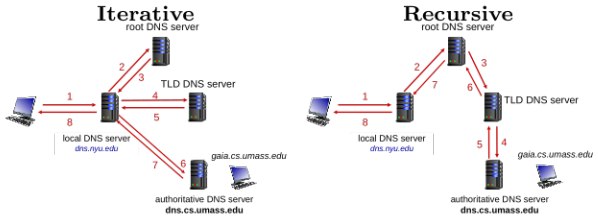
HTTP - Application Protocol

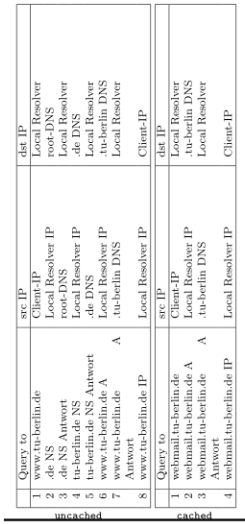
`non-persistent`	one object per TCP connection
`persistent`	multiple connection per TCP connection
`pipelined persistent`	multiple requests, without blocking for responses, responses in order of requests
`persistent out-of-order`	multiple requests, responses can be in different order

DNS - Application Protocol

- **NS Record (Name Server Record)** Specifies the authoritative name servers for a domain. In this case, the authoritative name servers for `tu-berlin.de` are
- **Authority Section** Lists the authoritative name servers for the queried domain. These servers are responsible for providing the official DNS records. The TTL (Time To Live) for these records is 28,800 seconds.
- **Additional Section** Provides extra information, such as the IP addresses of the authoritative name servers, to speed up resolution.

- **Recursive DNS:** The local DNS resolver does all the work and returns the final answer to the client.
- **Iterative DNS:** The resolver receives a referral and queries the next DNS server (root, TLD, authoritative) in the hierarchy.
- **Root DNS Servers:** There are 13 root servers worldwide, distributed via Anycast, responding from the closest instance.
- **Local DNS Queries:** Queries from clients to local resolvers are typically recursive.
- **Root and TLD Server Queries:** Queries to root and TLD servers are iterative, with referrals to the next server.
- **Authoritative Server Queries:** Queries to authoritative servers are also iterative, with the final answer provided (e.g., A record).





local DNS-Server	everyone has one
Authoritative Name-Server	accountbale for a specific host
Root Name-Server	13 globally, per anycast

`A`	Hostname, IPv4
`NS`	for Authoritative Name-Server
`CNAME`	for aliases (CDNs)
`MX`	Mail
`AAAA`	IPv6

IP - Internet Protocol

IP Address: Identifier for a host/router interface.

- **IPv4**: 32-bit decimal
- **IPv6**: 128-bit hexadecimal

Key Concepts

- **Interface**: Connects a host/router to a physical link.
- **Network**: Physical reachability without routers.
- **Link-local addresses**: Non-routable (e.g., `169.254.0.1/16`).
- **CIDR**: Divides network and host portions (e.g., `a.b.c.d/x`).

Special Addresses

- IPv4: `127.0.0.0/8` (Loopback), `224.0.0.0/4` (Multicast), `10.0.0.0/8`, `192.168.0.0/16` (Private).
- IPv6: `::1/128` (Loopback), `2000::/3` (Global Unicast), `FC00::/7` (Private).

Protocols

- **DHCP**: Dynamic IP assignment.
- **IPv6 SLAAC**: Stateless address configuration (router prefix, host ID).
- **NAT**:
  - One `IP` for local devices, not directly addressable.
  - `NAT Table`: Maps `WAN` and `LAN` addresses/ports.

`NAT Table`

External Address	External Port	Internal Address	Internal Port

CIDR	Subnet mask	# of IP addresses	# of usable IP addresses
/32	255.255.255.255	1	1
/31	255.255.255.254	2	2*
/30	255.255.255.252	4	2
/29	255.255.255.248	8	6
/28	255.255.255.240	16	14
/27	255.255.255.224	32	30
/26	255.255.255.192	64	62
/25	255.255.255.128	128	126
/24	255.255.255.0	256	254
/23	255.255.254.0	512	510
/22	255.255.252.0	1,024	1,022
/21	255.255.248.0	2,048	2,046
/20	255.255.240.0	4,096	4,094
/19	255.255.224.0	8,192	8,190
/18	255.255.192.0	16,384	16,382
/17	255.255.128.0	32,768	32,766
/16	255.255.0.0	65,536	65,534
/15	255.254.0.0	131,072	131,070
/14	255.252.0.0	262,144	262,142
/13	255.248.0.0	524,288	524,286
/12	255.240.0.0	1,048,576	1,048,574
/11	255.224.0.0	2,097,152	2,097,150
/10	255.192.0.0	4,194,304	4,194,302
/9	255.128.0.0	8,388,608	8,388,606
/8	255.0.0.0	16,777,216	16,777,214
/7	254.0.0.0	33,554,432	33,554,430
/6	252.0.0.0	67,108,864	67,108,862
/5	248.0.0.0	134,217,728	134,217,726
/4	240.0.0.0	268,435,456	268,435,454
/3	224.0.0.0	536,870,912	536,870,910
/2	192.0.0.0	1,073,741,824	1,073,741,822
/1	128.0.0.0	2,147,483,648	2,147,483,646
/0	0.0.0.0	4,294,967,296	4,294,967,294

`*often used for transfernets, f.e. two router`

TCP - Transport Protocol

```
connection oriented, handshake before transmission
point to point, reliable datastream
flow control and congestion control
mss - maximum segment size
```

`Sequencenumber`	Byte Stream Nummer of the first 1 byte of the segmentdata
`packet loss detect`	retransmission timeout or 3 double ACKs

Retransmission timeout - $T_{interval}$

$RTT_{est,0} = RTT_{sample,0}$

$$RTT_{est} = (1-\alpha) \cdot RTT_{est} + \alpha RTT_{sample},\ \alpha = 0.125$$

$$RTT_{dev} = (1-\beta) \cdot RTT_{dev} + \beta(|RTT_{sample} - RTT_{est}|),\ \beta = 0.25$$

$$T_{interval} = RTT_{est} + 4 \cdot RTT_{dev}$$

Retransmitted Packets do not count into the RTT because, TCP does not not know if the ACK is from the original packet or the retransmitted.

Flow Control - FC
not overflow receiver's buffer
receiver informs sender about free buffer
sliding windows with size n, n bytes are sent
ideal $w_{\text{size}} = \text{delay} \cdot \text{bandwidth} = \text{RTT} \cdot \text{bitrate}_{\text{bottleneck}}$

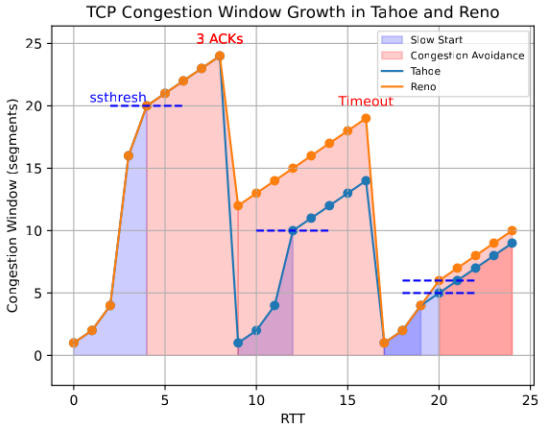
Congestion Control - CC
avoidance of lost packets (buffer overflow in router)
avoidance of long delays (queuing delays in router)

Slowstart - SS
exponentially increase: 2^{cwnd}

Congestion Avoidance - CA
linear increase: $cwnd++$

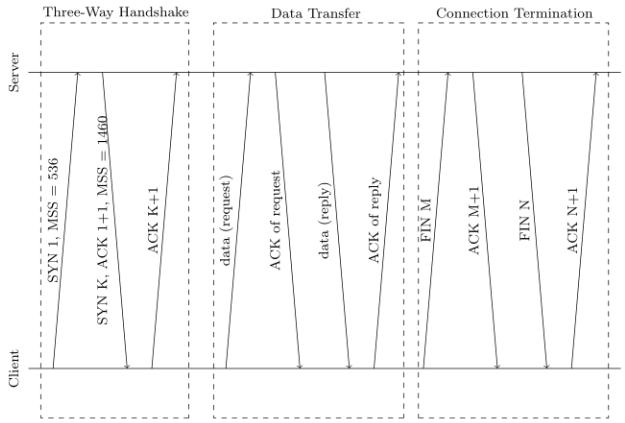
	Timeout	**3 additional Dup ACKs**
Tahoe	$cwnd_n = 1$ $ssthresh = \frac{cwnd_o}{2}$ and SS	$cwnd_n = 1$ $ssthresh = \frac{cwnd_o}{2}$ and SS
Reno	$cwnd_n = 1$ $ssthresh = \frac{cwnd_o}{2}$ and SS	$cwnd_n = \frac{cwnd_o}{2}$ $ssthresh = cwnd_n$ and CA

Both start with SS until $cwnd = ssthresh$ and then proceed doing CA until **Timeout** or **3 additional Dup ACKs** happen. if **3 additional Dup acks** are detected, the transmission round is is instantly stopped



$$Throughput_{mean} = \frac{3}{4} \cdot \frac{cwnd[bytes]}{RTT[s]} [bytes/sec]$$

$cwnd$ where first loss occurs



UDP - Transport Protocol

besto effort
connection-less
can get lost, can arrive in different order

Link Layer Data Link Layer:

- Error detection and correction.
- Broadcast channel: Multiple access.
- Link layer addressing.
- Half-duplex and full-duplex.

MAC Protocols:

- Channel partitioning (e.g., time, frequency).
- Random access: Handle collisions.
- Taking turns: Coordinate access to avoid collisions.
- Goal: Efficient, fair, simple, decentralized.

Addresses:

- IP: Network-layer addresses for routing.
- MAC: Data link-layer addresses for local communication.
- 48-bit MAC address: Embedded in NIC ROM.

Ethernet:

- Dominant LAN technology.
- Uses CSMA/CD: Collision detection and handling.
- Exponential backoff: Wait time after collision.
- Kollision handling: Random delay on successive collisions.

ARP:

1. A broadcasts an ARP query to FF:FF:FF:FF:FF:FF with B's IP.
2. B replies with its MAC address (which is send to A).
3. A updates its ARP table with the IP-to-MAC mapping and TTL.

Routing

adressing, path determination with routing algos, switchting, forwarding

- Global: Full topology and link cost knowledge. (Link-State)
- Decentralized: Knows only direct neighbors, iterative updates. (Distance-Vector)
- Static: Routes change slowly over time.
- Dynamic: Routes update frequently via periodic updates or link cost changes.

Intra-AS Routing: Routing within an autonomous system.

- IGP: Interior Gateway Protocols
- RIP: Routing Information Protocol
- OSPF: Open Shortest Path First (Link-State Algorithm)
- State: Each router knows its reachable networks.
- OSPF Advertisements: Spread state info, database sync.
- Link-State Database: Stores all router states.
- Hello Protocol: Finds neighbors.
- Hierarchical OSPF: Used in large networks.
- Router Hierarchy:
  - 1 Boundary Router
  - n Backbone Routers
  - m Area Border Routers
  - Internal Routers

Inter-AS Routing: Routing between autonomous systems.

- BGP: Border Gateway Protocol (de facto standard)
- BGP Sessions: BGP peers exchange routing info over semi-permanent TCP connections.
- Path-Vector Protocol: BGP tracks routes using AS paths.
- AS-PATH: Lists all ASs in a route.
- NEXT-HOP: Internal AS router to the next-hop AS.
- Route: Defined by prefix + attributes.
- Ingress Filtering: Gateway router validates/filters route proposals (e.g., to prevent loops).
- Routing Policy: Aligned with provider goals (cost efficiency).
- Local Preference: Routes with the highest preference are favored.
- **Route Selection Criteria**:
  - Highest Local Preference
  - Shortest AS-PATH
  - Best MED (Multi-Exit Discriminator)
  - Nearest NEXT-HOP (Hot Potato Routing)
  - Peer's IP Address

	Own Routes	Customers Routes	Siblings Routes	Providers Route	Peers Route
Advertising to Provider	yes	yes	yes		
Advertising to Customer	yes	yes	yes	yes	yes
Advertising to Peer	yes	yes	yes		

Hard State vs. Soft State

- **Hard State**:
  - Installed during connection setup, removed upon termination (e.g., TCP, SS7).
  - Assumption: State remains valid until explicitly changed.
  - Reliable signaling with explicit state deletion.
  - Inconsistency and overhead decrease with state lifetime.
  - Explicit removal improves consistency with minimal overhead.
- **Soft State**:
  - Installed during connection setup, maintained via refresh messages, removed by timeout or lack of refresh (e.g., RSVP, IGMPv1).
  - Assumption: State becomes invalid if not refreshed.
  - Uses a refresh timer at sender and a timeout timer at receiver.
  - Easier fault detection, only implicit reliability.

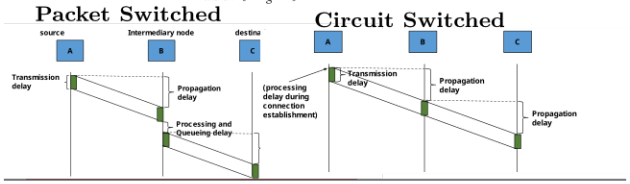
Internet Signaling

- **In-Band Signaling**:
  - `HTTP`: Messages combine signaling and data.
  - `SIP (Session Initiation Protocol)`: Uses the same channel for control and media setup.
  - `Telnet`: Commands and data are sent over the same connection.
- **Out-of-Band Signaling**:
  - `FTP`: Uses a dedicated control connection.
  - `H.323`: Control and media streams use separate channels.
  - `SS7`: Uses a separate network for signaling in telecom systems.

`packet switched`	logical connection, via multi-path
`circuit switched`	static connection for the whole communication, resources are allocated for the whole connection lifespan

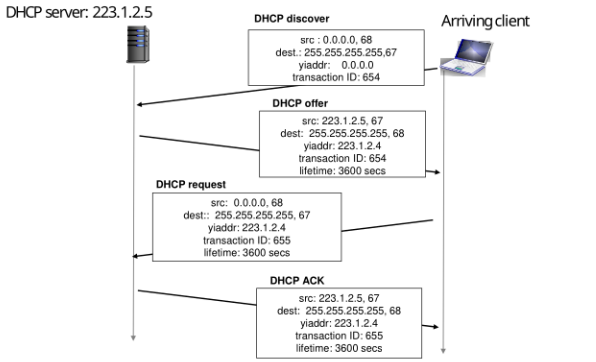
`Number of Users for CS:`

$$\#user = \frac{Bandwidth_{all}[\frac{mbit}{s}]}{Bandwdith_{user}[\frac{mbit}{s}]}$$



Types of Network Delays:

- `Transmission Delay`: Time required to push all bits of a packet onto the wire.
- `Propagation Delay`: Time for a bit to travel through the link, depending on the physical medium.
- `Processing Delay`: Time needed to inspect the packet header and determine its forwarding path.
- `Queuing Delay`: Time a packet spends waiting in the queue before transmission, influenced by queue length.

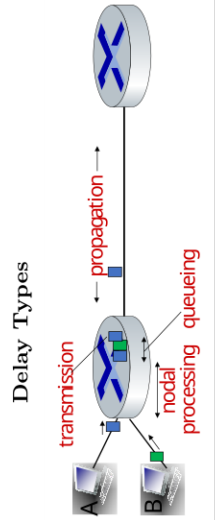


`Forwarding Table`

Prefix	Next Hop	Interface
192.168.1.0/24	10.0.0.1	eth0
10.0.0.0/8	192.168.1.1	eth1
172.16.0.0/16	10.0.0.2	eth2

`if all network caches are empty:`

1. `ARP`
2. `DNS`
3. `IP`
4. `TCP/UDP`



d_{proc}: nodal processing
- check bit errors
- determine output link
- typically ¡msec

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

d_{trans}: transmission delay
- L packet lengths [bits]
- R link transmission rate [bps]

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

d_{prop}: propagation delay
- d length of physical link $[m]$
- s propagation speed $2 \cdot 10^8 [m/s]$

$d_{pop} = d/s$

$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$

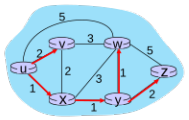
Prefix	Symbol	Power of 10
Tera	T	10^{12}
Giga	G	10^9
Mega	M	10^6
Kilo	k	10^3
Hecto	h	10^2
Deca	da	10^1
Base Unit	-	10^0
Deci	d	10^{-1}
Centi	c	10^{-2}
Milli	m	10^{-3}
Micro	μ	10^{-6}
Nano	n	10^{-9}

Switches break collision domain but not broadcast domains. Router break collision domain and broadcast domain. Hubs break nothing.

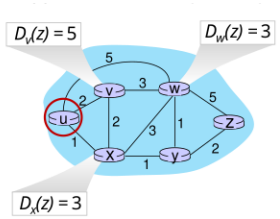
0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F

Bytes $= \frac{\text{Bits}}{8}$, Bits $=$ Bytes $\cdot 8$

Step	N'	v D(v),p(v)	w D(w),p(w)	x D(x),p(x)	y D(y),p(y)	z D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



8 *Loop*
9
10
11



Bellman Ford equation says:

$D_u(z) = \min \{ c_{u,v} + D_v(z), c_{u,x} + D_x(z), c_{u,w} + D_w(z) \}$
$= \min \{2 + 5, 1 + 3, 5 + 3\} = 4$

node achieving minimum (x) is next hop on estimated least cost path to destination (z)

1. **Omit leading zeros:** Remove leading zeros in each 16-bit segment.
2. **Use double colons (::) for consecutive zeros:** Replace the longest contiguous sequence of all-zero segments with `::`. This can be done only once in an address.
3. **Do not shorten single zeros unnecessarily:** If there are multiple zero sequences of the same length, replace only the first one to avoid ambiguity.

Example

Consider the following IPv6 address:

`2001:0db8:0000:0000:0000:ff00:0042:8329`

Applying the rules:

1. Remove leading zeros in each segment:

   `2001:db8:0:0:0:ff00:42:8329`

2. Replace the longest sequence of consecutive zeros with `::`:

   `2001:db8::ff00:42:8329`

Thus, the shortened IPv6 address is:

`2001:db8::ff00:42:8329`
