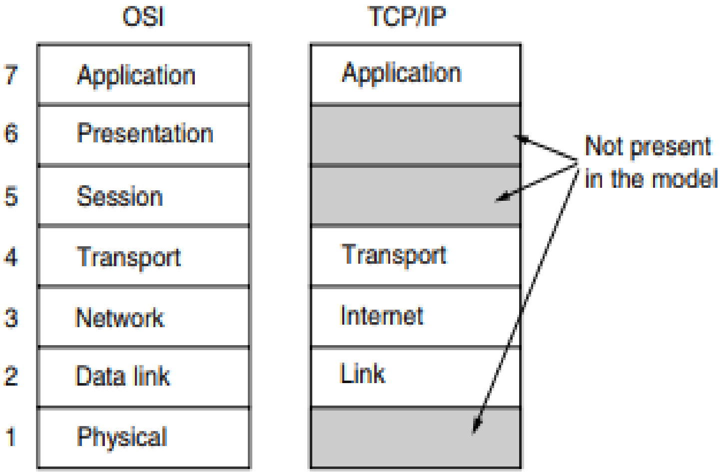


Figure 1-21. The TCP/IP reference model.

Link Layer

Node-to-Node delivery on the same network

- Protocols: Ethernet, Wi-Fi, Bluetooth, USB, NFC, RFID, ...
- Address: MAC (hardware) address

Internet / Network Layer

What guarantees can or should a Network offer?

- Guaranteed Delivery: All packets sent will eventually arrive at destination
- In order packet delivery: Packets arrive in the order they are sent
- Guaranteed Delivery within specified time
- Guaranteed Bandwidth: Sending host is guaranteed a specified bit rate (eg. 1Gbps) to the destination
- Security: No eavesdropping. No diversion to different hosts. No undetected modification.

Which guarantees are offered by the Internet (IP protocol)?

What guarantees can or should a Network offer?

- Guaranteed Delivery: All packets sent will eventually arrive at destination
- In order packet delivery: Packets arrive in the order they are sent
- Guaranteed Delivery within specified time
- Guaranteed Bandwidth: Sending host is guaranteed a specified bit rate (eg. 1Gbps) to the destination
- Security: No eavesdropping. No diversion to different hosts. No undetected modification.

Which guarantees are offered by the Internet (IP protocol)?

None of the above.

Instead it offers a *"best-effort" delivery* service.

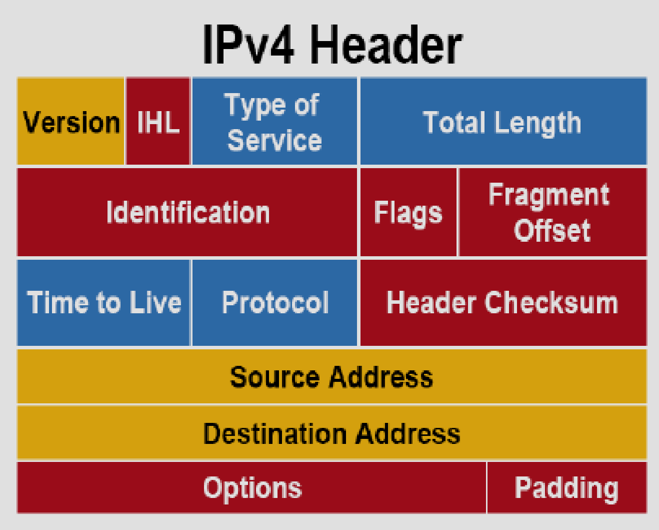
Network Layer Functions

Delivery of packets to devices anywhere in the network.

This requires

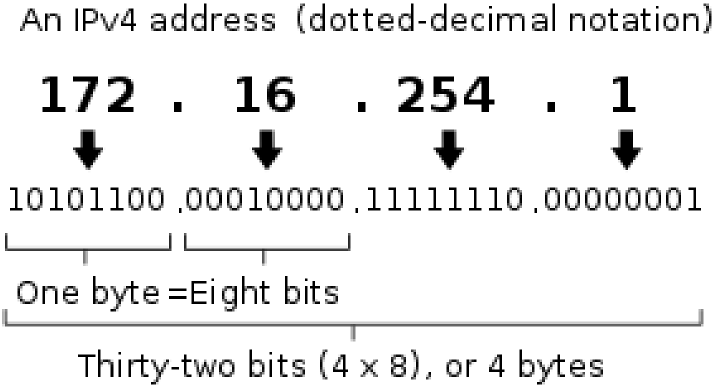
- **Addressing:** Each device is assigned a unique IP address
- **Packetization**: Divide data into manageable packets
- **Routing**: Direct packets across the network from source to destination

IP Datagram consists of



followed by the **payload** (typically a transport layer "packet")

IPv4



Notionally, high end bits are network identifier, rest is host

Subnet Addressing - Subnet mask

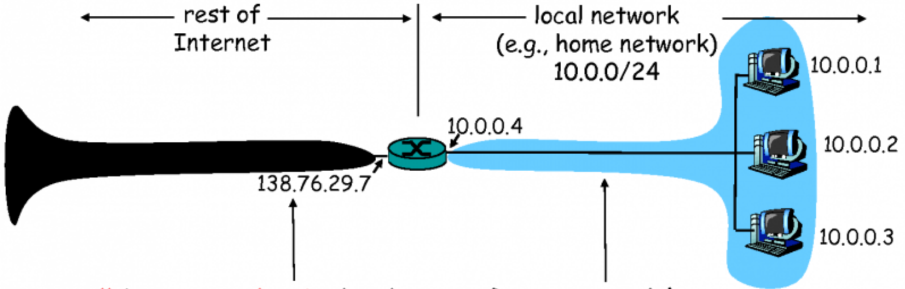
- Subnetworks are a logical division of the IP network address space
  - Also known as: Subnetting
- Written using the / to provide a shorthand reference
- eg. 198.0.1.130/24 *or* 198.0.1/24
  - 24 bits allocated to network prefix
  - *Remaining* 8 bits are the host addresses
- Subnet mask: eg. 255.255.255.0
  - Masks off network part of address to leave host's space

IPv4 Reserved Addresses

- Localhost: 127.0.0.1 (actually the entire 127/8 range)
- Local private networks: 10/8, 172.16/12, 192.168/16, ...
- Multicast: 224. - 239. (Most-significant bit pattern of 1110)
- Limited (local) broadcast: 255.255.255.255/32
- Complete list: https://en.wikipedia.org/wiki/Reserved_IP_addresses

These are not routable on the Internet.

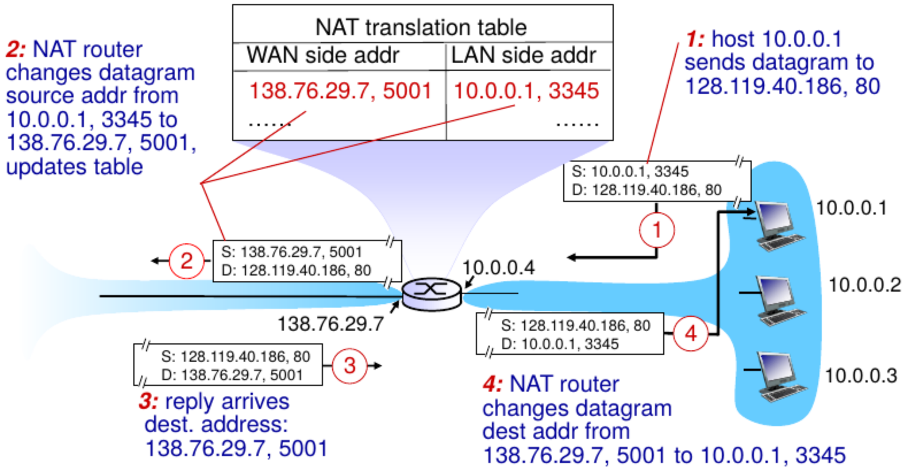
NAT: Network Address Translation



All datagrams *leaving* local network have **same** single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation



Kurose and Ross, Computer Networking: A Top-Down Approach

NAT: Usability vs. Security Impact

Pros

- **The “Natural Firewall”**: An attacker cannot initiate a connection to an internal host.
- **Topology Hiding**: The attacker sees 1 IP, not 50 endpoints.
- **IPv4 Conservation**: Connect many devices with a limited nb. of IP addresses.

Cons

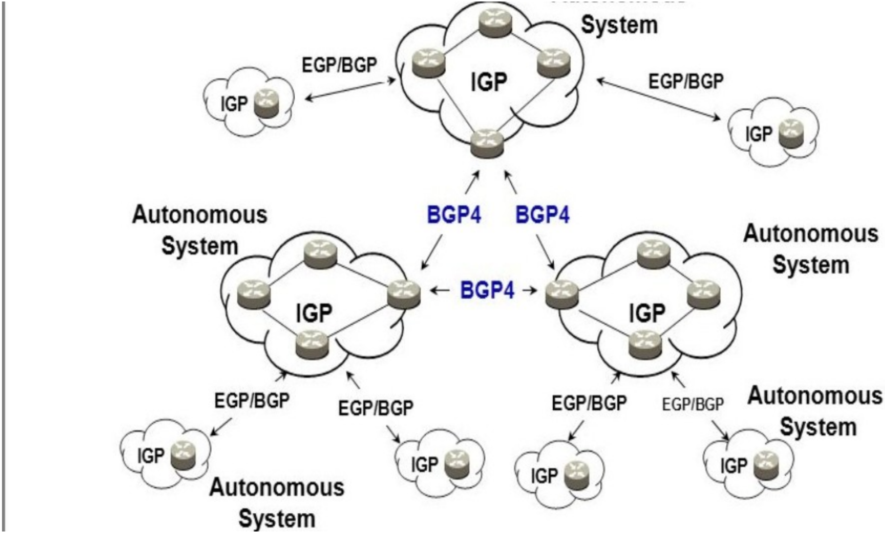
- **Breaks End-to-End Connectivity**: e.g., P2P, VoIP and Gaming. Requires workarounds (STUN/UPnP) to “punch holes.”
- **Loss of Attribution**: External logs only show the gateway IP, not which internal device launched an attack.

Warning: NAT is NOT a Security Feature

NAT stops *unsolicited* packets, but allows all *solicited* traffic.

- If a user clicks a phishing link, NAT allows the malware in. It is no substitute for an actual Packet Filtering Firewall.

Internet Routing



Routing Mechanics: Populating vs. Using the Table

How the router learns the path vs. how it forwards the packet.

1. Creating the Table

- **The Goal**: Build a map of the network.
- **Input**: Updates from neighbors or static routes.
- **Process**: Algorithms (e.g., Dijkstra) to calculate the "Best Path."
- **Result**: A forwarding table mapping IP prefixes to interfaces.
- Inspect/modify with `route` / `Get-NetRoute`

2. Using the Table

- **The Goal**: Move the packet *fast*.
- **Input**: An incoming packet's Destination IP.
- **Process**: **Longest Prefix Match**.
  - If matches for 10.0.0.0/8 and 10.1.1.0/24 exist, the /24 wins (more specific).
- **Result**: The packet is moved to the outbound interface.

Border Gateway Protocol: BGP

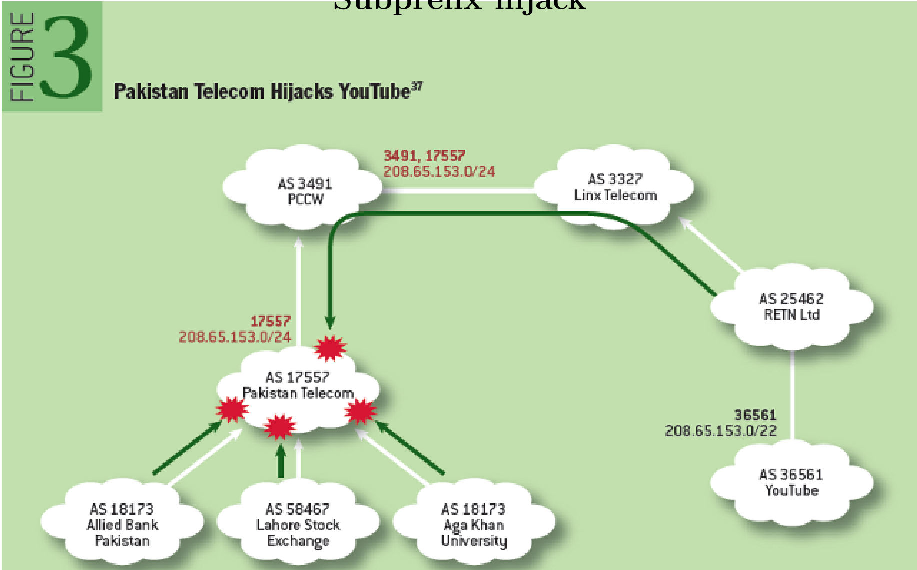
- Used between routers of neighboring ASs to exchange information about available routes.
- Routing is based on local criteria, not necessarily efficiency criteria
  - eg. Company A has a peering agreement with Company B
  - Price of having traffic carried on a particular route
  - Route length
  - Politics
- Data entry is often manual (potential for human error)
- BGP lacks basic authentication mechanisms

Possible Attack: BGP Hijacking

BGP operates on trust. An attacker can exploit the **Longest Prefix Match** rule.

- **The Attack**: The victim announces `10.0.0.0/8`. The attacker announces `10.1.0.0/16` (a sub-section of the victim's IP).
- **The Result**: Because the attacker's route is **more specific**, routers globally prefer the attacker's path. Traffic is intercepted or blackholed.

Subprefix hijack



- 2008: Youtube down globally when blocked from within Pakistan

Transport Layer

Transport Layer: TCP vs. UDP

TCP (Transmission Control Protocol)

- Connection-oriented: **3-Way Handshake** (syn, syn-ack, ack) to establish state
- Guaranteed delivery, flow + congestion control
- **Use Cases**: Web (HTTP), Email (SMTP), SSH, ...

UDP (User Datagram Protocol)

- Connectionless: Fire-and-Forget. No handshake, no confirmation.
- Fast, low overhead, no guarentees.
- **Use Cases**: Streaming, VoIP, DNS.

Possible Attacks:

- **TCP SYN Flooding**: Abuse the *state*. Initiate thousands of handshakes but never finish them. The server runs out of RAM waiting for the final ACK.

Application Layer

The Web (HTTP vs HTTPS)

HTTP (Port 80)

- Request (GET, POST, ...) - response model
- Stateless, cleartext
- **Risk**: Everything (passwords, session cookies, credit cards) is sent as readable text.
- **Vulnerability**: Anyone on the same Wi-Fi or LAN can read the traffic (Packet Sniffing).

HTTPS (Port 443)

- HTTP inside a TLS tunnel
- **Security**: Provides **Confidentiality** (Encryption), **Integrity** (Signatures) and **Identity** (Certificates).
- Prevents listening in, but also Man-in-the-Middle (MitM) attacks.

TLS (Transport Layer Security): Core Functions

provides the security layer for many application layer protocols

Confidentiality (Encryption)

- Ensures that data is unreadable to eavesdroppers (e.g., Wi-Fi sniffers).
- Implemented via **Symmetric Encryption** (e.g., AES, ChaCha20).

Integrity (Hashing)

- Ensures the data was not tampered with in transit.
- Implemented via **HMAC** (Hash-based Message Authentication Code).

Authentication (Identity)

- Ensures you are communicating with the intended server, not an imposter.
- Implemented via **X.509 Certificates** and the **Chain of Trust** (Certificate Authorities).

How TLS Works: The Handshake

The goal: safely agree on a shared secret key over an insecure wire.

- **Asymmetric (Public Key)**:
- Used *only* during setup.
- Slow, computationally expensive.
- Used for Diffie-Hellman Key Exchange.
- **Symmetric (Session Key)**:
- Used for the actual data stream.
- Fast, hardware-accelerated.
- Client and Server use a handshake to agree on algorithms used and to exchange keys.

Security Context

- Because TLS hides the payload, firewalls cannot see malware inside HTTPS traffic.
- Connection information (IP addresses, ports, amount of data exchanged) is not encrypted.

Remote Administration: Telnet vs SSH

Telnet (Port 23) - The Legacy

- Obsolete, but common in old routers/IoT.
- **Flaw**: Everything is in cleartext (including passwords).

SSH (Secure Shell - Port 22)

- The encrypted replacement for Telnet/FTP.
- Uses Public Key Cryptography for authentication and encryption.

Security Context: Brute Force

SSH used on most servers making it the #1 target for attacks.

- **Attack**: Brute forcing or randomly guessing passwords and usernames.
- **Defense**: Disable password login; use key-based authentication only.

Infrastructure: DNS & DHCP

DNS (Port 53)

- Translates Names to IPs.
- **Issue**: UDP/cleartext by default, no authentication.
- Various Attacks: Cache Poisoning, DNS Tunneling, Amplification, Typosquatting

DHCP (Ports 67/68)

- Assigns IP addresses (and other information, like DNS servers, gateway, etc) to new devices.
- **Issue**: No Authentication.
- **Attack: Rogue DHCP**. An attacker races the real server to assign a malicious gateway IP to the victim, creating a Man-in-the-Middle.

Email

SMTP (Simple Mail Transfer Protocol)

- *Pushing* mail from Sender → Server → Server.
- **The Flaw**: Designed without sender validation. By default, anyone can send mail claiming to be `admin@google.com`.

IMAP & POP3

- *Pulling* mail from Server → Client (Mail client on PC or Phone).
- **The Flaw**: Legacy versions transmit your email password in cleartext.

Security Context: Email Spoofing

Because SMTP trusts the sender, modern security relies on DNS records to patch the gap:

- **SPF/DKIM/DMARC**: DNS text records that list which IP addresses are *actually* allowed to send email for a domain.

File Sharing

FTP (File Transfer Protocol - Ports 20/21)

- **Use Case**: Uploading files to web servers or mainframes.
- **Vulnerability**: Cleartext Authentication.
- *Fix*: Always use **SFTP** (SSH File Transfer Protocol).

SMB (Server Message Block - Port 445)

- **Use Case**: Windows Network Neighborhood, Printer Sharing.
- **Vulnerability**: A complex, "chatty" protocol with a history of Remote Code Execution (RCE) bugs.

Security Context: Lateral Movement

- e.g., *EternalBlue* (used by WannaCry) exploited a flaw in SMBv1 to let malware jump from computer to computer automatically without user interaction.

Interacting with Text-based Protocols (CLI)

Many older protocols are text based (SMTP, FTP, HTTP, ...), e.g.,

SMTP (Raw Interaction)

```
$ telnet mail.server.com 25
HELO attacker
MAIL FROM: <boss@corp.com>
RCPT TO: <victim@corp.com>
DATA
Subject: Fire!
Please help.
.
QUIT
```

Note: Allows manual spoofing if no SPF/DMARC exists.

SMB

SMB is a binary protocol, but has simple text-based clients similar to FTP.

SMB (Using smbclient)

```
# 1. Enumeration (List Shares)
$ smbclient -L //10.0.0.5 -N

# 2. Connection
$ smbclient //10.0.0.5/C$ -U admin
Enter password:
smb: \> ls
  Windows                              D    0  ...
  Program Files                        D    0  ...
smb: \> get xyz
```

Security

Packet Filtering Firewalls

The Gatekeeper: Inspecting Layer 3 (IP) and Layer 4 (TCP/UDP).

- **Location**: Sits at the network boundary (Router/Gateway).
- **Logic**: Compares packet headers against an **Access Control List (ACL)**.
- **Criteria**:
  - Source & Destination IP.
  - Source & Destination Port.
  - Protocol (TCP/UDP/ICMP).
  - Protocol Headers (e.g., TCP SYN).
- **Action**: `ALLOW` or `DROP`.

Example ACL Rule Set

Src	**Port**	**Dest**	**Action**
Any	443	Web Srv	ALLOW
Admin	22	Web Srv	ALLOW
Any	Any	Any	DENY

Default Deny: The final rule (Implicit Deny) ensures that anything not explicitly allowed is blocked.

Virtual Private Networks (VPN)

What is a VPN?

- A secure, encrypted tunnel between two points over an untrusted network.
- **Goal**: To make a remote device appear as if it is physically plugged into the local network.
- **Common Protocols**: WireGuard, IPsec, OpenVPN.

How it Works: Encapsulation

1. **Original Packet**
2. **Encryption**: The OS encrypts the *entire* original packet.
3. **Encapsulation**: The encrypted blob is wrapped inside a new IP header.
4. **Transit**: Routers only see the outer header (Dest: VPN Server).
5. **Decryption**: The VPN Server decrypts the payload, and forwards the original packet.

Note: Content of the traffic (including addresses) is hidden, but not the **volume** or **timing**.

Tools

Connectivity & Path (Layer 3)

Diagnosing reachability and routing path issues.

Function	Usage Example
ping Sends ICMP Echo Requests to check if a host is online and measure latency (RTT).	`ping google.com` `ping 192.168.1.1`
traceroute (Linux) / **tracert** (Win) Maps the path packets take to the destination by incrementing TTL. Reveals where a connection dies.	`traceroute 8.8.8.8`

CLI Tools: Interface Configuration (Layer 2/3)

Function	Usage Example
ip addr (Replaces `ifconfig`) Shows IP addresses, Subnet Masks, and MAC addresses for all interfaces.	`ip addr show`
ip route (Replaces `route`) Displays the kernel routing table and the Default Gateway.	`ip route`
ip neigh (Replaces `arp`) Displays the ARP cache (Neighbor table).	`ip neigh`

Windows **PowerShell** equivalents are `Get-NetIPAddress`, `Get-NetRoute`, and `Get-NetNeighbor`.

CLI Tools: Sockets (Layer 4) & DNS (Layer 7)

Function	**Usage Example**
ss (Replaces `netstat`) Dump socket statistics. Fast way to see listening ports.	`ss -tunlp` (TCP, UDP, Numeric, Listening, Process)
dig Detailed DNS lookup. Shows TTL, flags, and exact answer section.	`dig google.com MX` `dig @1.1.1.1 google.com`
nslookup Simple name resolution (Windows/Linux).	`nslookup google.com`

CLI Tools: Security & Advanced Debugging

Function	Usage Example
netcat (nc) Read/Write data across networks. Used for port scanning, chat, or file transfer.	`nc -v google.com 80` `nc -l 1234` (Listen)
wireshark GUI packet analyzer.	`wireshark`
tcpdump / tshark Command-line packet analyzers. Capture raw traffic for analysis.	`tcpdump -i eth0 port 80` `tshark -Y "http.request.method == POST"`
nmap Network exploration tool. Scans for open ports and OS versions.	`nmap -sV 192.168.1.1`

Up Next ..

Further Studies

- UPnP, STUN etc punch holes into NAT to allow certain incoming traffic. Why can this be problematic for security? Find some vulnerabilities of these techniques. How can they be countered?
- SSL Inspection: In a corporate environment, some firewalls can inspect HTTPS traffic to look for malware (or leaked data). How is this possible given that TLS provides end-to-end encryption?
- BGP & The Chain of Trust: Discuss how HTTPS (TLS) mitigates some of the problems of BGP hijacks. (Example: An attacker successfully hijacks a BGP prefix for a bank and diverts all traffic for the bank's web server to a machine controlled by the attacker. Can they decrypt the traffic? Would the user notice? How?)

Lab today

- Lab 7: Network scanning