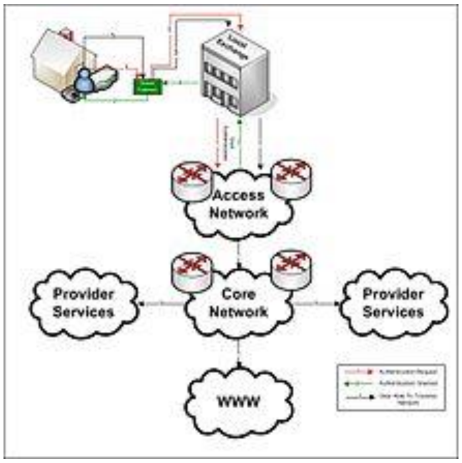
Lecture 1: Overview

Networks – Course summary – Ahmed Nouralla

- **Computer Network:**
  - Collection of computing devices that are connected in various ways to communicate and share resources.
- **Internet**
  - **Billions of connected computing devices**
    - These devices are called **hosts (end systems)**
    - They are connected - with the help of **networking hardware** - through **communication links** that can be wired (copper/fiber) or wireless (radio/satellite).
    - **Networking hardware** (router, switch, bridge, hub, repeater, NIC, etc.) manipulates the connection between end systems, they are required for communication and interaction between network devices.
  - **Network of networks (Interconnected ISPs)**
    - Internet Service Providers (ISP): organizations that provide services including internet access, web hosting, and programming interfaces to apps.
- **Network structure**
  - **Network edge:** the applications and hosts (end systems) that use the internet service, they can be clients, or servers (e.g., in data centers).
    - **Edge (gateway) router:** the router located at the network edge that is connected to other routers in other autonomous (independent) systems.
    - **Internal router:** connects only to hosts and routers within its own autonomous system**.**
  - **Access networks:** connects hosts to their direct edge router through **physical media** (in contrast with **network core**, interconnecting local provides).
    - **Physical media:** physical communication links that can be wired or wireless, in case of wireless, the medium is the air.



  - **Network core:** mesh of interconnected routers, interconnects service providers, forming the network of networks (the internet).
    - All the network functionality happens at core and through the media.
    - Two main functions of routers:
      - **Packet Forwarding**: moving packets from router's input to appropriate router output.
      - **Packet Routing:** determines source-destination route taken by packets.
        - **Routing algorithms**: seeks out the most efficient route as circuits become available.
  - **From another perspective: network consists of nodes and links between them.**
    - Nodes refer to hosts, or networking hardware.
- **Access networks examples:**
  - **Home access network:** the home DSL router is connected to the central office DSLAM that provides telephone lines. The **DSL router** usually includes many functionalities:
    - The **routing** functionality (for wireless devices)
    - The **switching** functionality (ethernet cables to the wired devices such as desktop)
    - The **DSL modem** functionality (Line port).
    - The **Network Address Translation (NAT)** for incoming IP address.
    - **A Firewall** that monitors traffic and allows/blocks them (security).
  - **Cable network:** utilizes existing television infrastructure coaxial cables.
    - Unlike DSL, homes are connected to ISPs through Cable Modem Termination System (CMTS).
    - The Hybrid Fiber Coaxial (HFC) shared cable carries multiple signals for multiple users at different frequencies for internet/TV.
  - **Enterprise access networks**
    - Typically used in organizations, the institution hosts/servers are connected (through Ethernet switches) to the institutional router (that is directly connected to the ISP).
  - **Wireless access network**
    - Hosts are connected to routers via base stations (access points).
    - Has two types: WLAN and WWAN

LAN (Local Area Network): a network that covers a relatively small geographical area (home/organization)

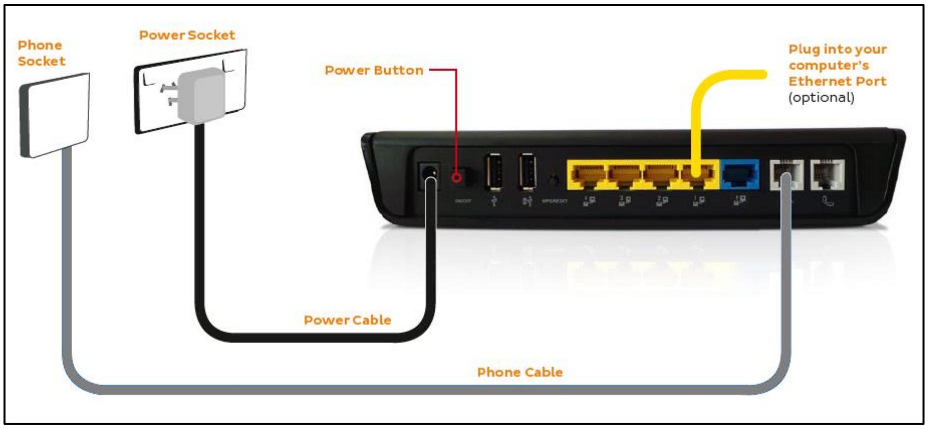
WAN (Wide Area Network): a network that covers a large geographical area (few kilometers up to countries), the internet may be considered the largest WAN.

WLAN (Wireless LAN): uses Wi-Fi protocols 802.11 to transfer data within one building (100ft),

WWAN: uses mobile cellular networks (2G, 3G, 4G LTE, 5G) to transfer data for larger distances than WLAN (10's of kilometers).

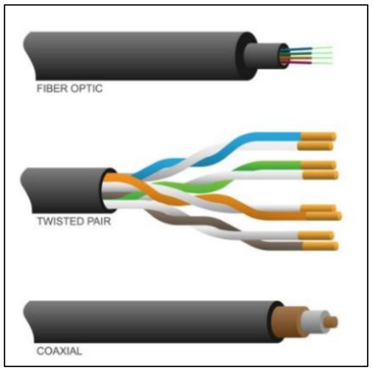
○

DSL: Digital Subscriber Line, the technology used to transmit digital data over telephone line.
Modem: is any device that is used to convert digital data into a format that is suitable for transmission.
DSL Modem: is the device that connects a computer or a router to the telephone line that provides the DSL service for internet access
DSLAM (DSL Access Multiplexor) combines all incoming links into one line and redirects the voice to the telephone net and the digital data to the internet.



Home DSL router

- **Physical (communication) links** are used to transmit data (bits) over the network through some physical medium that can be guided (wired) or unguided (wireless).
  - **Twisted pair cable:** two insulated copper wires (telephone)
  - **Coaxial cables:** concentric copper conductor surrounded by a concentric conducting shield with the two separated by a dielectric. Cable is bidirectional and broadband (TV)
  - **Fiber optic cable:** glass fiber carrying light pulses, each pulse is a bit, high-speed operation, low error rate (internet)



- **Sending data over the network**
  - The host machine breaks the data into chunks of size L bits known as packets.
  - The packets are sent over the network at some rate R (bits/sec)
  - **Packet transmission** **delay** = time needed to push L-bit packet into the link = L/R (assuming zero propagation delay)
    - **Propagation (transfer) delay** is the time needed for signals to travel across direct wires.
  - Packets (network layer) are grouped and transmitted using one of two methods: Circuit Switching & Packet Switching.

Network bandwidth consumption: the network throughput; rate (bit/s) at which the packets are sent over the network, it's the most important property of physical media.

Network bandwidth (capacity): the maximum possible network bandwidth consumption; the maximum link transmission rate or link capacity of the network, measured in bit/s.

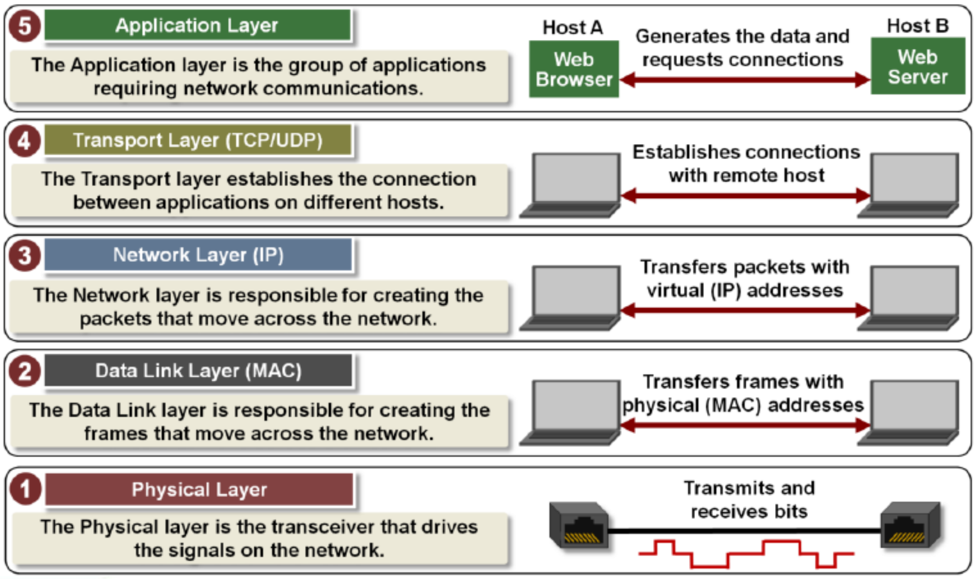
Packet Transfer: moving packet from A to B (speed depends on distance)
Packet Transmission: pushing packet bits into the link (speed depends on packet length)

Circuit Switching	Packet Switching
A physical dedicated path between source and destination is established and reserved only for the transmit **(connection-oriented communication)**	No physical path is established or reserved. Packets are forwarded from one router to the next at full link capacity until it reaches destination **(connectionless communication)**
All packets use the same path.	Packets travel independently, each packet knows the destination address and its order (for reassembling upon arrival).
Bandwidth wastage (commonly used for telephone lines)	No bandwidth wastage (used for internet).
"Store and forward" transmission is not supported (data cannot be stored in an intermediate location and forwarded later).	- Supports "Store and forward" transmission. - End-end delay = 2L/R since the packet need to be delivered entirely to be stored before forwarding. **-** If the arrival rate to a link exceeds transmission rate of the link for a period of time, packets will **queue** and can be **lost** if buffer fills up.

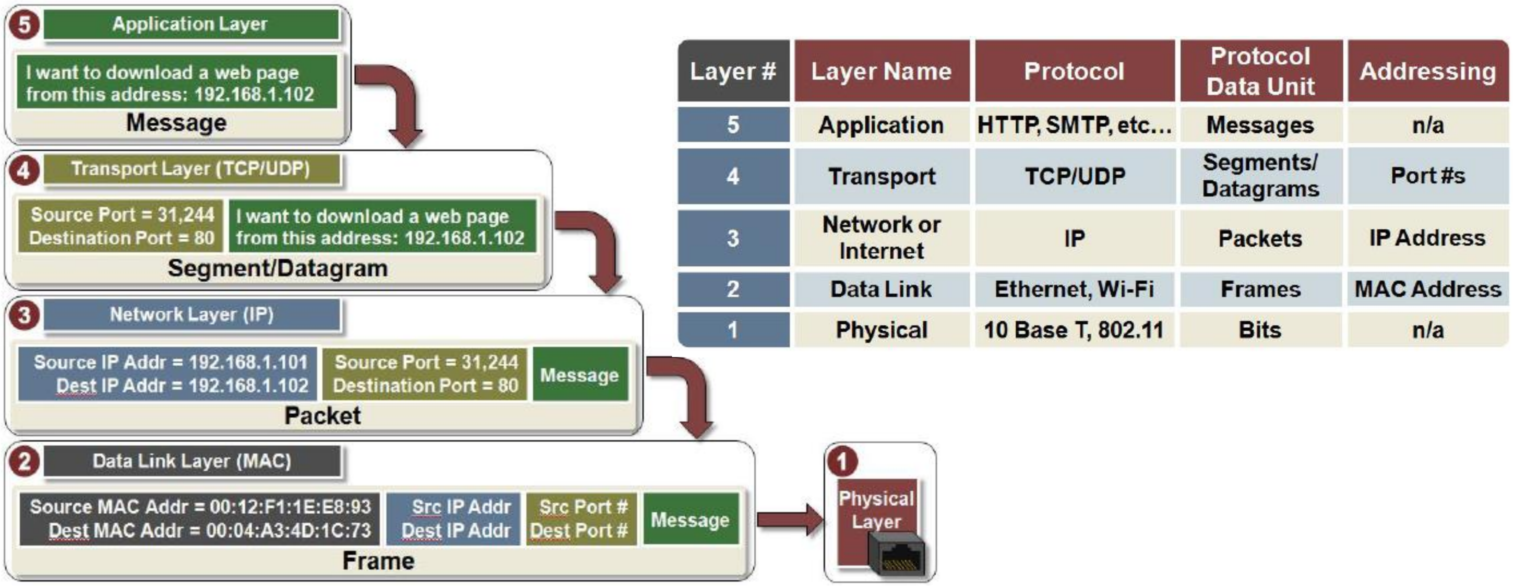
- **Internet structure:**
  - On top level: global national/international ISPs as well as content providers like Google who run their own network to bring content and services close to end-users. Top-level networks communicate through peering links using Internet Exchange Points (IXP)
  - Regional networks also arise at a lower level, they connect top level ISPs to access networks, which provides clients with the internet service.
- **Network Security:**
  - Network Security is about how hackers can attack computer networks, how to defend those attacks, and how to design architectures that are immune to attacks.
  - Internet was not originally designed with much security in mind for several reasons:
    - The purpose was just to connect mutually trusting users with each other's.
    - They were more focused on developing internet protocols.
    - Considering security in all layers was costly.
  - **Malware**
    - Malicious Software, intended to cause damage to a system.
    - **Forms:**
      - **Virus:** self-replicating infection that gets into the system by receiving and executing some object (e. g. e-mail attachment)
      - **Worm:** self-replicating infection that gets into the system by passively receiving object that gets itself executed.
      - **Spyware:** aims to gather info about the infected machine, like keystrokes, visited websites, etc. and uploads them to the hacker's collection site.
      - **Ransomware, adware, trojan horses, rogue software, wiper, and scareware.**
  - **Botnets: networks infected by a malware**
    - Each device in the network runs one or more bots.
    - Can be used for spam, stealing data, hijacking computers, or to perform **Distributed Denial-of-Service (DDoS) attacks**, in which the attackers break into the hosts around the network and send fake packets to the target, making resources unavailable to legitimate traffic.
  - **Packet Sniffing:**
    - Gathering, collecting, and logging some or all packets that pass through a computer network, regardless of how the packet is addressed.
    - Hackers can sniff packets to collect sensitive data for malicious purposes.
  - **Packet (IP) spoofing:**
    - Sending packets with false source address, usually for malicious purposes.
- **Review from Tutorials: Classification of Computer Networks (size, technology, topology, performance metrics), network delays (nodal processing, queuing, transmission, propagation).**

Lecture 2: Application Layer

- **Proprietary system**
  - A system that uses technologies and protocols kept private by a particular commercial vendor (e.g., Microsoft Windows)
- **Interoperability (compatibility)**
  - The ability of software and hardware on multiple machines and from multiple commercial vendors to communicate.
- **Open systems**
  - Systems based on a common model of network architecture and a suite of standard protocols used in its implementation (e.g., UNIX)
  - Open protocols are defined in RFC, they allow interoperability, unlike proprietary ones.
- **Network protocol**
  - A set of rules to support communication for network applications, including how to format, transmit, and receive exchanged data.
  - Protocols differ in data transmission speed, reliability, persistency, statelessness.
    - **Persistent** protocol establishes a persistent connection before sending data.
    - **Stateful** protocols: the receiver (usually a server) collect/store client session information.
- **Internet protocol suite (TCP/IP)**
  - The conceptual model and set of communication protocols used in the internet, it specifies how data should be packetized, addressed, transmitted, routed, and received.
  - This functionality is organized into five layers (According to Kurose-Ross)



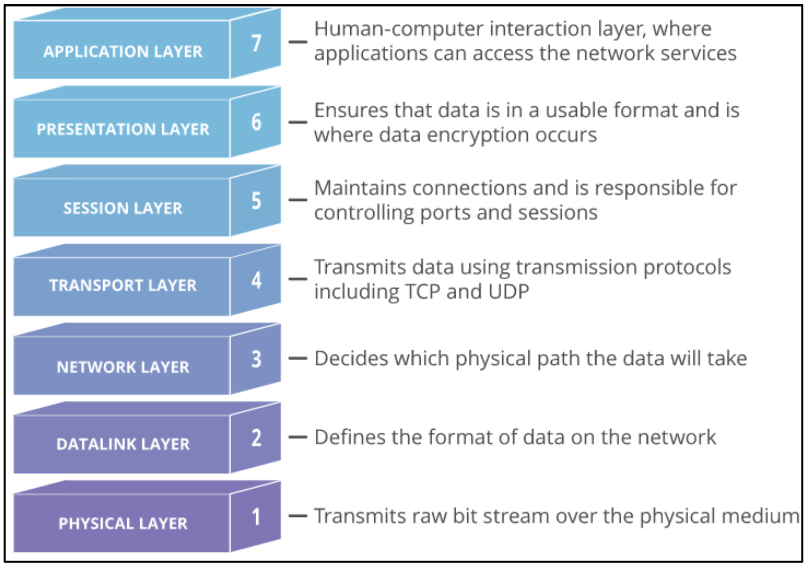
5
Application Layer
The Application layer is the group of applications requiring network communications.
Host A
Web Browser
Generates the data and requests connections
Host B
Web Server
4
Transport Layer (TCP/UDP)
The Transport layer establishes the connection between applications on different hosts.
Establishes connections with remote host
3
Network Layer (IP)
The Network layer is responsible for creating the packets that move across the network.
Transfers packets with virtual (IP) addresses
2
Data Link Layer (MAC)
The Data Link layer is responsible for creating the frames that move across the network.
Transfers frames with physical (MAC) addresses
1
Physical Layer
The Physical layer is the transceiver that drives the signals on the network.
Transmits and receives bits



5
Application Layer
I want to download a web page from this address: 192.168.1.102
Message
4
Transport Layer (TCP/UDP)
Source Port = 31,244
Destination Port = 80
I want to download a web page from this address: 192.168.1.102
Segment/Datagram
3
Network Layer (IP)
Source IP Addr = 192.168.1.101
Dest IP Addr = 192.168.1.102
Source Port = 31,244
Destination Port = 80
Message
Packet
2
Data Link Layer (MAC)
Source MAC Addr = 00:12:F1:1E:E8:93
Dest MAC Addr = 00:04:A3:4D:1C:73
Src IP Addr
Dest IP Addr
Src Port #
Dest Port #
Message
Frame
1
Physical Layer

Layer #	Layer Name	Protocol	Protocol Data Unit	Addressing
5	Application	HTTP, SMTP, etc…	Messages	n/a
4	Transport	TCP/UDP	Segments/ Datagrams	Port #s
3	Network or Internet	IP	Packets	IP Address
2	Data Link	Ethernet, Wi-Fi	Frames	MAC Address
1	Physical	10 Base T, 802.11	Bits	n/a

- **Open Systems Interconnection (OSI) Reference model** is a more comprehensive model that splits the application layer into 3 layers: presentation, session, and application.
    - **Session layer** services:
        - Authentication
        - Authorization
        - Checkpointing and recovery
    - **Presentation layer** services:
        - Data conversion
        - Character code translation
        - Compression
        - Encryption and Decryption


APPLICATION LAYER
7
Human-computer interaction layer, where applications can access the network services
PRESENTATION LAYER
6
Ensures that data is in a usable format and is where data encryption occurs
SESSION LAYER
5
Maintains connections and is responsible for controlling ports and sessions
TRANSPORT LAYER
4
Transmits data using transmission protocols including TCP and UDP
NETWORK LAYER
3
Decides which physical path the data will take
DATALINK LAYER
2
Defines the format of data on the network
PHYSICAL LAYER
1
Transmits raw bit stream over the physical medium

- **Application layer:**
    - **Network applications:**
        - Runs on different host systems and communicate over network.
        - Each app requires certain services from transport layer.
            - Some apps tolerate **data loss** (audio/video), while others don't (file transfer).
            - Some apps tolerate **delay** (audio), while others don't (interactive games).
            - Some apps tolerate low **throughput** ("elastic" apps), while others don't (multimedia).
            - Some apps require very **secure** connection (banking), while others don't (games).
    - **Client/Server vs Peer-to-Peer Architectures.**

	Client/Server	**Peer-to-Peer**
Idea	**Server:** host machine in data center, always on, has a static (permanent) IP address, provides services to clients. - **Common types of servers:** web, mail, FTP, database, DHCP, DNS, Proxy. - "Server" may also refer to the software service running on some host. **Client:** host machine that can connect to a server, may have a dynamic IP address. - Only clients can initiate connection to servers, **never** the other way.	P2P overcomes the problem of a centralized server that can have high traffic, low performance (bottleneck), or security risks. 1. File is divided into parts. 2. Peers (nodes) exchange missing parts of the file, until all peers get all parts.
Node roles	Nodes are divided into clients (active) and servers (passive)	Every node act as a client and a server simultaneously
Usage scale	Widely used	Less used
Complexity	Simpler (centralized algorithms)	More complex (distributed algorithms)
Efficiency	Likely to decrease with the number of clients	Likely to increase with the number of peers
Stability	A single point of failure (a centralized server)	No single point of failure (more stable)
Cost	More expensive	Cheaper (due to absence of a server)
Examples	Most of existing web applications.	BitTorrent (file sharing) Bitcoin (decentralized digital currencies) Open Garden (mobile internet connection sharing)

- **App-layer protocols** defines the type, syntax, semantics, and rules for messages exchanged by applications.

	Acronym	Protocol name	Port #	Usage	Utilizes
	HTTP	Hyper-Text Transfer Protocol	80	Retrieve and view web pages **(more details below)**	TCP (except HTTP/3)
	FTP	File Transfer Protocol	20, 21	Copies files from servers to clients. Port 20: data connection. Port 21: control connection. **(more details below)**	TCP
	TFTP	Trivial File Transfer Protocol	69	Was used for booting network devices through LAN.	UDP
	DHCP	Dynamic Host Configuration Protocol	67, 68	Automates the IP address, subnet mask, gateway host, DNS server assignment for new devices connected to a network. Port 67: client to server. Port 68: server to client.	UDP
	DNS	Domain Name System Protocol	53	Converting host names to IP addresses.	UDP (except long responses)
Mail protocol suite	**SMTP**	Simple Mail Transfer Protocol	25, 465, 587	Sending emails.	TCP (usually)
	POP	Post Office Protocol	110	Receiving emails. **Main differences:** - POP3 stores data locally, and removes it from server (accessible offline, more secure). - IMAP stores data on server (requires connection, more user friendly).	TCP
	IMAP	Internet Message Access Protocol	143, 220 993		TCP
	MIME	Multipurpose Internet Mail Exchange	-	Mail protocol extension to support attachments. (not a protocol, but a standard supplementary protocol)	-
	P2P	Peer-to-Peer protocols	-	Used in P2P networks that share resources or workload without a centralized system.	-
	SSH	Secure Shell	22	Secure login, file transfer, port forwarding.	TCP (usually)
	Telnet	Teletype Network Protocol	23	Unencrypted (not secure) text communication.	TCP
	SNMP	Simple Network Management Protocol	161	Collecting, organizing, and modifying information about managed IP networks devices.	UDP
	SIP	Session Initiation Protocol	5060	Used for internet telephony.	TCP/UDP
	RTP	Realtime Transport Protocol	5004, 5005	5004: media 5005: protocol	UDP (mostly)
	Skype	Skype Protocol	?	**Proprietary** protocol for Skype video communication service (All the protocols mentioned above are open except this one)	?

- **Mail protocol suite:**
  - **Email address format: <local_part>@<domain_name>**
    - **Local part:** can be unquoted, or quoted (less restrictive)
    - **Domain name:** hostname of email server, must comply to DNS naming standard.
  - **Mail server:** hardware machine receiving/storing **(incoming)** or distributing **(outgoing)** mail to clients.
    - **Example:** Microsoft Exchange Server, IBM Lotus Domino.
  - **Mail User Agent (MUA):** app sending/receiving mail on behalf of the user.
    - **Example:** Microsoft Outlook, Mozilla Thunderbird.
  - **Email Routing**
    - **Mail Submission Agent (MSA):** receives mail from MUA and submits it to MTA.

    - **Mail Transfer Agent (MTA):** a program running on mail server, performs checks (for destination, against spam, etc.) then redirects the mail to other servers.
    - **Mail Delivery Agent (MDA):** program running on destination mail server, redirects mail to receiver.
  - **SMTP server:** software forwarding emails, **POP(IMAP) server:** software retrieving emails.
    - SMTP is handshaking protocol, no default encryption mechanisms.
    - HTTP is not dedicated for emails but can be used with web clients.
  - **Email client:** manages user emails locally (e.g., Outlook), or remotely (e.g., Gmail).

- **File Transfer Protocol (stateful):**
  - Client (e.g., Linux FTP) instantiates two connections with the FTP server (e.g., FileZilla)
    - **Control connection**: exchanging commands and replies (authentication and commands).
    - **Data connection**: file transfer only.
  - Client sends requests through the control connection. After the
    - Connection is established.
    - Authentication is successful.
    - File exists and ready for transfer.
    - … the client instantiates the data connection. After the file is sent, the client requests to close the connection.
  - **Response code** is returned from server that indicates success/failure.
  - **FTP is extremely old, not secure by default**, but its principles are adopted in many other protocols.
  - **FTP has two modes**, passive (the client initiates both connections) and active (the client connects from a random unprivileged port N and sends PORT, then server connects back to the listening client at port N+1)

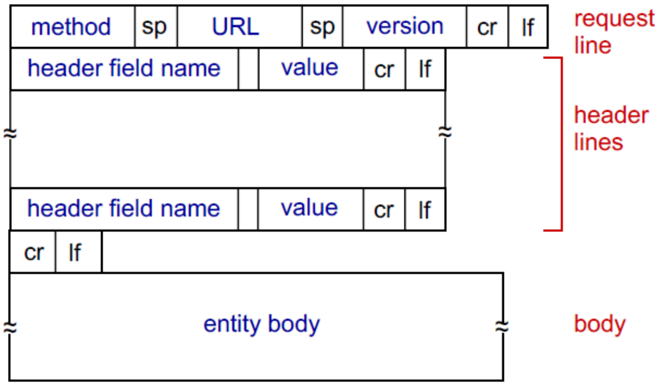
Common FTP commands	
RETR	retrieve file
DELE	delete file
DSIZ	get directory size
MKD	make directory
TYPE	set transfer mode (ASCII/Binary)
THMB	get image file thumbnail
ABOR	abort an active transfer
QUIT	disconnect from server

FTP response codes	
1xx	action started successfully
2xx	action completed successfully
3xx	command accepted; other data is requested.
4xx	temporary error
5xx	permanent error
6xx	protected reply.

- **TCP/UDP Comparison.**
  - **TCP service (connection-oriented)**
    - **Provides:** reliable data transport, flow control, congestion control
    - **Doesn’t provide:** timing, minimum throughput guarantee, security
  - **UDP service (connectionless)**
    - **Provides:** faster and more efficient, but unreliable data transport.
    - **Doesn’t provide:** flow control, congestion control**,** timing, minimum throughput guarantee, security.
- **Web and HTTP**
  - **Web pages** consist of objects (HTML files, images, audio, etc.)
    - The base HTML-file references all other objects.
    - Each object is addressable by a URL: host_name.domain/path/to/file.extention
  - **How browsing works:**
    - User asks for a specific website (types its URL in the browser address bar).
    - DNS protocol converts the domain name to the IP address of the website web server **(more details next lecture)**
    - Client (browser) initiates a TCP connection (creates **socket**) with the web server at **port** 80.
    - Server accepts TCP connection from client (responds with acknowledgement).
    - Client sends an HTTP **request** to the web server, requesting a web page, the server sends back a **response** with the requested data.
      - The first request requests the base HTML file, after which the referenced files are requested.
      - **HTTP is stateless** (i.e., information about previous sessions is not stored).
      - **HTTP versions:** HTTP/0.9, HTTP/1.0, HTTP/1.1, HTTP/2.0, HTTP/3.0
    - Client and server keep exchanging HTTP messages through the TCP connection.
    - TCP connection is closed.

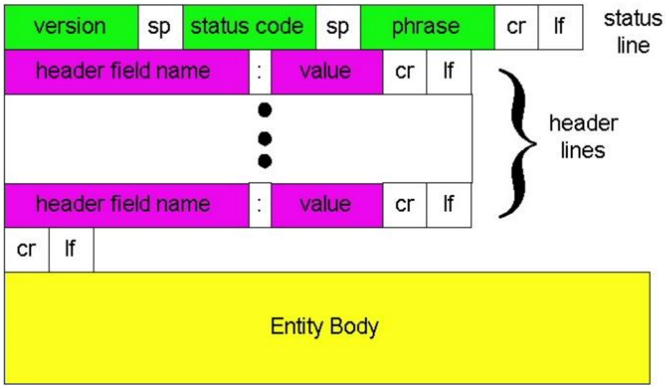
Non-Persistent HTTP	Persistent HTTP
At most one object is sent over TCP connection, after which the connection is closed.	Multiple objects can be sent (bidirectionally) over a single TCP connection.
HTTP response time = 2 RTT(*) **per object** (**), one for initiating the connection, and another for sending HTTP request and receiving the response. If a file is being received, the file transmission time is added.	HTTP response time = 2 RTT **for all objects** (**) Because the server keeps the connection open after sending response. HTTP/1.1 supports persistent connection. HTTP/2.0 supports multiple requests and responds in the same connection, with the ability to prioritize them.
(*) Round-Trip Time (RTT): time needed for a packet to go from client to server + ack. response time. (**) If files are being received, the transfer time is added to the HTTP response time.	

Common HTTP commands	
GET	Get data from server (read-only)
POST	send data to the server (input can also be quickly sent through query strings '?')
PUT	update resource (resource is the object at target URL)
DELETE	remove a resource
HEAD	requests resource header (to get metadata about the resource without downloading it).
Different types of data can be passed through HTTP, including web forms data and plain XML. ▪ Extensible Markup Language (XML): defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.	



HTTP request message format

HTTP response codes		
1xx	Informational response	100: Continue
2xx	Success	200: OK
3xx	Redirection	301: Moved permanently
4xx	Client error	400: Bad request 403: Forbidden 404: Not found
5xx	Server error	500: Internal server error 502: Bad gateway 505: HTTP version not supported.



HTTP response message format

- **Web Cookies:**
  - Small pieces of data stored on the user's computer by a browser while browsing a website.
  - Designed to make websites remember stateful information about the user, by storing the session-id (in webserver DB) at the first time the user enters a website. This id is then included in subsequent HTTP requests by the browser, which makes the server "remember" the user.
  - **Usages**
    - **Session management:** Shopping carts (online stores) - Session state (whether the user is logged in).
    - **Personalization:** Authorization (remember me checkbox) - Memorizing user preferences.
    - **Tracking:** Track user activity and use it for recommendation ads or (sometimes) malicious purposes.
- **Web Caching**
  - Used to satisfy user requests quickly without involving user origin server, which reduces server lag/traffic and reduces response time for client requests.
  - Frequently requested resources can be cached (in a proxy server for example) for quick retrieval.
  - **Web proxy server** acts as an intermediary between a client and a server, it sends HTTP requests on behalf of the client, simplifying/controlling the process and potentially masking the request origin (less used today).
    - **Proxy** is a general term for any intermediate server that communicates with other servers on behalf of a client, the concept is not bound to a specific implementation.

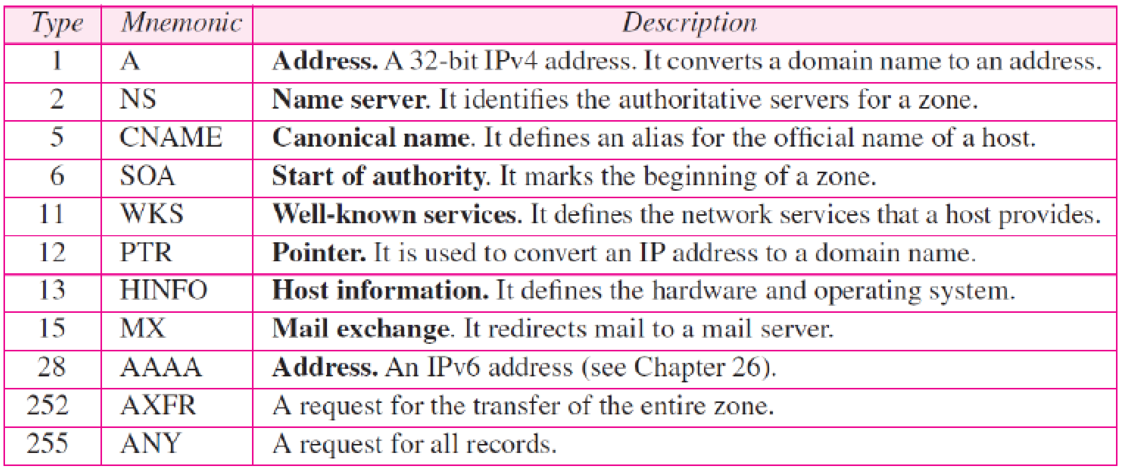
Lecture 3: DNS

- **Network Domain:** group of network hosts, identified by a domain name.
- **Domain Name System (DNS)**
    - Global naming system for computers, services, or other resources connected to the internet.
    - A globally distributed, coherent, scalable, reliable, dynamic database with a special lookup mechanism.
    - This database contains information for different network resources, stored in Resource Records (RR)
- **Top-Level Domains (TLD)**
    - Domains at the root of the DNS, the last label of a Fully Qualified Domain Name (FQDN)
    - **Examples:** .com, .net, .uk, .fr, etc.
    - **Subdomain:** www., shop.(aka hostname), etc.

DNS Database	
Global	Any device can perform lookup.
Distributed	No single computer has all DNS data.
Cache-able	Data can be cached for performance.
Loosely Coherent	Each DB subset has a serial that is incremented on update, changes are replicated, and caches expire timely.
Scalable	No limit on DB size or queries per second.
Reliable	Data is replicated from master to multiple slaves, that can be queried for performance.
Dynamic	DB is updated dynamically, modification triggers replication.

- **DNS structure:**
    - **Namespace:** set of names that are used to identify name servers.
    - **Name servers:** DNS servers that are queried for IP addresses (e.g., com DNS server).
        - **Authoritative server:** contain zone file (text file describing a DNS zone).
            - Typically, there is one primary authoritative server, and one or more secondary servers.
        - **Caching server:** stores DNS answers and reuses them.
            - A server can be authoritative and caching at the same time.
    - **Resolvers:** client side of the DNS, responsible for initiating and sequencing the appropriate queries leading to the full hostname to IP translation.
        - **Stub resolver:** the DNS resolver client library used in UNIX, acts as the interface between applications requiring DNS and the **recursive resolver**.
        - **Recursive resolver:** does the hard work of sending and receiving DNS requests to name servers.
- **DNS services:**
    - Hostname to IP address resolution (translation) **(more details next page)**
    - Host/mail server aliasing (multiple addresses leading to same IP)
    - Load distribution (multiple IPs for the same address; multiple servers hosting the same website).
- **DNS terms**
    - **Domain name:** any name in DNS format (e.g., foo.bar.example.)
    - **DNS Label:** any string between two dots, unless the dot is escaped by '\' (e.g., 'bar' in the previous example).
    - **DNS Zone:** set of names under the same authority (e.g., all websites ending with .us)
    - **Delegation:** transfer of authority from parent to child domain. (e.g., wikipedia.org. is a delegation from org.)
    - **Resource Record (RR):**
        - A field in the DNS DB (a response to a DNS query), contains data about some network resource.
        - Each RR has a type, types that share goal can further be grouped into "categories".
        - **RR "categories" or "internal types"**
            - **Authority:** NS, SOA
            - **DNSSEC** (security extensions): DS, DNSKEY, RRSIG, NSEC
            - **Meta types** (transfer info b/w DNS nodes): OPT, TSIG, TKEY, SIG(0)
            - **Indirect** (cause resolver to change search direction): CNAME, DNAME
            - **Terminal** (won't lead to further queries)
                - Address records: A, AAAA
                - Informational (carry info to apps): TXT, HINFO, KEY, SSHEP.
            - **Non-terminal** (contain domain names that may lead to further queries)
                - MX, SRV, PTR, KX, A6, NAPTR, AFSDB.
    - **RRSet:** set of all RRs of the same type (e.g., all A types).
    - **Time-To-Live (TTL):** each RR has a TTL field, contains the time after which the RRSet expire and has to be updated.

- **Common RR types**

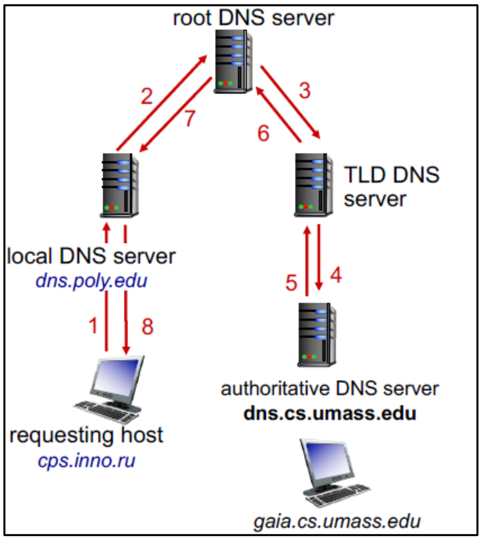


Type	*Mnemonic*	*Description*
1	A	**Address.** A 32-bit IPv4 address. It converts a domain name to an address.
2	NS	**Name server.** It identifies the authoritative servers for a zone.
5	CNAME	**Canonical name.** It defines an alias for the official name of a host.
6	SOA	**Start of authority.** It marks the beginning of a zone.
11	WKS	**Well-known services.** It defines the network services that a host provides.
12	PTR	**Pointer.** It is used to convert an IP address to a domain name.
13	HINFO	**Host information.** It defines the hardware and operating system.
15	MX	**Mail exchange.** It redirects mail to a mail server.
28	AAAA	**Address.** An IPv6 address (see Chapter 26).
252	AXFR	A request for the transfer of the entire zone.
255	ANY	A request for all records.

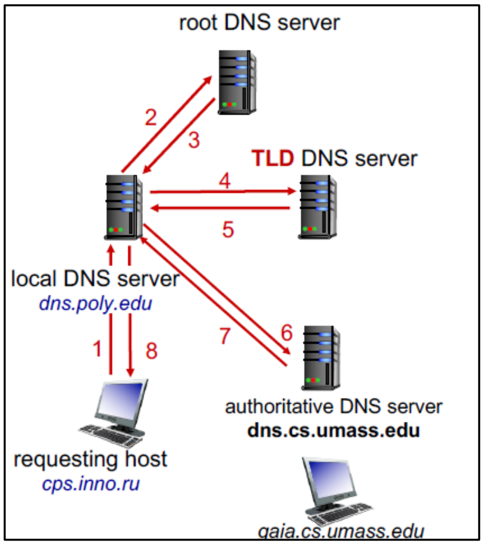
- **DNS protocol in lower layers**
  - DNS protocol uses port 53 and runs over UDP (default) or TCP (occasionally)
    - **UDP**
      - 512-byte payload that can expand by mutual agreement (EDNS0)
      - For transporting requests and responses that fit in one packet.
    - **TCP**
      - For zone transfers to maintain DB coherence.
      - When the size of request or response is greater that a single packet.
  - DNS message format, Question section (repeated in response) - RR format
    - To avoid repeating domain name, DNS uses an offset pointer that points to a previous occurrence of it (e.g., in RR, a 2-byte offset pointing to the domain name from the question record is being used).
    - Contents of fields (Name, Value) in RR format depends on the type of the query and response
      - Type = A/AAAA ⟹ Name = domain name, Value = IP address.
      - Type = NS ⟹ Name = partial domain, Value = name of DNS server for this domain.
      - Type = CNAME ⟹ Name = hostname, Value = canonical hostname (useful for aliasing)
      - Type = MX ⟹ Name = domain in email address, Value = canonical name of mail server.
- **Hostname to IP address resolution:**
  - The most important job of DNS, it doesn't only help humans to reference machines, but also allows hosting servers to change their IP addresses easily without having to change their domain name (by updating RR).
  - The story begins when the client enters some URL in the browser and presses enter).
  - URL is inspected from right to left, dot separated, last dot represents the **root name server**

How it works	
Recursive DNS Query	**Iterative DNS Query**
▪ The client checks: • Its local OS cache. If not cached, ask… • The local (default) name server (belonging to the ISP). If not cached, ask… • The root name server (13 worldwide servers). If not cached, ask… • The organization authoritative name server (which always have the answer). ▪ After the answer is retrieved, it propagates back to the client and gets cached in all levels. ▪ All in all, the client sends only one DNS query (if the address is not in OS cache) and all other servers sends queries and receive responds on behalf of the user (burden is on the server to resolve the query).	▪ The client checks: • Its local OS cache. If not cached, ask… • The local name server. If not cached, ask… • The root name server, which doesn't know and won't cache the full address, but will refer the local name server to the TLD name server to ask (on behalf of the client). • The TLD name server will refer the local name server to the organization server to ask. • The organization authoritative name server will return the IP to the local name server, which will cache it and send it back to the client. ▪ All in all, the client (or the local name server) sends multiple DNS queries to different destinations (burden is on the client).

- The answers to client queries can be **authoritative** (if it came from the organization authoritative DNS server), **or non-authoritative** (if it was cached somewhere else).
- `nslookup` is a command-line tool that is used to retrieve address/name mapping.



Recursive DNS Query

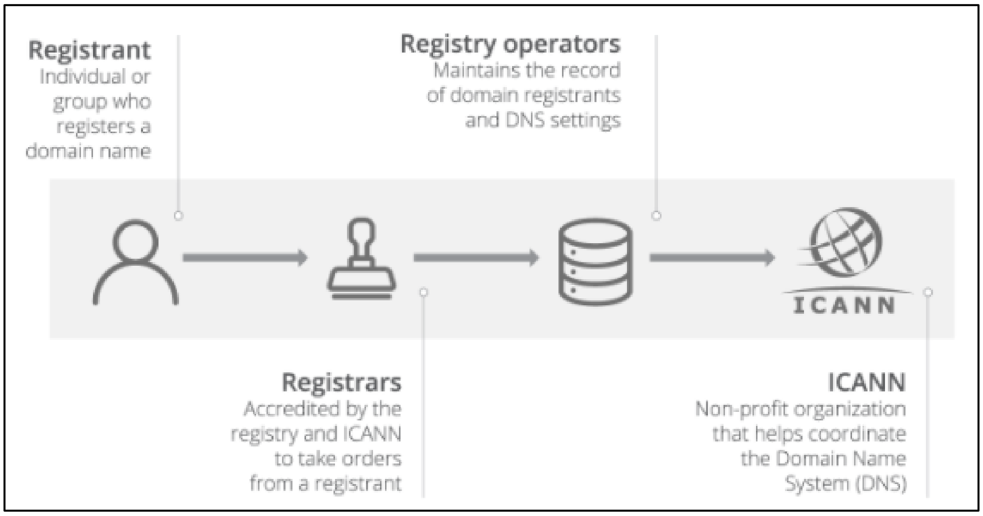


Iterative DNS Query

- **IP address to name translation (inverse lookup):**
  - A separate server hierarchy stores inverse lookup tables.
  - Record type PTR is being used (Name = IP address, Value = domain name).
  - Inverse lookup doesn't necessarily exist.

IANA: Internet Assigned Numbers Authority
ICANN: Internet Corporation for Assigned Names and Numbers

- **Inserting records into DNS**
  - A startup company X creates their own website (www.X.com)
  - X need to register it at some DNS registrar
    - **Domain name registrar** provides domain name registration for the general public.
      - They don't "own" domain names; domain name registry organizations do.
    - **Domain name registry:** organizations that manage TLDs, they are managed by the IANA (a department of the ICANN)
  - X will provide the registrar with the names and IP addresses of their authoritative name servers (primary and secondary)
  - The registrar will insert RRs (typically, A, NS) into the TLD server.
    - (X.com, dns1.X.com, NS)
    - (dns1.X.com, 212.212.212.1, A)


Registrant
Individual or group who registers a domain name
Registry operators
Maintains the record of domain registrants and DNS settings
ICANN
Registrars
Accredited by the registry and ICANN to take orders from a registrant
ICANN
Non-profit organization that helps coordinate the Domain Name System (DNS)

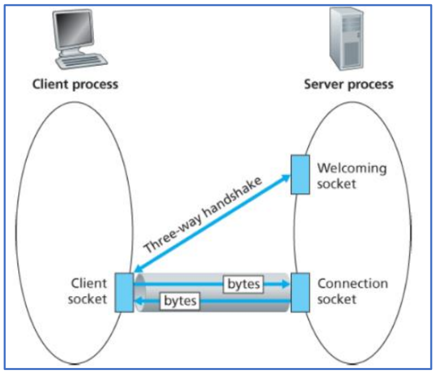
***From Wikipedia**: "Some registries sell the names directly, and others rely on separate entities to sell them. For example, names in the .com top-level domains are in some sense sold "wholesale" at a regulated price by VeriSign, and individual domain name registrars sell names "retail" to businesses and consumers."*

Lecture 4 – Transport Layer I

- **Transport Layer:** allows traffic to be directed to specific networking application **processes** through ports.
  - **Port:** a 16-bit number used to direct traffic to a specific **service** running on a networked computer.
  - **Segment:** the data unit in transport layer.
    - UDP segments can be called datagrams. "Kurose, Ross" book reserves the term for "IP datagrams".
- **Multiplexing (mux) and Demultiplexing (demux):**
  - The sender **multiplexes** (encapsulates) the message into segments, adds the transport header (with source/destination port numbers) to that segment, passes it to the network layer.
  - The receiver receives segments, use the header to demultiplex them to get the message, pass it to the application layer.
  - TCP uses connection-oriented mux/demux, while UDP uses connectionless mux/demux.
- **Socket:** the interface between processes on different hosts.
  - **Socket address:** the tuple (protocol type, IP address, port number).

Socket	Port
• Represents a single connection between two network applications. • Is used for process-to-process communication over the network, or for Inter-process communication. • When a socket is created (opened by a server) it should be **bound** to a port number before usage. • A socket **(connection)** is identified by a 4-tuple: source IP/Port, destination IP/Port	• Representing an endpoint (channel) for network communications. • Several ports on a server are used for different services. • A port is identified by a 16-bit number. • There are well-known port numbers.
Several clients want to access the same webpage. *All of them need to use the HTTP service provided by the server at **port 80.*** *Server will handle all of them, by sending the page to each client through a separate socket intended for that client.*	

- **Socket programming** allows programmers to write network applications that communicate through sockets, following the client/server architecture.
  - **Socket programming with TCP** (connection oriented, reliable, byte-stream oriented)
    - Server process is running, and is listening to incoming requests at the special "welcoming" `ServerSocket`, that is used by all clients.
    - Client contacts server by creating a TCP socket, specifying the destination IP address and port number.
    - Server accepts client connection and opens another `ConnectionSocket` that will be used by that particular client for data transfer.



  - **Socket programming with UDP** (connectionless, no handshaking required, unreliable)
    - Server process is running and listening to incoming requests.
    - Client attaches the destination IP/Port pair to each packet it sends (for routers to route the packet)
      - The OS also attaches the source IP/Port
    - Server is not guaranteed to receive the packet, packets are not guaranteed to reach in-order.

Reliable transfer over UDP

- Reliability can be added at application layer by introducing application-specific error recovery.
- UDP checksum (one's complement sum) is being used to detect errors in transmitted segments.
  - Sender calculates the checksum for the data + IP pseudo header (12 bits: src/dest IP addresses, segment length, protocol type, and reserved 8 bits) and sends it along with the segment.
  - Receiver computes the checksum of the received segment and compares it to the received checksum, error on mismatch.
- HTTP/3 is using Google QUIC protocol that utilizes UDP.

Principles of Reliable Data Transfer (RDT)

- TCP guarantees data to arrive in-order, without errors, and without duplicates.
- Error control in TCP is achieved through acknowledgement, checksum, and time-out.
- Finite State Machines can be used to represent states/events/actions that are involved in RDT process, with transitions between states labeled as “event/action” to transit from state A to B.
- Below, we develop several versions of RDT, depending on the underlying channel being used.

rdt1.0: reliable data transfer over a **reliable channel**

- Sender sends data into the underlying channel from which the receiver reads.
- No data loss will happen since the channel is reliable.

rdt3.0: over a channel with bit errors and losses

- Sender waits ACK for a reasonable amount of time before retransmission (timeout)
  - If ACK was just delayed (not lost), sequence number will handle duplicates.
  - Receiver will send a duplicate ACK for sender not to retransmit again.
- rdt3.0 is correct, but has a bad performance, sender utilization is very low because it sends one packet a time and waits for ACK.

rdt2.0: over a **channel with bit errors (channel may flip bits by mistake, but may not lose a whole packet)**

- Receiver uses positive and negative acknowledgement ACK/NAK to indicate if the received packet is ok, or had errors **(error detection, feedback)**, sender retransmits on NAK **(error recovery).**
- rdt2.0 flaw: ACK/NAK signal can itself get corrupted.

rdt2.1: a modification that avoids rdt2.0 flaw.

- If ACK/NAK gets corrupted, retransmit along with same **sequence number (0 or 1)**
  - Sequence number normally alternates, on retransmission, the same one is sent.
- To avoid duplicates, receiver will discard packets with the same sequence number.
- Requires twice as many states as rdt2.0, since each state must remember the expected sequence number.

rdt2.2: a NAK-free protocol

- Provides the same functionality as rdt2.1, without using NAKs.
- On receiving packets, receiver sends ACK with the sequence number of the last successfully received packet (either this one if it’s received successfully, or the previous one otherwise).
  - Two consecutive ACK with the same sequence number at sender = NAK, sender retransmits.

- **Automatic Repeat Query (ARQ)** is any error control method that uses ACKs and timeouts to achieve reliable data transmission over an unreliable channel. ARQ types include:
  - **Stop-and-wait** ARQ (used in rdt2.0+ we developed above)
  - Go-Back-N, and selective repeat (reject), discussed below.
- **Protocol pipelining (concept)**
  - Protocol pipelining allows sending multiple yet-to-be-ACKed packets.
    - This utilizes sender, maintain reliability, while dramatically improving performance
    - Requires buffering and increase range of sequence numbers.
  - Two approaches towards pipelined error recovery

Go-Back-N (GBN) ARQ	Selective Repeat (Reject) ARQ
• Sender can have up to N unACKed packets in pipeline.	• Sender can have up to N, unACKed packets in pipeline.
• Receiver sends ACKs only in order, if there is a gap, sender discards the ACKs after it, and retransmits all unACKed packets.	• Receiver sends one ACK for each packet, regardless of the order, the packets that arrive out of order are buffered until the gap is closed, then reordered, the window is advanced to the next not-yet ACKed packet.
• Sender maintains timer for the oldest unACKed packet, retransmits all unACKed packets on timeout.	• Server maintains timer for each unACKed packet, retransmits the unACKed packet on timeout and restart the timer.

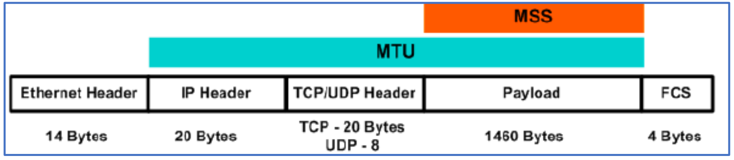
Lecture 5 – Transport Layer II

- **TCP provides data transfer that is:**
  - **Reliable, in-order, byte-stream oriented**
  - **Connection-oriented** (through handshaking)
  - **Point-to-point:** one sender, one receiver.
  - **Pipelined:** several yet-to-be-ACKed segments can be sent bounded by a window size.
  - **Full duplex:** bidirectional data flow in same connection.
  - **Flow controlled:** sender will not overwhelm receiver.
  - **Congestion controlled:** network will not be congested (crowded).

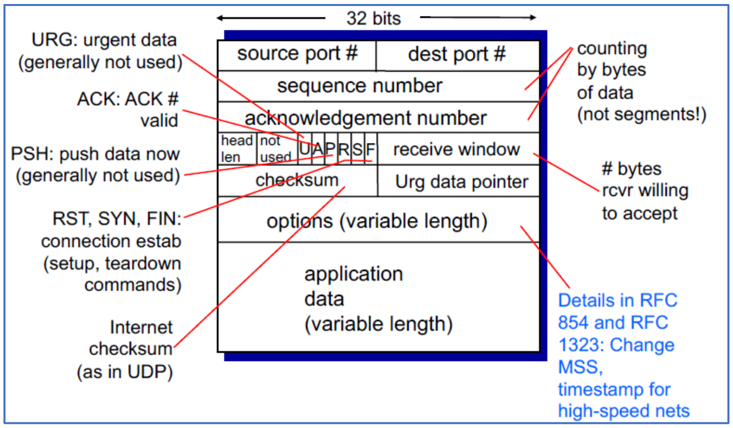
Transmission modes in networks

- **Simplex:** a sender can only send data in one direction.
- **Half duplex:** data can be sent bidirectionally, **but not at the same time.**
- **Full duplex:** data can be sent bidirectionally **and at the same time.**

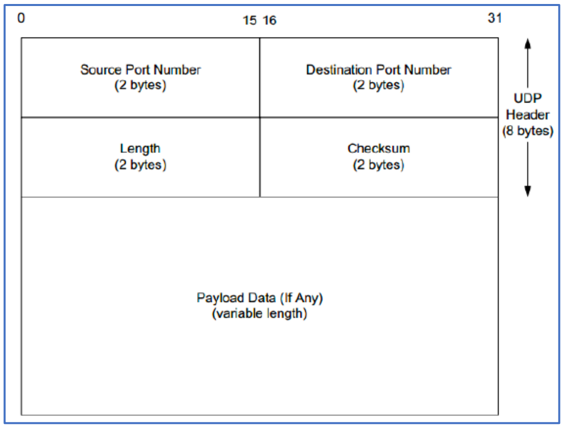
- **Connection management with TCP**
  - **Handshaking:** the process of establishing connection between two different hosts, during which they agree to some connection parameters (variables)
  - **Two-way handshake:** (request + accept) won't work for establishing TCP connections.
    - Both parties won't be able to agree on timeout (retransmitted sender messages may create half-open connection or duplicated data).
    - Won't allow bidirectional data transfer, since only one party will be able to synchronize its Initial Sequence Number (ISN) with the other.
      - **ISN**: a randomly generated (to prevent spoofing) number that allows client/server to know the order of data and to help with error control (the next expected byte number (in this stream)).
  - **Three-way handshake**: the process of establishing the reliable TCP connection between the client and the server in 3 steps (one step = one packet going between A and B):
    - Step1 (SYN): client sends its ISN x to the server, initiating the connection (SYNchronize)
    - Step2 (SYN-ACK): server responds with SYN ACK $x + 1$ then sends its own ISN y to client.
    - Step3 (ACK): client responds with SYN ACK $y + 1$ and the connection is established, data can then be transferred bidirectionally (this packet may contain data)
    - At the end, both parties send a FIN flag to terminate connection, and both needs to ACK the FIN.

Maximum Segment Size (MSS)	**Maximum Transmission Unit (MTU)**
• Max payload (message = application layer data) size that can be transmitted in one transport-layer **segment.** • TCP default = 536 bytes, can be changed only during connection setup.	• Max payload (IP packet = network layer data) size that can be transmitted in one datalink layer **frame.** • Ethernet default MTU = 1500 bytes that is a standard, however, it can also be changed during connection setup.
	

- **TCP/UDP segment structure**
  - When MTU is greater than some threshold, TCP-only sender does segmentation.
    - UDP data are not segmented, that's why we may say "UDP datagram" instead of "UDP segment".
  - **(TCP Packet) segmentation:** splitting payload into segments of equal size.



32 bits
source port #
dest port #
sequence number
acknowledgement number
head len
not used
U A P R S F
receive window
checksum
Urg data pointer
options (variable length)
application data (variable length)
URG: urgent data (generally not used)
ACK: ACK # valid
PSH: push data now (generally not used)
RST, SYN, FIN: connection estab (setup, teardown commands)
Internet checksum (as in UDP)
counting by bytes of data (not segments!)
# bytes rcvr willing to accept
Details in RFC 854 and RFC 1323: Change MSS, timestamp for high-speed nets



0
15 16
31
Source Port Number (2 bytes)
Destination Port Number (2 bytes)
Length (2 bytes)
Checksum (2 bytes)
Payload Data (If Any) (variable length)
UDP Header (8 bytes)

- **Round-Trip Time (RTT) and timeout in TCP**
    - The time for a signal to be sent and ACKed (including propagation (transfer) time)
    - RTT can be estimated by measuring the time taken by several recent transfers and taking the average.
    - TCP timeout can then be set accordingly.
        - Too short timeout ⇒ many unnecessary retransmissions.
        - Too long timeout ⇒ slow reaction to segment loss.
- **TCP timeout:**
    - TCP maintains a time for the oldest unACKed segment and retransmits on timeout.
    - **Typical scenarios:**
        - **Lost ACK:** segment was received, ACK was lost, receiver ACKs the retransmitted segment.
        - **Premature timeout:** segment was received, ACK was delayed, sender retransmitted, sender ignores second ACK.
        - **Cumulative ACK:** two segments are sent, ACK for first one was lost, ACK for second was received, sender will ***not*** retransmit anything in this case, receiver couldn't have ACKed the second segment without getting the first one, the first ACK was indeed lost.
    - **Doubling time interval:**
        - When TCP retransmits, it doubles the timeout value (become twice more patient) to avoid congesting the (potentially already congested) network with too many retransmissions.
        - But this can make timeout get very long, increasing end-to-end delay.
    - **TCP fast retransmit** (useful with pipelined implementation)
        - An improvement: on receiving triple duplicate ACKs, don't wait for timeout, retransmit immediately.
        - This is because it's highly likely that the packet was lost and not delayed, since 3 ACKs are received.
- **TCP flow control:**
    - Receiver controls sender so it won't overwhelm it (overflow its buffer) by sending too much, too fast.
    - Can be achieved by having the receiver informing the sender about its available buffer size (by including rwnd value in the TCP header of segments it sends to the sender), so the sender acts accordingly (by limiting the amount of unACKed data it sends (i.e., controlling window size)).
    - Multiple connections for a single file can also be used.
- **Principles of congestion control:**
    - **Network congestion:**
        - Load on a network becomes higher than its capacity (not uniform)**.**
        - Multiple flows competing for bandwidth.
    - **Congestion consequences**
        - Queuing delay, packet loss (router buffer overflow), blocking of new connection, bandwidth wastage (due to retransmission), low goodput or even hardware physical damage.
            - **Goodput:** application-level throughput (i.e., number of useful bits delivered per unit time).

Goals of congestion control

- Adjusting to the bottleneck/variations in bandwidth
- Sharing bandwidth between flows
- Maximizing throughput

    - **Ideally, no problem will happen if:**
        - Router buffer is infinite (lol)
        - Sender has full knowledge of router buffer and sends only when they are available.
        - Sender has full knowledge of packet state (whether its lost) and retransmits only then.
        - But, none of this is possible ⇒ realistic methods for congestion control are needed.
    - **General approaches for congestion control**
        - **Do nothing)**
            - Many packets will drop, totally unpredictable performance, may lead to **congestion collapse.**
        - **Reservations:** pre-arrange bandwidth allocations for flows
            - Requires negotiation before sending packets, must be supported by the network.
        - **Dynamic adjustment**: use probes (checks) to estimate level of congestion and adjust accordingly.
            - Requires distributed coordination.
    - **Congestion collapse:** when congestion prevents/limits transfer of useful information.
        - Occurs at network bottlenecks (choke points) such as the connection b/w LAN and WAN.
        - Leads to high traffic, low available useful throughput, packet delay/loss, poor Quality of Service.
        - **Nagle's algorithm** is one of the algorithms used for improving efficiency of TCP/IP networks by reducing the number of packets that need to be sent over the network.

- **TCP congestion control**
    - TCP uses congestion window and receiver window to regulate data flow, minimize (avoid) congestion and improve network performance by transmitting data no more than window size.

<table>
<tr><th>Congestion window (<code>cwnd</code>)</th><th>Receiver (advertised) window (<code>rwnd</code>)</th></tr>
<tr><td>• Max amount of unACKed segments that can be sent.
• Determines send rate (how much data in transit)
• <code>cwnd</code> is flow controlled by sender.
• <code>cwnd</code> is used to avoid overrunning routers in the middle of the network (avoid congesting it)</td><td>• Receiver host buffer size
• Determines max receive rate.
• <code>rwnd</code> is flow controlled by receiver.
• <code>rwnd</code> only protects receiver, it doesn’t provide congestion control.</td></tr>
<tr><td colspan="2"><b>Congestion control require that</b>
$effective\_window = min(cwnd, rwnd) - \#inflight\_packets$
$LastByteSent - LastByteAcked = \#inflight\_packets \leq min(cwnd, rwnd)$
$TCP\ sending\ rate \cong cwnd/RTT$</td></tr>
</table>

-
    - **To utilize (while not congesting) network:**
        - TCP sender starts with a default small window size (**slow start** threshold = 1 MSS)
        - On each ACK, increment `cwnd` by 1 **MSS (additive increase).**
            - Incrementing `cwnd` on each ACK effectively doubles the sending rate per RTT.
            - This exponential increase stops if one of the following happens.
                - When detecting a packet loss: reset `cwnd` = 1 MSS, set $ssthresh = cwnd/2$
                - If $cwnd \geq ssthresh$, enter **congestion avoidance** mode.
                    - Increase `cwnd` by 1 MSS every RTT.
                - If three duplicate ACKs are received, fast retransmit and enter **fast recovery** mode.
                    - Increase `cwnd` by 1 MSS for every duplicate ACK until the segment is received successfully, then enter congestion avoidance mode again.
                    - Fast recovery is recommended, but not required component of TCP.
        - When sender doesn’t receive ACK within timeout (or receives 3 duplicate ACKs), TCP **detects possible congestion** and starts to **adjust** (using some algorithm)
            - Stop incrementing `cwnd` to avoid losing more packets.
            - Check the network bandwidth, if low, respond to congestion

TCP Reno algorithm	TCP Tahoe algorithm
On loss detection by timeout, reset `cwnd` = 1 MSS On loss detection by 3 duplicate ACKs, cut `cwnd` in half **(multiplicative decrease)**	On any loss detection, reset `cwnd` = 1 MSS

- **TCP fairness**
    - If K TCP sessions share the same bottleneck link of bandwidth R, each should have average usage rate of R/K
    - Additive increase and multiplicative decrease guarantee TCP fairness.
    - Web browsers can utilize multiple parallel TCP connections to the server, taking unfair share of the bandwidth.
- **UDP is not fair:** multimedia apps don’t want the overhead of managing congestion control that may make their apps slower to avoid congestion, so they use UDP, tolerating packet loss, while being faster in practice.

Lecture 6 – Network Layer I

- **Network layer**
    - Responsible for routing packets from source to destination.
    - Network layer protocols exists on every host **and router** (unlike transport, hosts only)
    - The smallest exchangeable data unit is called a packet, or datagram.
    - **Router's functionality:**

> ***Routing algorithm** defines a **routing protocol.***
> ***Routing protocol** sends control information to the router's **routing table(s).***
> ***Routing tables** are used to construct the router's local **forwarding table**.*

- **Forwarding:** a local router action
    - Moving packets from router input to appropriate output line (or broadcast channel).
    - Appropriate output is chosen based on the information in the router's **forwarding table**.
    - **Forwarding table**
        - Is updated periodically to reflect network state.
        - Drastically affects network performance.
        - Matches longest prefix matching destination IP address to choose the output link
- **Routing:** a network-wide process
    - Determine the least cost path taken from host A to B using a **routing algorithm.**
    - **Routing algorithm classifications**

Adaptive (dynamic) routing algorithm	**Non-adaptive (static) routing algorithm**
uses dynamic information (network topology, load, delay, etc.) to select routes and update forwarding tables.	the routes are selected initially and never changes (changes very rarely and manually by reconfiguring forwarding tables)
Global (Centralized)	**Decentralized**
Each node knows the entire network topology. Each node knows about all other nodes.	All nodes use the same (distributed) algorithm. Each node contributes to the result by providing information about its immediate neighbors only.
Load sensitive	**Load insensitive**
Link costs change dynamically, to reflect network congestion.	Link costs are static (most routing algorithms).

	Link-state (e.g., OSPF)	**Distance vector (e.g., BGP)**
Algorithm class	Centralized	Decentralized
Brief description	Each node talks to **each other node**, regarding information about its neighbors	Each node talks to **its neighbors only** regarding the estimated least cost path to all nodes.
Complexity	$O(\|N\| * \|E\|)$ Quadratic convergence, relatively fast	No guarantee on maximum number of messages before convergence, may not converge at all.
Robustness	Broadcasting incorrect link information affects only node neighbors.	Broadcasting incorrect information may affect the whole network.

- Router's architecture:
    - **Control plane:** router's CPU, handles management and routing protocols.
    - **(Forwarding) data plane** consists of input links, output links, and switching fabric.
    - **How it works**
        - **Input links** buffer arriving data, extract packet header, verify its correctness (e.g., using checksum), decrement time-to-live, keep statistics, sends packets to the CPU.
            - Each input link may have its own CPU and snapshot of the forwarding table.
        - **The CPU** executes the routing protocol by extracting destination address from packet header and using the forwarding table (maintained by the CPU) to determine appropriate output link.

      - **The switching fabric** (network of connections inside the router) propagates packets from input link to the output link chosen by the CPU.
        - Switch fabric can be a shared memory, a communication bus, or an interconnection network.
      - **Output links** buffer the packets received from the switching fabric, reverses the operations done by the input links then transmits the data.
    - **Other issues**
      - **Input and output link buffers** are used to deal with differences between links and switching fabric speeds.
        - **Packet scheduling problem:** which packet to process first from a buffer? FIFO, Weighted Fair Queuing (WFQ), etc.
      - **Bottlenecks** (limited buffer capacity, link transmission speed, speed of switching fabric, configuration/search time in forwarding table, etc.) are dealt with by router designers.

- **Network service model:**
  - Defines the required services and service qualities provided by a network layer channel.
  - **Example (for a single datagram)**: guaranteed delivery (before timeout)
  - **Example (for a flow of datagrams)**: in-order delivery, minimum bandwidth, interpacket gap
    - **Interpacket gap:** minimal time gap between consecutive packets to allow receiver to be ready for the next packet (clock recovery, power-up, etc.)
- **Internet Protocol (IP):**
  - The network layer protocol used by the internet to exchange data between internet-connected hosts.

Global (public) IP addresses	**Local (private) IP address**
• Accessed publicly by anyone on the internet. • Used for global communications between any two internet-connected hosts.	• Non-routable to the internet • Used for local communications in a private network.

IPv4 address	**IPv6 addresses**
• 32-bit (4 octets) identifier for a router/host interface ▪ **Host** typically has two interfaces (wired Ethernet, wireless 802.11) ▪ **Router** typically has multiple interfaces.	• 128-bit addresses introduced to compensate the lack of IPv4 addresses. • Eight 16-bit fields, each represented by a 4-digit case-insensitive hexadecimal number, separated by a colon

Private IPv4 addresses

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255
- 169.254.0.0 – 192.254.255.255 (for Automatic Private IP Addressing (APIPA))

Special IPv4 addresses

- 0.0.0.0 – invalid, unreachable unknown, not assigned.
- 255.255.255.255 – broadcast
- 127.0.0.1 **-** localhost

Rules for IPv6 addresses

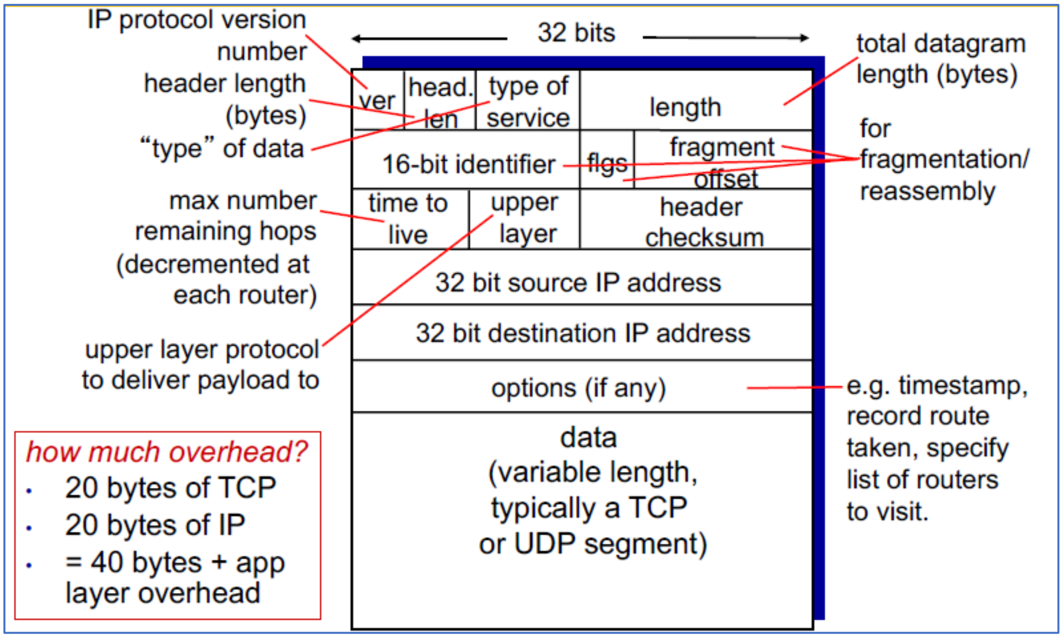
- Leading zeros are optional.
- Successive fields of zeros are represented by :: but only once.
- Recommended to use :: only once with the rightmost set of consecutive zeros.
- In URL, enclose IPv6 address in []
- Subnet prefix also uses CIDR notation.
- IPv4 can be represented in IPv6 notation with leading zeros (deprecated).

Special addresses

: : 1 – localhost, :: – Unspecified.

  - **Subnet:** portion of a network with hosts having the same IP address prefix, can reach each other without needing a router (first and last IP address bit cannot be used for subnets)
    - **Classless Inter-Domain Routing (CIDR):**
      - Internet address assignment strategy, the idea of dividing the network into subnets with variable length masks, to minimize the size of routing table (only prefix is included).
      - **CIDR notation:** a shorter way to represent IP addresses with subnet mask in format (a.b.c.d/x)
    - **IP addressing**
      - ISP gets a block of addresses from ICANN.
      - ISP allocates a portion of its address space to each organization it supplies.
      - Each organization may have smaller subnets, each serving several hosts.

- **IPv4 datagram format**


IP protocol version number
header length (bytes)
"type" of data
max number remaining hops (decremented at each router)
upper layer protocol to deliver payload to
32 bits
ver
head. len
type of service
length
16-bit identifier
flgs
fragment offset
time to live
upper layer
header checksum
32 bit source IP address
32 bit destination IP address
options (if any)
data (variable length, typically a TCP or UDP segment)
total datagram length (bytes)
for fragmentation/ reassembly
e.g. timestamp, record route taken, specify list of routers to visit.

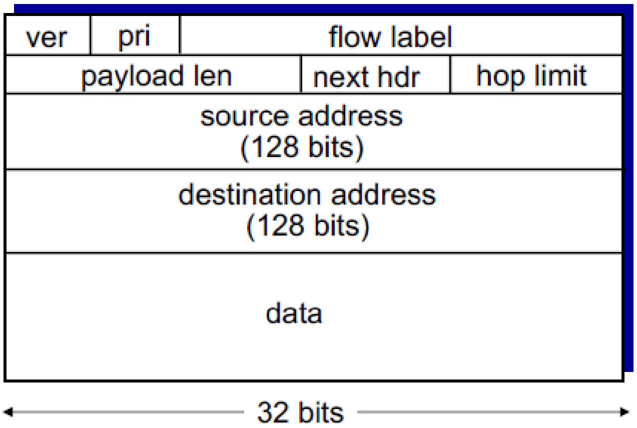
how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

Segmentation (happens at transport layer): chopping large application message (larger than MSS), into smaller payloads, each one is sent in its own segment.

Fragmentation (happens at network layer): chopping large IP datagram (packet larger than MTU) into smaller payloads, each sent in its own (ethernet) data link frame and **reassembled** upon arrival (using IP header info).
- 16-bit id: fragment length and id.
- fragflag: true → this is not the final fragment.
- offset: determine fragment seq.

- **IPv6 datagram format**



New fields in IPv6:
- **Priority**: identify priorities among datagrams in flow.
- **Flow label:** identify datagrams in the same flow.
- **Next header:** identify upper layer protocol for data.

Fields from IPv4 removed from IPv6:
- **Checksum:** removed to reduce processing time at each hop**.**
- **Options:** included in the upper layer pointed to by next hdr.

ICMPv6 was introduced with additional message types and multicast group management functions.

- **IPv6 tunneling:**
  - Transferring IPv6 packets as a payload across an IPv4 channel; since all routers cannot be upgraded at once to support IPv6 and network cannot operate with both IPv4 and IPv6 simultaneously.

- **Dynamic Host Configuration Protocol (DHCP)**
  - Application layer protocol, UDP port 67 (at destination server), 68 (source port used by client)
  - Automatically assigns (IP address, subnet mask, default gateway, address of primary/secondary DNS servers) to hosts dynamically as they join the network.
    - **Default gateway:** IP address of the first-hop router for that client.
  - **How it works:**
    - Host joining the network broadcasts **"DHCP discover"** message.
    - DHCP server sends **"DHCP offer"** to host offering a free IP address.
    - Host sends **"DHCP request"** to DHCP server (usual behavior: accepting the offer).
    - DHCP sends back **"DHCP ack"**.

Address Resolution Protocol (ARP): link layer protocol providing IPv4 to MAC address translation **(check lecture 8)**

- **Software Defined Networking (SDN)**
  - An approach to network management that attempts to centralize network intelligence in one component by separating the routing process **(control plane)** from the forwarding process **(data plane)**.
  - This allows dynamic management of huge networks, making it more flexible and optimized for certain services that change dynamically (same communication links being used for DNS, DHCP, FTP, etc.).
  - **How it works:**
    - A controller software (Network Operating System) is running on a network host that have a global view of the network, it computes and distributes forwarding (flow) tables to all routers in the network.
    - No need to have a distributed control protocols running on every router (traditional networking).
    - No need for special complex network hardware (e.g., firewall devices).
  - **Early implementations**
    - **OpenFlow:** communication protocol allowing access to the forwarding plane.
      - Each router contains a flow (**match plus action**) table (computed and distributed by controller) containing tuples (rule, action, stats)
        - **Flow:** group of consecutive packets having the same flow label.
        - **Match plus action** abstracts different types of services provided by different network hardware.

	Match	**Action**
Router	Longest destination-IP prefix	Forward to some output link
Switch	Destination MAC address	Forward or flood (forward to all active ports except sender)
Firewall	IP address, TCP/UDP port #	Allow or block traffic
NAT	IP address, port #	Rewrite address and port

      - **Generalized forwarding:** for a flow matching a specific **rule**, the router takes an appropriate **action**, it also stores **stats**.
        - **Rule** can be a condition on source/destination IP/MAC address, port number, etc.
        - **Action** on matching packets can be to: drop, forward, modify, send to controller for more processing, etc.
        - **Stats** stored such as #packets, #bytes matching the rule, last-updated time for record.

Generalized forwarding	**Destination-based forwarding**
• Packet forwarding approach used by OpenFlow. • Packets are forwarded based on the rules defined by the controller in the flow table. • Entries in flow table are constructed based on multiple parameters of the network and modified dynamically, does not necessarily take the destination address into account.	• Traditional packet forwarding mechanism. • Packets are forwarded based on the destination address in the packet, a forwarding table entry is used to redirect the incoming packet. • Entries in the forwarding table are constructed based on a reversed path from destination to source.

    - **NOX:** platform for C++ programmers to develop network controllers that work with OpenFlow.
- **Network Address Translation (NAT)**
  - Service (method) provided by routers that:
    - Provides host security by hiding the source IP and port from the public network.
    - Help compensate the lack of IPv4 address by using only one public IP for an entire subnet.
      - Each host in the subnet has private IP addresses.
      - A NAT enabled router acts on behalf of all hosts by modifying IP addresses and port numbers for outgoing and incoming traffic according to router's NAT table.
  - **NAT drawback:** misuse of ports, routers modifying layer3 packets (violation of end-end communication concepts), a workaround instead of using IPv6, interference with P2P applications.
  - **Universal Plug and Play protocol (UPnP)** allow discovering nearby NAT-enabled router.

Lecture 7 – Network Layer II

- **Network layer relies on protocols for control and routing.**
    - **Routing protocols**
        - Responsible for setting up routing tables.
        - **Examples – check last tutorial for more details.**
            - Routing Information Protocol (RIP) – Application layer
            - Border Gateway Protocol (BGP) – Application layer
            - Open Shortest Path First (OSPF) – Link layer
            - Protocol Independent Multicast (PIM) – Transport layer
    - **Control protocols**
        - **Internet Group Management Protocol (IGMP)** – Network layer
            - Used to establish IP multicast group membership.
                - **IP multicast:** sending IP datagrams to multiple hosts (group) in a single transmission, as opposed to unicast – sending to a single host.
        - **Internet Control Message Protocol (ICMP)** – Network layer
            - Helper protocol used by network devices to send error/operational (success/failure) messages, or to do simple diagnostic queries (e.g., ping, traceroute)
            - Messages are encapsulated in IP datagrams as 8-bytes payload **(type, code, checksum, info)**
                - type of ICMP message: request (by host to router) or reply (from receiver)
                - code (subtype, has a different meaning for each type)
                - checksum (over entire message)
                - additional information (depends on type, code) or zeros if not applicable.

Example scenario	ICMP messages
Host pinging another host	- OS kernel translates each ping to an **ICMP echo request** (type=8, code=0) - The pinged host replies with **ICMP echo reply** (type=0, code=0).
Host *A* asks host *B* for a timestamp (elapsed milliseconds after UTC midnight of the current day)	- *A* issues an **ICMP timestamp request** (type=13, code=0) - *B* issues an **ICMP timestamp response** (type=14, code=0)
Host discovering the router on the network	- Host issues an **ICMP router solicitation** message (type=10, code=0) to all routers on the network. - Router issues an **ICMP router advertisement** (type=9, code=0) to all hosts, indicating that it is available.
IP datagram is discarded by host/router because of TTL=0	- Host/router sends back an **ICMP error message: time exceeded** (type=11, code=0) - It also sends IP header of the discarded datagram and first8 bytes of payload back to the application.
Client requests a service at some non-listening port.	- Server issues an **ICMP error message: destination port unreachable** (type=3, code=3)

Lecture 8 – Link Layer

- **Datalink (link) layer**
    - Responsible for transferring network layer datagram from one node to a physically adjacent node over a link.
    - **Terminology:**
        - **Frames:** layer-2 packets encapsulating network datagram.
        - **Nodes:** hosts and routers.
        - **Link:** communication channel connecting two nodes (i.e., wired, wireless)
    - **Network Interface Card/Controller (NIC)**
        - **Alternative names:** network adaptor, LAN adaptor, physical network interface.
        - Hardware chip in every host and router (attached to system bus) that connects it to a network by implementing link layer and physical layer protocols.
        - **How it works:**
            - Sender NIC controller encapsulates the datagram in a frame by adding a header.
                - Header contains MAC destination address, info for error checking, flow control, etc.
            - Sender passes the data to the receiver side that examines the header, extracts the datagram, and passes it to the network layer.
    - **Media Access Control (MAC)**
        - MAC is considered a sublayer of link layer, with **multiple access protocols** (channel access methods) operating in it – **discussed below**.
            - FDMA, TDMA, CDMA, WDMA, CSMA, CSMA/CD, ALOHA, Slotted ALOHA
        - Responsible for controlling access to the medium link between two nodes.
        - Defines the rules by which the frame is transferred over the link.
        - MAC *address*: unique hardware address assigned to network nodes by NIC vendors.
            - 6-bytes (48 bits), represented as 6 hexadecimal dash/colon-separated octets.
            - IEEE manages MAC address space, allocating blocks to manufacturers.
    - **Common link layer protocols**

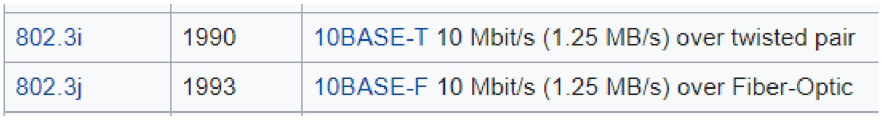
Protocol	Description
ARP	**Address Resolution Protocol** (works between layer 2 and layer 3) o Provides IPv4 to MAC address translation, used by Ethernet devices (e.g., ethernet switches) to know the MAC address of a target IP destination, for forwarding traffic on the same network. o **How it works:** ▪ Host sends a broadcast ARP packet to all devices in the network asking which device has that IP address. ▪ Each device has an ARP table with (IP, MAC, TTL) entries for the devices it discovered in the network. o The ARP table is learned and updated dynamically (Plug-and-play protocol) o Entries are said to be in **soft state,** they timeout (removed) unless refreshed. ▪ It checks the table and responds with MAC to the broadcast if it has the intended IP.
Frame relay	Less popular WAN protocol used commonly for voice communications.
IS-IS	**Intermediate System to Intermediate System protocol**
PPP	**Point-to-Point Protocol:** designed for point-to-point links with one sender, one receiver.
HDLC	**High-level Data Link Control:** was used for point to multipoint connections.

- MAC is not a protocol, it is a **sublayer** with protocols (e.g., FDMA) operating in it.
- Ethernet is not a protocol, it is **a technology** (group of protocols)
- VLAN is not a protocol, it is **when the broadcast domain is physically separate, but logically treated as one**.
- DOCSIS is not a protocol, it is a **standard** for data transfer over TV cables.
- IEEE 802 is not a protocol, it is a family of standards for LAN, MAN, and PAN.
    - The most important ones being **802.3** for Ethernet, **802.11** for Wireless LAN (WLAN), and **802.1Q** for VLAN.

- **Ethernet:** dominant (most widely used) wired network technology.
    - **History**
        - Ethernet can be considered a collection of protocols for wired networks (e.g., LAN, MAN, WAN)
        - Current standards are controlled by IEEE.
        - Ethernet LAN originally consisted of a single coaxial cable called **ether** or segment.
        - It used the bus topology and operated at 10 Mbps, with 500m length limit.
        - Now organizations use star topology (switch in center, nodes don't collide), with ethernet cables supporting up to 40 Gbps but a length limit of 100m (fiber optics are used for longer distances)
    - **Ethernet Frame structure**
        - **Preamble (8 bytes):** 7 copies of "10101010" followed by "10101011" used to synchronize clock rate between sender and receiver.
            - Some sources divide this into preamble (7 bytes) and Start Frame Delimiter (1 byte).
        - **Addresses (12 bytes):** destination/source MAC addresses, each of 6-bytes
            - If source = destination, drop frame.
            - If destination = FF:FF:FF:FF:FF:FF, broadcast.
            - Other valid MAC address, pass to network layer.
        - **Type (2 bytes):** indicates higher layer protocol (e.g., IP, Novell IPX, AppleTalk)
        - **Payload (variable length 46-1500 byte)**
        - **CRC (typically 4 bytes)** used for error detection – **discussed below**.
            - Some sources generalize this to Frame Check Sequence (FCS)
    - **Ethernet Switch**
        - Active, self-learning (plug-and-play, no configuration needed), transparent-to-hosts **link layer device** that stores and forwards ethernet frames between nodes.
        - **How it works:**
            - Switch examines an incoming frame, adds sender MAC address to the switching table if it wasn't there (learning).
                - **Switching table entry:** (host MAC address, interface to reach host, insertion timestamp).
            - Switch examines destination MAC address, then selectively forward the frame to the device having that address, or floods (forward to all except sender) if the address is not found in the switching table.
        - **Switch benefits**
            - Using switches in a star topology eliminates collisions between nodes (no bandwidth wastage, max throughput achieved)
            - Allows using heterogeneous links (of different types and speeds)
            - Ease of network management (detecting and disconnecting malfunctioning adaptors, gathering statistics about bandwidth usage, collision rates, and traffic types).
        - **Switch vs Router (main differences)**

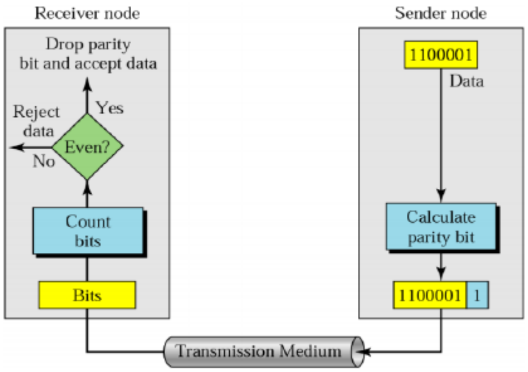
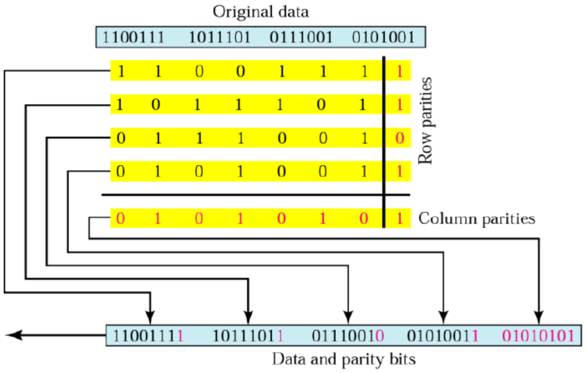
	Switch	**Router**
Layer	Examines link layer packets	Examines network layer packets
Forwarding table	Learned from the network by flooding (e.g., ARP packets)	Computed using routing algorithms
Typically used with	Ethernet frames and MAC addresses	IP datagrams and IP addresses.

    - **Ethernet is connectionless:** no handshaking between sender and receiver NICs.
    - **Ethernet is unreliable:** no ACKs or NAKs are being used.
        - Relies on higher layer (e.g., TCP) reliability mechanisms.
    - **Standards**



802.3i	1990	10BASE-T 10 Mbit/s (1.25 MB/s) over twisted pair
802.3j	1993	10BASE-F 10 Mbit/s (1.25 MB/s) over Fiber-Optic

- **Link layer Services**
  - **Framing:** encapsulation of network datagrams into link layer frames (adding frame header)
  - **Link access:** service provided by MAC protocol.
  - **Reliable delivery:** using ACKs and retransmissions, or error correction on wireless links.
  - **Flow control:** making sure sender doesn't overwhelm receiver.
  - **Half-duplex and full-duplex transmission (check lecture 5)**
  - **Error Detection and Correction (EDC):**
    - Bit errors caused by noise or signal attenuation are detected by receiver using EDC redundant bits added by the sender side and sent along with data.
    - On error detection, the receiver drops frame, tells the sender to retransmit, or **correct** the error itself.
    - Error detection is not 100% reliable, all methods below may miss errors.
    - **Error types:**
      - Single bit error (one bit is flipped)
      - Burst error (multiple consecutive (or nearly consecutive) bits corrupted).
    - **Error detecting techniques (calculation methods for EDC bits).**

<table>
<tr><th>Single (even) parity check</th><th>Multiple (two dimensional) parity check</th></tr>
<tr><td>• 1 is added to the block if it contains odd number of ones, 0 otherwise, making the total number of 1's even (another variant uses odd parity)
• Method can only detect when an odd number of single bit errors happens</td><td>• Single parity check calculated for each row and column of the data. A single parity error can be corrected using this method (Forward Error Correction (FEC).
• Method won't detect the case when 4 bits (2 in one data unit and corresponding 2 in another data unit) are all flipped.</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><th colspan="2">Checksum</th></tr>
<tr><td colspan="2">• Sender divides the data into k segments, each of size m bits.
• Sender sends a negated (1 complement) sum of k segments along with the data (the checksum).
• Receiver adds the data to the checksum, if the result is zero, data is accepted, otherwise discarded.</td></tr>
<tr><th colspan="2">Cyclic Redundancy Check (CRC)</th></tr>
<tr><td colspan="2">• Most popular widely used error checking method because of its efficiency in detecting multiple bit errors.
• Sender and receiver agree on the generator (G) standard value.
○ IEEE standard constant binary (8, 12, 16, or 32)-bit number.
○ First bit in the generator must be 1.
• Sender calculates a CRC code $R = (D * 2^r)$ **mod** G and sends it as EDC.
○ Multiplication by 2^r is equivalent to adding r trailing zeros.
○ The bold **mod** is a special mod calculated using modulo-2 binary long division and is not the same as the usual mod.
• Receiver checks that concat(D, R) **mod** $G = (D * 2^r \text{ XOR } R)$ **mod** $G = 0$
○ If not, an error is detected.
○ **Fact:** $A \text{ XOR } B = (A + B) \% 2 = (A - B) \% 2$

Data (D): d-bits.
Generator (G): $r + 1$ bits.
CRC code (R): r bits.

Generator is generally an n^{th} order polynomial $G(x)$, evaluated at $x = 2$ since all computer binary math is done in modulo-2 arithmetic.</td></tr>
</table>

- **Multiple access protocols ≡ channel access methods ≡ MAC sublayer protocols**
    - Distributed algorithms defining the rules for nodes sharing a **broadcast domain**, they control how communication can be done without **interference** or **collisions**.
        - **Broadcast domain:** logical division of the network in which all nodes share a common **broadcast link** that can be a shared wire (e.g., Ethernet cable) or medium (e.g., common Radio Frequency (RF))
            - When any node transmits a frame, all other nodes in the broadcast domain receive a copy.
            - **Ideally**, multiple nodes sharing a broadcast channel must fairly share the channel throughput while maximizing link utilization without the need of complex coordination mechanisms or a specialized coordinator (fully decentralized).
        - **Interference:** two nodes transmitting data at the same time.
        - **Collision:** a node receives two or more signals at the same time.
    - **Categories of multiple access protocols**

Channel partitioning

"divide the channel between hosts equally."

*efficient with **high** load on the link, inefficient with **low** load*

- **Frequency Division Multiple Access (FDMA):**
    - Divide the channel into non-overlapping frequency bands.
    - Assign fixed bands to each link user, unused time go idle.
    - Two communicating users must send/listen at the same band.
    - Not to be confused with FDM (Frequency Division Multiplexing) operating in the physical layer.

- **Time Division Multiple Access (TDMA)**
    - Assign timeslots of fixed length (e.g., packet transmission time) to each user.
    - A user can only use the link during its slot, unused slots go idle.

- **Wavelength Division Multiple Access (WDMA)**
    - Used with optical fiber links, assigns wavelengths to users instead of frequencies.

- **Code Division Multiple Access (CDMA) – Check tutorial 14 for the math used.**
    - Used with mobile devices sharing data through air.
    - Allows simultaneous transmissions between different nodes.
    - Sender and receiver must agree on a CDMA code that will be used to encode/decode the data.
        - CDMA codes must follow specific mathematical properties.
        - Signals interfere (add up) in-transit.

Random access

"let collisions happen, recover from them."

*efficient with **low** load on the link, inefficient with **high** load*

- When a node has packet to send, transmit at full capacity.
- With no coordination between nodes, collisions will happen. Detect and recover from them (e.g., using randomly generated delayed retransmissions).
- **Example protocols (discussed below)**
    - ALOHA, slotted ALOHA, CSMA, CSMA/CD (used by original Ethernet), CSMA/CA (used by 802.11).

Taking turns

"allow only one node to transmit data over the channel at a time."

Tries to take the best of both pervious approaches; to work well with both high and low load.

Polling protocol: one master node coordinates access to the link, it polls the nodes one by one (round-robin) telling them the maximum number of frames they need to transmit.

- **Cons:** polling delay, latency, single point of failure (the master node)
- **Used by** 802.15 and Bluetooth.

Token-passing protocol: no master node, a node that has the "token" frame can transmit up to some maximum number of frames, then pass the token to the next node (round-robin)

- Immediately pass the token if nothing to transmit.
- **Cons:** token overhead, latency, single point of failure (the token).
- **Used by** Token-Ring LAN technologies.

- **ALOHA:** Additive Links Online Hawaii Area

Pure ALOHA	Slotted ALOHA
Assume a broadcast domain of N nodes and a frame transmission time of 1 unit. When a frame arrives (at t_0) at node x, algorithm **transmits immediately**. $P(\text{transmission succeeds}) = P(\text{node x transmit}) *$ $* P(\text{no other node transmit in } [t_0 - 1, t_0]) *$ $* P(\text{no other node transmit in } [t_0, t_0 + 1]) =$ $= p * (1-p)^{N-1}(1-p)^{N-1}$ Efficiency of the method = $N * P(\text{transmission succeeds})$ Max efficiency is calculated by finding $p(N)$ that maximizes the efficiency, then taking the limit as $N \to \infty$ - Efficiency = $Np(1-p)^{2(N-1)}$ - Max efficiency = $\frac{1}{2e} \cong 18.4\%$	More optimized version, divide the time into slots, **nodes can only transmit at the beginning of a slot** (thus reducing number of possible collisions, they can happen only at the beginning of a slot and detected immediately, retransmit next slot) • **Pros:** simple, decentralized, active node can utilize full link capacity. • **Cons:** wasting time due to (collisions, idle slots, waiting for next slot to retransmit), needs clock synchronization between nodes (to agree on time slots). - Efficiency = $Np(1-p)^{N-1}$ - Max efficiency = $\frac{1}{e} \cong 36.8\ \%$

- **CSMA/CD:** Carrier Sensing Multiple Access with Collision Detection (**CSMA/CA** – **A**voidance)
  - When a frame arrives, nodes check channel state (whether being used or not) before transmitting.
    - If the channel was being used, wait until it's idle.
    - Sensing is done by the network adaptor; it checks the signals passing through channel.
    - Easy to implement in wired networks, hard in wireless.
  - A collision can happen due to the propagation delay (at the time of check, channel was idle, during transmission, two or more signals interfere).
  - Abort and random-delay-retransmit on collision detection.
    - Random delay is a critical parameter, long delays affect performance, small delays may not avoid collisions.
    - **Binary exponential backoff algorithm**: $delay = random\_between([0,\ \min(2^c - 1, 2^{10} - 1)])$ where c is the number of experienced collisions.
  - Ethernet originally used unslotted CSMA/CD with binary backoff for half-duplex communications.

- **DOCSIS:** Data-Over-Cable Service Interface Specifications
  - Standard specifies cable access network architecture and its protocols.
  - It allows using TV infrastructure to provide internet access to users.
  - DOCSIS utilizes the three classes of multiple access protocols we discussed above.
    - **FDMA:** downstream signal containing (internet frames, TV channels, timeslots allocation for TDMA) from CMTS to multiple users over different frequencies.
    - **TDMA:** upstream signal (internet frames, TV control messages) from multiple users to CMTS in different timeslots.

Recall from Lecture 1.

Cable network: utilizes existing television infrastructure coaxial cables.

- Unlike DSL, homes are connected to ISPs through Cable Modem Termination System (CMTS).
- The Hybrid Fiber Coaxial (HFC) shared cable carries multiple signals for multiple users at different frequencies for internet/TV.

- **Virtual Local Area Network (VLAN)**
  - Allows a single logical broadcast domain to by physically separated.
    - Allows a single LAN infrastructure to be used for multiple virtual LANs.
    - Users can then join their LAN (receive broadcast messages, etc.) even when they are connected to a different switch.
  - **Port-based VLAN**: switch ports are grouped (by switch management software) so that <u>a single physical switch operates as multiple virtual switches</u>.
    - **Features**:
      - **Traffic isolation**: frames belonging to a VLAN ports only reach these ports.
      - **Dynamic membership**: ports can be dynamically assigned among VLANs.

      - **Forwarding between VLANs:** done via routers (a router must have a link in each VLAN port group)
      - **Spanning multiple switches:** port groups can be spread among multiple physical switches; the switches must be connected through a **trunk port** that carries frames between VLANs defined over multiple switches.
        - Frames passing through trunk port must be valid IEEE **802.1Q** frames; it adds additional headers to the frames (CRC must be recomputed)
          - 2-byte tag protocol identifier
          - Tag control information (12bit VLAN ID, 3bit priority field)
    - Same functionality can be achieved through **MAC-based LAN** (devices belonging to a certain MAC address pool are assigned to a certain VLAN).

- **Multi-Protocol Label Switching (MPLS)**
  - A routing technique with additional link layer features (2.5-layer protocol)
  - **Implementation**: an extra header that prefixes packets (added between link layer frame and network layer packet)
    - **Header contents**
      - **Label (20 bit):** node identifier to be used for forwarding.
      - **EXP (3 bits, now called Traffic Class):** used for Quality of Service.
      - **S (1 bit):** bottom of stack flag, signifies that the current label is the last in the stack.
      - **TTL (8 bit):** Time-To-Live.
    - Initial goal of MPLS was to support fast IP packet forwarding using fixed length label instead of complex IP calculations (shortest prefix matching).
      - **How it works:**
        - An MPLS capable router (label-switched router) forwards packets based on the information in MPLS forwarding table (label, outgoing link) without inspecting IP address.
        - **MPLS forwarding** decisions are more **efficient** and **flexible** that IP forwarding decisions, they use **traffic engineering** methods to precompute **optimal paths** and **backup paths** and switch between them (**fast reroute**) quickly if link fails (useful for Voice communications over IP).
          - The dynamic information used by MPLS for forwarding decisions (e.g., link bandwidth) are propagated using a modified link-state routing protocol (e.g., OSPF, IS-IS) that floods this information into the network.
          - Two MPLS capable routers use the Resource Reservation Protocol with Traffic Engineering (RSRV-TE) signaling protocol to propagate this routing information between them.
        - **IP forwarding** decisions are based on destination IP only, while **MPLS forwarding** decisions can be based on both source and destination label.
    - Now MPLS is being used for general packet forwarding of various networking protocols.
- **Data center networks:**
  - 10's to 100's of thousands of physically close network hosts used by big companies for e-commerce, content servers, search engines, data mining, etc.
  - Provides many services to a massive number of clients, applies mechanisms for **managing/balancing load**, and **avoiding** process/network/data bottlenecks.
    - **Load balancer:** receives external client requests, distributes workload among servers, and returns the results to clients (hiding data center internals from clients).
    - **Reliability via redundancy:** having many connections between internal switches (levels of switching) to ensure reliability and increase throughput.

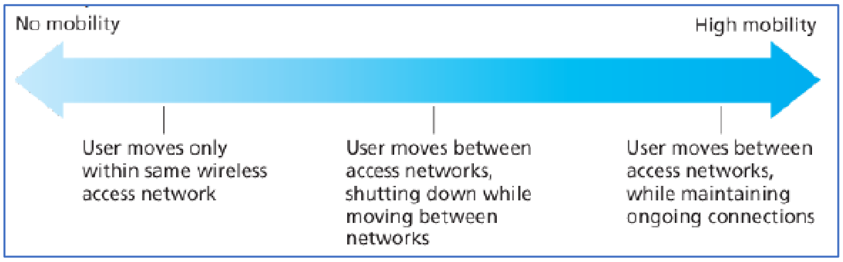
*// By now you should be able to describe using technical terms what happens from the moment you connect your device to a wired network, open the browser and query http://www.google.com and get the webpage displayed (that will be a very long story). Not http**s** as it requires understanding of some security topics to be covered in the last lecture.*

Lecture 9 – Wireless and mobile networks

- **Wireless network elements**
  - **Wireless hosts:** devices (laptops, smartphones, PCs, etc.) running network applications, they might be stationary (connected to some wireless network and staying in its range) or **mobile**.
  - **Base station:** sends and receives data between wireless hosts associated with it through **wireless links**.
    - **Example: cell towers** in cellular networks, **access points** in 802.11 wireless LAN.
    - To ***relay*** a message (packet) from wireless sender to wireless receiver means to send it indirectly through multiple intermediate nodes.
      - We say that sender used intermediate nodes as relays.
  - **Wireless links:** connecting mobile devices to base stations.
    - **Multiple access protocols** such as CDMA are used to coordinate access to the link.
    - **Wireless backbone link:** interconnects base stations.
      - However, base stations are usually interconnected using wires.
    - A wireless link is characterized by **data rate**, **transmission distance** and **spectrum** [Check]
    - **Drawbacks of wireless links (in comparison to wired):**
      - **Decreased signal strength** (signal attenuation due to **path loss** through matter)
      - **Interference by other sources** (e.g., other devices using the same frequency).
      - **Multipath propagation** (signal arriving to receiver at different times because it took different paths of different lengths and obstacles).
      - **Signal errors**
        - **Signal-to-Noise Ratio (SNR):** the larger the value the easier it is to extract signal.
        - **Bit-Error Rate (BER):** number of errors per unit time = $\frac{\text{bits received}}{\text{total bits sent}}$
        - There is a tradeoff between SNR and BER.
      - **Hidden terminal problem:** A can reach B, B can reach C, A cannot reach C (directly).
        - This problem can be solved using a wireless bridge that interconnects base stations.
- Wireless LAN (802.11)
  - **How it works:**
    - A wireless host in range of a base station (e.g., Access Point) needs to connect to the internet.
    - It **scans** the network and selects an AP to be associated to.
      - **Active scanning**
        - Host broadcasts a **probe request** (asking for available APs)
        - APs in range reply with **probe responses** (announcing their existence)
      - **Passive scanning**
        - APs are broadcasting **beacon frames** (with network Service Set Identifier (SSID) and MAC addresses) periodically to announce their existence.
    - Host sends an **association request** to one AP (AP may perform authentication)
    - The AP replies with an **association response.**
    - The host will typically run DHCP then to get an IP address in AP's subnet.
- Wireless LAN types
  - **Infrastructure (managed) mode** (e.g., Wi-Fi with access point)
    - All communications between **wireless hosts** must pass through the **base station**.
      - Hosts and station form the **Basic Service Set (BSS),** aka **cell** (for cellular networks)
    - **Handoff (handover)**: a mobile device moving out of base station (A) range to base station (B) range is handed-off from A to B (i.e., control responsibility is passed from A to B without loss or interruption of service).
    - **Taxonomy:**
      - **Single hop:** hosts connect to base station which connect to wired internet.
      - **Multi-hop:** a node can use other nodes as relays (carriers) to send data to the base station and the public internet, increases coverage area.
  - **No infrastructure (Peer-to-Peer or ad-hoc) mode:**
    - No base station, two wireless hosts can directly communicate when in range, nodes organize themselves into a network and route data between each other.

        - **Taxonomy** (nodes cannot connect to internet in both types)
            - **Single hop:** hosts connect directly to each other to communicate.
            - **Multi-hop:** hosts form a private wireless network; they can relay data to each other.
                - Mobile Ad-hoc Network (MANET) and Vehicular Ad-hoc Network (VAENT) belong to this category.

- **Mobility:** degree of movement.


No mobility
High mobility
User moves only within same wireless access network
User moves between access networks, shutting down while moving between networks
User moves between access networks, while maintaining ongoing connections

A wireless host

- In its **home network**
- Connected to its AP (**home agent**) that performs mobility function on behalf of it.
- Having a **permanent IP address** (should always be reachable using it)
- Not moving outside AP range

Is not mobile.

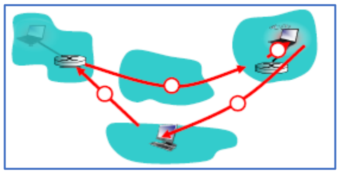
Assume that:

- The host moved to another location (**visited network**)
- It connects to an AP there (**foreign agent**)
- The AP gave it a **care-of-address (COA).**

Then the host should still be reachable using the same permanent IP (It depends on the application being run, a wireless car maintaining a TCP connection to some application while moving will need to maintain the same IP)

How to solve this problem? ***Mobility management***

- **Mobility management:**
    - **Approach 1 (using routers):**
        - Foreign agents advertise that they can route to the host permanent IP.
        - The usual process of routing table exchange will handle the rest.
        - When host moves to another foreign agent, it will do the same, and the old one will handoff the information to the new one.
        - This approach is **not scalable**, and hence not used in practice.
            - Routers must maintain information about millions of mobile users as they move.
    - **Approach 2 (using hosts) – used in practice.**

Indirect routing	**Direct routing**
- Home router handles traffic redirection from correspondents to mobile. - Results in **triangle routing problem** - Ongoing connections can be maintained (new address is transparent to correspondent) 	- Home router gives correspondents the current care-of-address of mobile to contact directly. - Solves the triangle routing problem. - Ongoing connections are interrupted (correspondent must contact the new address whenever mobile joins a new network) • Can be solved using **anchor foreign agent** (the foreign agent in the first visited network). It handles chaining traffic when mobile moves transparently from the correspondent.

- **How home agent knows the current mobile's IP?** When mobile enters the visited network, it gets **<u>registered</u>** (the foreign agent contacts the home agent giving it the COA of mobile for direct/indirect routing)

- **Mobile IP (MIP):**
    - IETF standard communication protocol for mobility management (allowing mobiles to change location while maintaining a permanent IP).
    - **Standardizes**
        - **Indirect routing:** the home router encapsulates correspondent packets by adding the COA header.
        - **Agent discovery:** the home/foreign agents advertise the routing situation by broadcasting ICMP messages (type 9).
        - **Registration:** through MIP registration request and response messages.

// Check Textbook for details: mobility in cellular networks, MSC, GSM, HLR, VLR, MSRN, wireless impact on higher layers.

Lecture 10 – Network Security

> ***Identification*** *occurs when a subject claims an* ***identity*** *(such as with a username).* ***Authentication*** *occurs when a subject proves their* ***identity*** *(such as with a password). Once the subject proves identity,* ***authorization*** *techniques can grant or block access to resources based on their proven identities.*

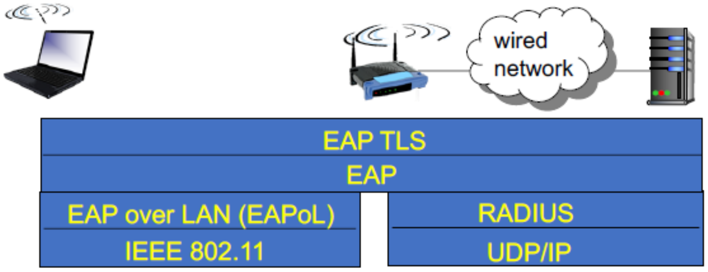
- **General attacks**
  - **Sniffing (eavesdropping or snooping):** listening to an ongoing conversation by intercepting (stopping and reading) messages.
  - **Spoofing (impersonation):** faking source address in packets.
  - **Hijacking:** taking-over a connection by replacing sender or receiver with himself.
  - **Denial-of-Service:** preventing a service from being used by others (e.g., by overloading resources) – the hardest to detect and mitigate.
- **Cryptography:** the study of techniques for secure communications in the presence of third parties.
  - **Confidentiality:** only sender and intended receiver should understand message content.
    - Sender encrypts message, receiver decrypts it.
  - **Message integrity:** message do not get altered (in transit or afterwards) without detection.
  - **Alice and Bob:** placeholders (people, computers, programs, services, …) for discussions about cryptography.
    - **Typical scenario:**
      - Alice and Bob want to communicate securely over an insecure channel.
      - Trudy (intruder) may insert, read, modify, or delete message into/from the connection.
  - **Encryption:**
    - Converting a message $\boldsymbol{m}$ from its original format (**plaintext**) into a **ciphertext** using an encryption key K_e, aiming to protect m from being understood by anyone (having access to the ciphertext) except the intended receiver, who should be the only one having the decryption key K_d to decrypt the message and retrieve the message.
      - We denote the encrypted message by $K_e(m)$, the property $K_d\big(K_e(m)\big) = m$ should hold.
      - **Symmetric-key encryption:** the same symmetric key is used for encryption and decryption.
        - How do parties agree on the key? **Diffie-Hellman** public key exchange algorithm.
      - **Asymmetric-key encryption:** public key can encrypt messages; only private key can decrypt them (private key should not be shared with anyone at all) [How]
        - Private key is denoted with a (-) superscript, public: (+)
        - **RSA** uses an asymmetric key to share the symmetric key, then both parties can continue to communicate securely.
    - **Breaking encryption** (typical methods: brute force and statistical analysis)
      - **Ciphertext-only attack -** given only a ciphertext.
      - **Known-plaintext attack -** given a (or many) plaintext and its corresponding ciphertext.
      - **Chosen-plaintext attack -** given an encryptor that yields ciphertext given any plaintext.
    - **Standards:**
      - **Data Encryption Standard (DES):** standard symmetric key algorithm used for encrypting a 64-bit input using a 56-bit key in 16 rounds of function application, each using a different 48bits of the key.
        - Was broken using brute-force in less that a day, not used today.
      - **Advanced Encryption Standard (AES):** a new standard symmetric key algorithm replacing DES, it uses a key of 128, 192, or 256-bit to encrypt a 128-bit input.
        - Immune to brute-force attacks.
    - **Block vs stream cipher:**
      - **Block cipher** encrypts a fixed size input to produce a fixed size output.
      - **Stream cipher** encrypts input bit-by-bit given a key stream.
  - **Digital signature:** cryptographic concept analogous to paper signatures, ensures authentication and data integrity.
    - Bob sends a message and the signature $m, K_B^-(m)$ to Alice.

  - Alice can prove that Bob is the author of the message and the message didn't get modified by verifying that $K_B^+\left(K_B^-(m)\right) = m$
- **Message digest (fingerprint):** cryptographic **H**ash function that is used mainly to convert a long message into a fixed-length string.
  - **Properties**
    - Same message yields same hash (operation should also be fast)
    - Small change in message should result in a huge change in hash.
    - Operation is not invertible (many to one), finding an x given h such that $h = \boldsymbol{H}(x)$ should be computationally expensive.
  - **Applications:**
    - **Integrity verification**
      - Internet checksum: produces 16-bit sum of message but it's a poor hash function since it's easy to find a message with the same hash value.
    - **Signature generation and verification.**
      - It's infeasible to compute the key encryption of a large message, instead we encrypt $H(m)$ and the rest of process is the same (Alice verifies $K_B^+\left(K_B^-(H(m))\right) = H(m)$)
  - **Examples**
    - Message Digest algorithms (e.g., MD5)
    - Secure Hash Algorithms (e.g., SHA-1)
- **Certification Authorities (CA)**
  - The receiver needs to be sure that the public key received is the one belonging to the sender, and not someone else (to detect impersonation attacks)
  - CA binds public key to a particular entity (person, router, website) certifying that this public key belongs to this entity.
  - **How it works:**
    - Alice wants to get Bob's public key; she gets his certificate $C_B = K_{CA}^-(K_B^+)$ (from him or anywhere else).
    - Applying the CA public key on the certificate, Alice can verify $K_B^+ = K_{CA}^+(C_B)$
    - The only way public key can be fake is if the intruder has the CA private key.

- **To summarize (cryptography):**
  - **Confidentiality (no eavesdropping):**
    - Can be achieved by sharing an encrypted symmetric key between parties (public key cryptography (e.g., RSA))
      - **Alice:** send $K_B^+(K_s), K_s(m)$
      - **Bob:** get $K_s = K_B^-\left(K_B^+(K_s)\right), m = K_s\left(K_s(m)\right)$
      - They use only the symmetric key for future messages (as it's faster and more efficient).
  - **Sender authentication (no impersonation) and message integrity (message don't get modified):**
    - Can be achieved by digital signatures.
      - **Alice:** send $K_A^-\left(H(m)\right), m$
      - **Bob:** verify $K_A^+\left(K_A^-\left(H(m)\right)\right) = H(m)$
    - **Receiver authentication** can be achieved in the same was as sender authentication, using K_B
  - **Confidentiality, sender authentication and message integrity all at the same time**
    - Can be achieved by combining the previous two processes (use three keys, K_A, K_B, K_s)
      - **Alice:** packetize P $= K_A^-\left(H(m)\right), m$ then use the same confidentiality process but on P:
        - **Alice:** send $K_B^+(K_s), K_s(P)$
        - **Bob:** get $K_s = K_B^-\left(K_B^+(K_s)\right), P = K_s\left(K_s(P)\right)$
      - **Bob:** extracts $P = K_A^-\left(H(m)\right), m$ and verifies $K_A^+\left(K_A^-\left(H(m)\right)\right) = H(m)$
    - This was implemented in an encryption program called **Pretty Good Privacy (PGP).**
  - We assumed that both parties shared their public keys and verified them, this can be achieved using CAs.

- **Secure Sockets Layer (SSL)**
  - Security protocol that provides confidentiality, authentication, and data integrity over TCP.
    - SSL version 3+ are now called **Transport Layer Security (TLS).**
    - Widely used by web browsers and webservers
    - Primarily used for securing transactions that run over **HTTPS (port 443)**
    - Provides API to applications (it works as a sublayer between application and transport layers).
  - **SSL cipher suite:** set of algorithms used by SSL to secure the connection, both parties agree (negotiate) on cipher suite during SSL handshake (client offers choices, server choses one).
    - Public key cryptography algorithm (e.g., RSA with a specific key length).
    - Symmetric encryption algorithm (e.g., DES, 3DES, RC2: block, RC4: stream)
      - **RC:** Rivest Cipher.
    - Message Authentication Code (MAC) key generation algorithm.
  - **How it works:**
    - **Handshake:** normal TCP 3-way handshake followed by **SSL handshake**
      - Client sends hello message (a request for connection establishment)
        - **Hello message contents**: client nonce, list of SSL cipher suite supported algorithms.
      - Server replies with its certificate, server nonce, and cipher suite choice.
        - Client verifies the certificate is valid.
      - Client sends an Encrypted Master Key (EMS) to server
        - Encrypted with server public key extracted from the certificate.
    - **Key derivation:** both parties use EMS and exchanged nonces to derive a set of keys for data encryption and verification.
      - After key derivation: client and server exchange (encrypted) MAC keys for all handshake messages to ensure no handshake data was tampered.
      - Nonces are used to prevent replay attacks (same nonce cannot be used)
    - **Data transfer:** break the TCP data stream into series of records (for data integrity) and transfer them.
    - **Connection closure:** special message to close the connection.
  - **Advantages over PGP approach**
    - Supports stream encryption of data (PGP relies on block cryptography).
    - Certificate exchange between parties during the handshake phase.
    - Uses a set of (derived) secret keys (PGP uses one symmetric key for the entire session).
- **IPSec:**
  - IP security protocol suite for encrypting IP packets' payload.
    - Packet's payload may contain TCP/UDP segment, ICMP message, OSPF message, etc.
  - **Virtual Private Network (VPN)**
    - Allows companies to have a private internal network without infrastructure overhead, by sending their IPSec datagrams over the public internet.
    - IPSec datagrams are normal IP datagrams with IPSec header appended to encrypted payload.
  - **Services**
    - (1) Data integrity
    - (2) Confidentiality
    - (3) Source authentication (replay attack prevention)
  - **Main protocols**
    - **Authentication Header (AH):** provides (1), (3) but not (2)
    - **Encapsulation Security Protocol (ESP):** provides (1), (2), (3) – more widely used.
  - **Operation modes**
    - **Transport mode**: only the payload is encrypted/authenticated.
      - IPSec datagrams are produced and received by hosts.
    - **Tunnel mode:** the entire IP packet is encapsulated as an encrypted payload to IPSec packet.
  - **Security Association (SA):** unlike IP, IPSec is connection-oriented; before sending data, the sender (one-way, simplex) establishes security parameters with the receiver by sending SAs parameters (encryption algorithm, encryption key, etc.), which are saved by the receiver in a **S**ecurity **A**ssociation **D**atabase (☹) to be able to decrypt subsequent datagrams.

- **Wireless LAN security**
  - **Wired Equivalent Privacy (WEP)**
    - Now superseded by Wireless Protected Access (WPA)
    - Efficient algorithm used RC4 to encrypt wireless packets separately (self-synchronized, packets don't have to be in order to be decrypted).
  - **Extensible Authentication Protocol (EAP)**
    - A protocol that works between a mobile client and an authentication server through an access point (that is wired connected to the server).
      - **RADIUS:** Remote Authentication Dial-In User Service



- **Firewall**
  - Isolates organization's internal network from public Internet, allowing some packets to pass, blocking (dropping) others.
  - **Usage:**
    - Prevent Denial-of-Service attacks (e.g., SYN flooding by sending many SYN requests to overload the server resources)
    - Prevent illegal modification/access of internal data.
    - Allow only authorized access to the network via Access Control Lists (ACLs)
      - **ACL:** table of rules (action, condition) applied top-to-bottom to incoming packets.
        - Conditions typically apply IP or port number restrictions.
  - **Types:**
    - **Packet filters:** examines all incoming and outgoing traffic packets, check against certain conditions (**firewall settings**) to satisfy **firewall policy.**
      - **Stateful:** maintains information (connection status) about previously blocked/allowed packets to help block **suspicious traffic**.
        - **Examples:**
          - TCP ACK with no previous SYNACK
          - Timed-out inactive connection (possible SYN flood attack).
      - **Stateless:** doesn't maintain any information.
    - **Application gateways:**
      - Application-specific server through which all application data must pass for checking.
- **Intrusion Detection System (IDS)**
  - A device/system that generates alerts when observing a malicious traffic, provides an extra layer of security, usually after a firewall (Intrusion Prevention System (IPS): blocks malicious traffic)
    - **Deep-packet inspection**: checking packet contents (e.g., against knows attack strings).
    - **Correlating packets**: identifying relations between packets to detect advanced attacks.
      - **Port scanning attack:** sending multiple packets with different port numbers to discover services provided by victim's server.
      - **Network mapping:** attackers trying to analyze/understand the physical connections and topology of victim's network, to identify the servers containing valuable information.
      - **Denial-of-Service.**
  - Multiple IDS provides different types of services (checks) at different network zones.
    - **Demilitarized Zone (DMZ):** physical/logical subnet facing the public internet, provides an extra layer of security by having nodes communicate to the internal LAN only through them (usually contains organization's DNS, FTP, and/or HTTP servers), and the rest of LAN being firewalled.