IK1203 Networks and Communication

Chapter 1

Nuts & Bolts

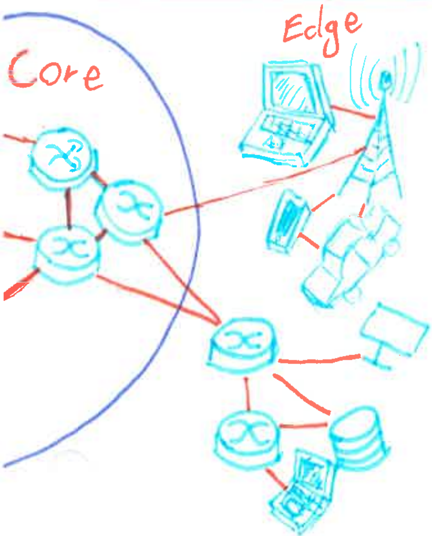
Hosts - end system
Bandwidth - transmission rate
Router & switches - forward packets
Packet - chunk of data
Protocol - controlled sending of data
Standards:
RFC - Request for comment
IETF - Internet engineering taskforce

Service View.

Two parts:
Infrastructure:
- Web
- VoIP
- Email
- games
- streams
- Social networks

Programming interface:
- Applications that "connect" to the internet.
- Analogus to postal service.

Network structure



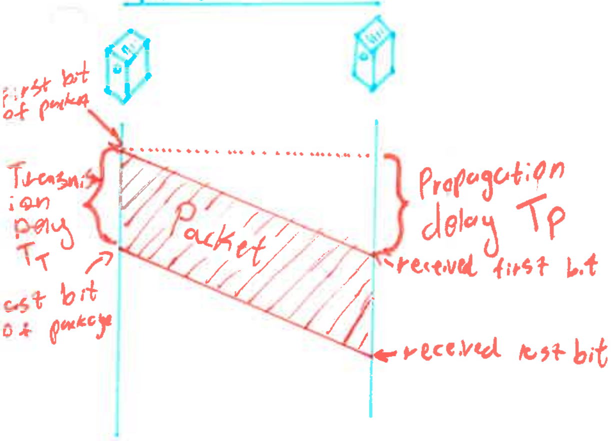
Core: internet connection routers
Networks of networks.

Edge: Clients, hosts and servers.
Datacenters

Speeds

xDsl (digital subscriber line)
↳ existing phone line
↳ 1-100 Mbps
Fible (optical)
↳ 10-100000 Mbps
DOCSIS (Data over cable service internet specification)
↳ 1-200 Mbps
wireless (broadband)
↳ 3G 4G 5G LTE
↳ 1-1000 Gbps

Transmission



Total time

$$T = T_p + T_t = \frac{d}{s} + \frac{L}{r}$$

d = distance (m)
s = propagation speed (ms^{-1})
L = packet size (bit)
r = Link rate bit s^{-1}

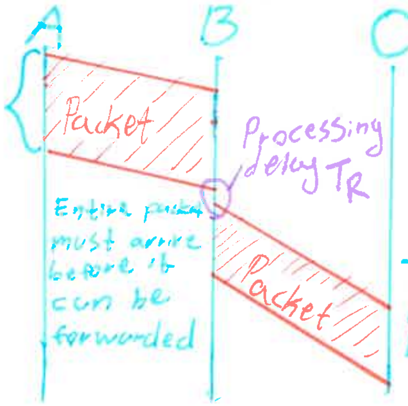
The core

Mesh of interconnected routers. Packets are sent between links to it's destination.

Routing - determines the source to destination route.

Forwarding - Moving packets from router input to apropriate output.

Packet switching



Processing delay:
The time needed to:
- Check and verify packet
- Decide what to do with it

Time to send packet A-C assuming same speed at both links.

$$T = 2(T_T + T_P) + T_R$$

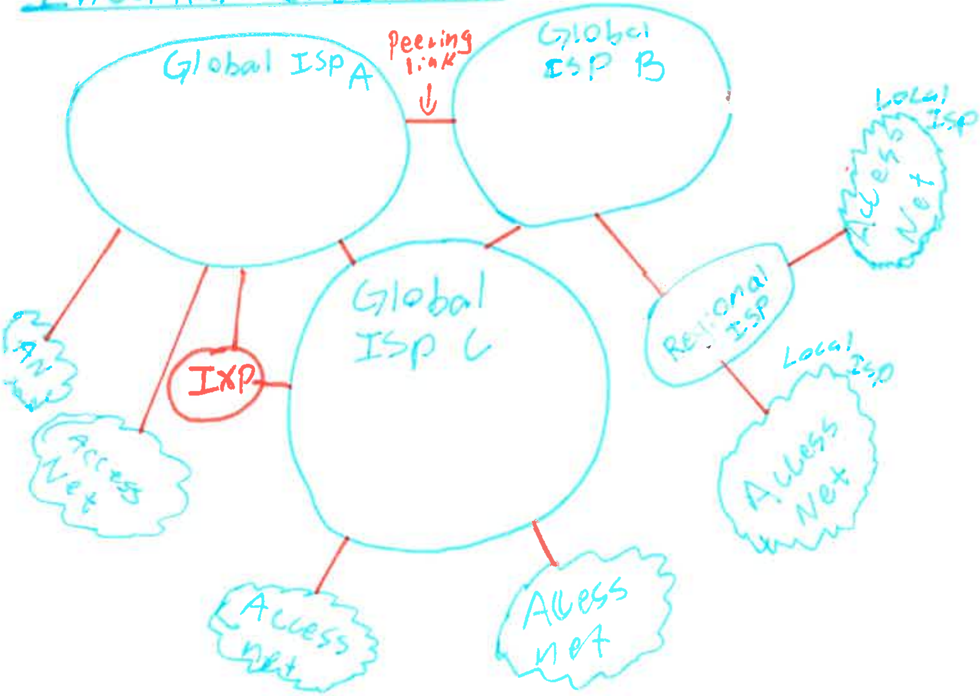
Queuing Delay & packet loss

Only one packet can be sent at a time on a link. other packets have to wait in a queue this introduces more delay known as T_Q. If queue is full packets are lost.

Internet protocol stack

Layer	
Application	- Network applications, HTTP, FTP
Transport	- Post Process Data transfer TCP, UDP
Network	- Routing from S. to D. IP
Link	- Transfer between neighbours, wifi ethernet
Physical	- bits in a "wire"

Internal Structure



Global ISP: AT&T, NTT, Telia...

Application Layers

The things you use the Internet for:

Mail, web, File sharing, IP telephony, Directory service

Network Layer

Delivers between hosts, Uses Header of IP datagrams to find out where to send packets.

Physical layer

The way the bits are transferred between transmitter & receiver pairs.

Physical link: The medium of the transport

Guided Media: Solid medium such as copper, fiber, coax

Unguided media: freely transported signal, eg Radio.

Twisted Pairs (TP): two insulated copperwires.

Cat-5- 100 Mbps, 1 Gbps Etrenet

Cat-6-10 Gbps

Cat-7- 40 Gbps

Cat-8-100 Gbps

Transport layer

Creates communication between applications on different hosts. "End to end" communication. Uses TCP and UDP,

TCP- Transmission control protocol

↳ reliable stream of data (ordered & confirmed)

UDP- User Datagram Protocol

↳ delivery of individual datagram (packets)

Link Layer

Transfers data between physically adjacent nodes over a link

A Link can be:

- IEEE 802.11- WLan
- Ethernet
- Bluetooth
- IEEE 802.15- W PAN
- 3G, 4G, 5G

Chapter 2 - Application Layer

Client - Server Architecture

Built from Clients & Servers:

Servers:
- Always active
- Permanent IP
- Datacenters for scaling

Clients:
- Communicate with server.
- connected when needed.
- may have dynamic Ip addresses.
- Do not communicate with each other.

Examples:

WEB: Browsers (client) fetch webpages from web-servers

Mail: Mail program (client) Connect to mail server

Web mail: webbrowser (client) connect to server. The server acts as a client to fetch emails from a mail server

P2P Architecture

Built from clients (nodes)

Peers request service from other peers. Peers may change their Ip address which is complex to manage. Self scaling - new peers bring new service demand and capacity.

Processes Communicating

Processes on different machines communicate by exchanging messages

This applies to both client-server and P2P networks.

Addressing Processes:

processes on a machine can be addressed with port numbers.

Http server: 80

mail server: 25

Application layer defines:

Types of messages:

request, response etc...

Message syntax:

Types of fields, how they are defined and how many

Message semantics:

Meaning of the content of the messages.

Protocols:

open:

HTTP, SMTP

Proprietary:

Skype,

Rules:

When and how processes send and respond to messages.

What transport service does an application need?

Data integrity:

Some applications can handle some data loss eg. Skype, twitch.

but others require 100% accurate data such as file transfers.

Throughput:

Some applications need high throughput such as Netflix, zoom.

While others are elastic such as file downloads and email.

Timing:

Some apps such as video games need low latency.

Security:

Encryption, data integrity.

Internet Transport Protocol Services

TCP:

+ Reliable transport
+ Flow control (receiver wont overload server)
+ Congestion control (throttle network when overloaded)
+ Connection-oriented (setup required)

Does not provide:

- timing
- minimum throughput garantee
- Security

UDP

- Unreliable
- no flow control
- no congestion control
- no timing
- no min throughput gurante
- no Security

TCP & Security

SSL can be introduced to provide encryption that tcp otherwise lacks.

SSL gives:

- encryption
- data integrity
- end-point authentication

Web & HTTP

Webpage consist of objects such as HTML, JPEG, Java and audio.

http://www.kth.se/home/picture.gif

- http → protocol
- www.kth.se → host name
- /home/picture.gif → Path name

HTTP

Client: browser that request receives, and display webpages

Server: webserver that sends web page using http

HTTP is sent in ASCII
HTTP is stateless.
HTTP uses port 80

HTTP Methods

HTTP/1.0:

GET:
Gets the entire website of specified URL.

POST:
Send data to server to give new information to server.

HEAD:
Gets only the HTTP header and not the entire web object.

HTTP/1.1:

PUT:
Uploads file in entity body to specified URL

DELETE:
Deletes file specified URL field.

HTTP Response

HTTP/1.1 200 OK — Status Line

Date: Sun 26 Sep....
Server: Apache....
Last-Modified: Tues 13 Sep....
ETag: "170E1-AFF-... "
Accepted-Ranges: bytes
Content-Length: 2654
Keep-alive: timeout=10, max=100
connection: keep-alive
content-Type: text/Html

} header

data data data..... — data, eg Html file

Status Codes

200 - OK
301 - moved permanently
400 - bad request
404 - not found
505 - HTTP version Not supported

Cookies

Four components:

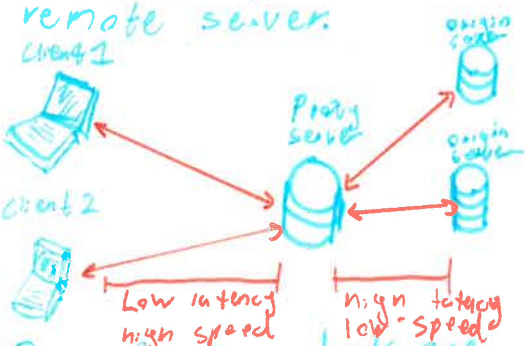
1) Cookie header line of HTTp response message.
2) Cookie header line in next HTTP request message.
3) Cookies are kept on clients computer and is managed by the users browser.
4) backend database at website server

Cookie usage

* automatic logins
* shopping carts
* user session
* keep a "state" at both ends

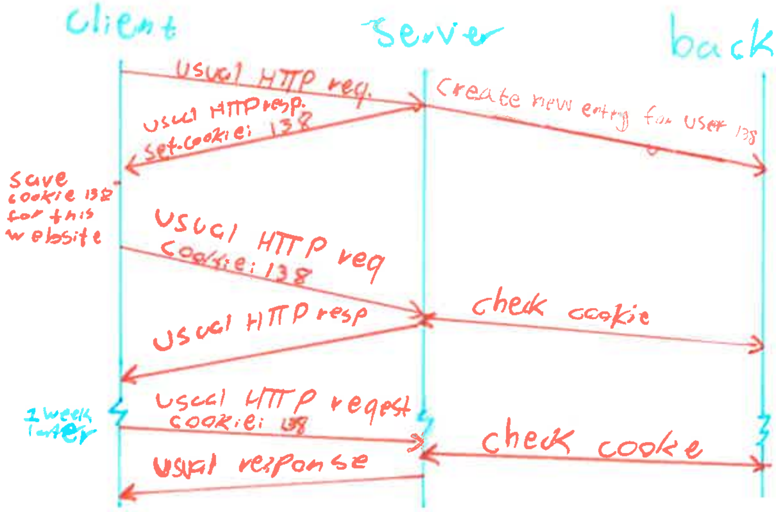
Web cashes (Proxy)

Proxys are used to minimize requests to a remote server.



Requests from clients are go through the server and if they have been stored from before they are instantly returned to the client. If not, a request is sent to the origin server and then stored on the proxy server before being returned to client.

Example



HTTP/2

What is new?

Multiplexing: loading multiple objects at the same time over single connection.

Compact header: Binary instead of text, and compressed

Backwards compatable: version negotiation

Electronic Mail (email)

3 major components:

User agent: Composes and reads emails. Client thunderbird/outlook etc...

Mail Servers: Mailbox: contains mails that came to a user
Message queue: outgoing messages waiting to be sent.

Transfer protocol (SMTP): Simple Mail Transfer Protocol
Uses TCP to reliably transfer email on port 25

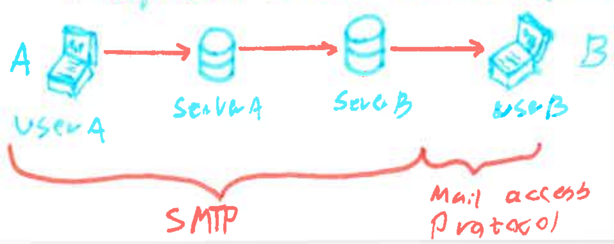
SMTP

Works with TCP on port 25
3 parts to transfer:

* Handshake
* Transfer
* Closure

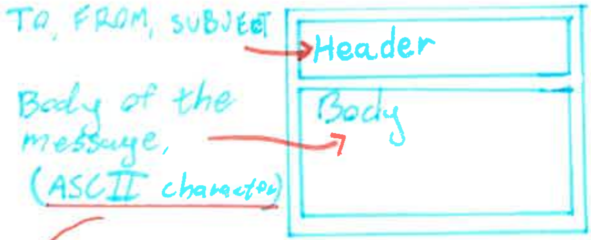
Command response like HTTP

* Commands: ASCII text
* Response: status code & phrase



Mail Message Format

Mail uses RFC 5322 format



All attachments, Images and objects are encoded into ASCII in order to be sent

Mail Access Protocol

SMPT only sends and stores the email on the recipients server.
Another protocol is needed to retrive the message from the server.

POP: Post Office Protocol (3)
Stateless
"Download and delete"

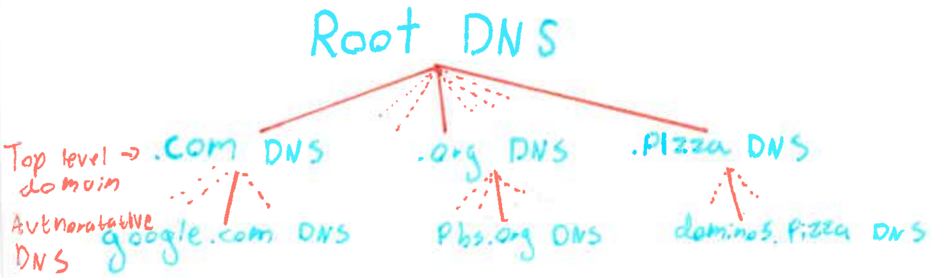
IMAP: Internet Mail Access Protocol
Keeps all messages on server
Allows for folder
Stores user state across session (stat

HTTP: gmail, Hotmail yahoo etc...

DNS (domain name system)

A way to assign websites names, such as google.com
"resolves" hostnames to IP

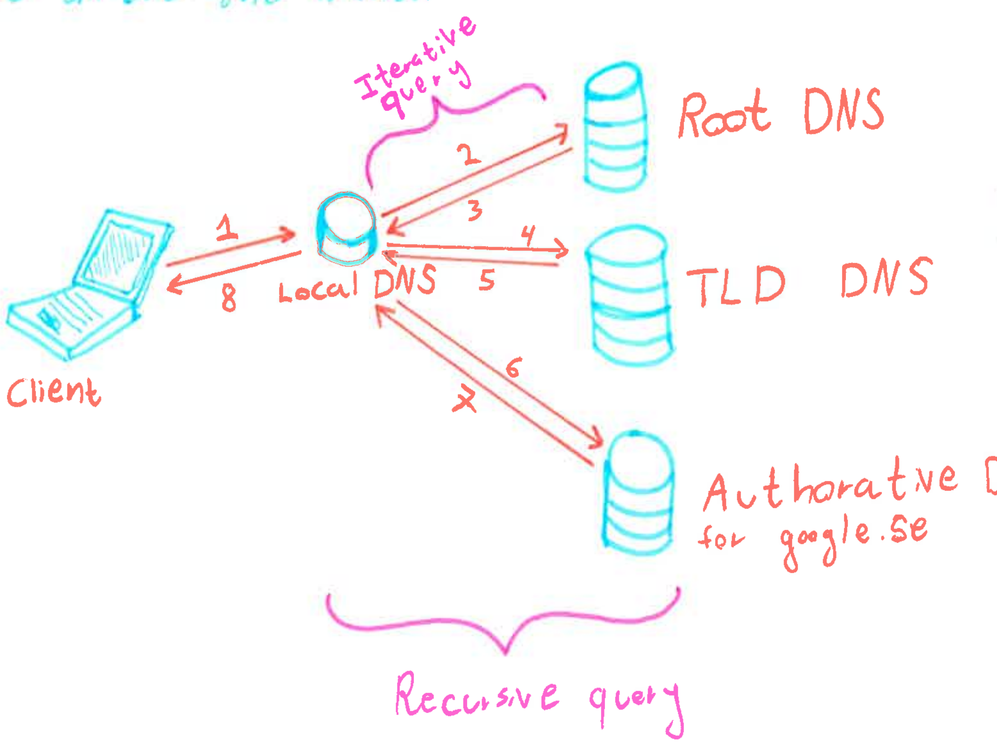
google.se ⟶ 172.217.20.35



There are 13 institutions with root-severs all over the world.

Local DNS

Does not belong in the DNS hieran
Instead acts like a proxy and store the most used requests to save time on geting responses for every request



This entire process does not happen every time, in reality a lo of these entries are already cashed on either the local DNS or the client. The root DNS is rarel visited.

DNS record

DNS uses resource records (RR) to store its entries.

RR format = (Name, value, type, ttl)
time to live

Type = A

name - hostname (maps.google.com)
Value - IP address

Type = NS

name - domain (google.com)
Value - Hostname of authoritative server for given domain.

Type = CNAME

Name - alias name for server
Value - canonical name for server

eg: ibm.com is an alias for servereast.backup2.ibm.com

Type = MX

name - name
Value - associated mailserver name

Chapter 3 - Transport Layer

Transport vs network layer

Network - logical communication between hosts.

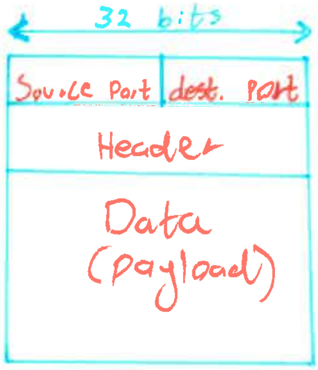
Transport layer - logical communication between processes.

The only devices with a transport layer are the two endpoints. not the routers between.

Concerned with:
reliability, congestion control, flow control, connection setup, delay guarantees, bandwidth guarantees.

Multiplexing & demultiplexing

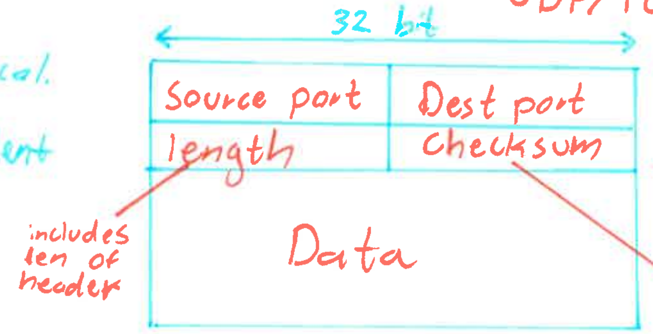
Both server and client allow for more than one connection at a time. this is done through multiplexing, which uses port numbers.



UDP/TCP format

UDP

User datagram protocol.
NO handshake
segments are independent
Reliability is added at application layer



UDP

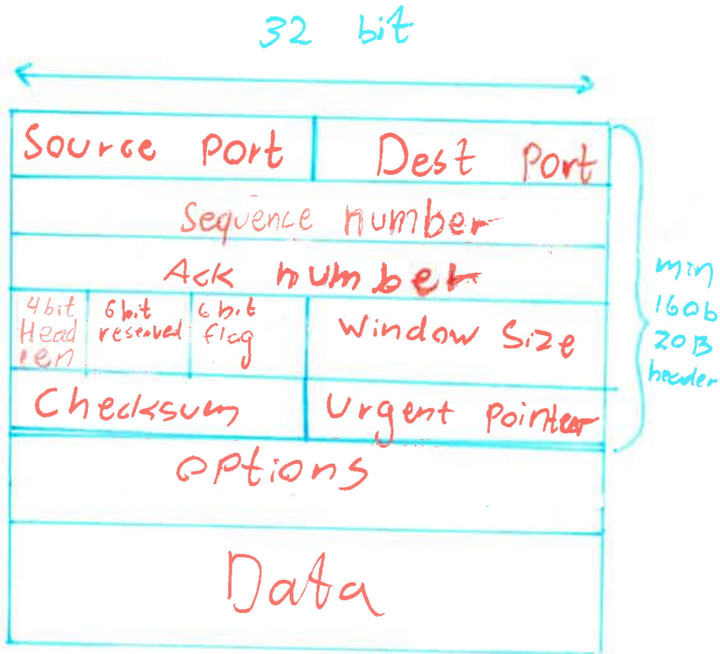
header is 64 bit/8 byte

That is an advantage over TCP

Detects errors in entire package including header

TCP

Transmission control protocol
Uses RFC 793
Connection management
Reliable, flow control, congestion control.
(Does not work for broadcast & multicast)
Decides packet size by itself



Reliability is achived by:

- Splitting data dynamically into the best size.
- Each segment maintains a timer
- Retransmit if no Ack is recieved before timeout.
- Recieved messages are acknowledged back to sender.
- Mantains a checksum for each segment
- Discard duplicate data.
- Provides flow control

Sequence num: First message is random the next ones are +1 for every message.

Header len: Used so you know how many header options are used if any. (max = 60 bytes)

Ack number: next sequence number that the receiver is ready to receive

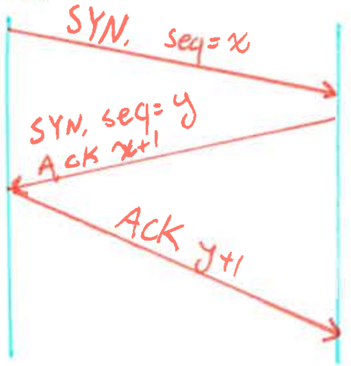
Window size: The size of the segment that the receiver is willing to accept. (bytes)

TCP Flags

ACK - Acknowledge number valid
RST - Reset connection
SYN - Synchronize sequence numbers
URG - Receive data immediatly (urgent)
PSH - Pass data to application.

Establish TCP Connection

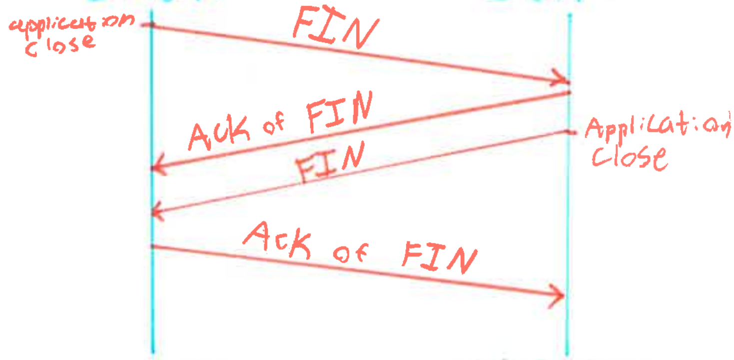
Uses a 3-way handshake



Sequence numbers x, y are unique to client & server. x, y are random to improve security & to prevent confusion with previous connections.

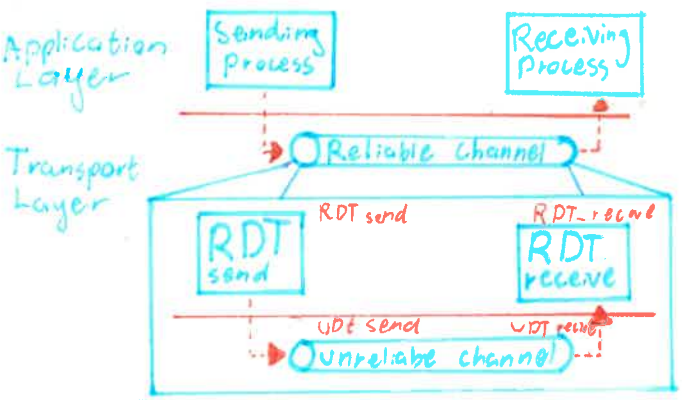
Teardown TCP connection

4 segments are needed to close a TCP connection



Normally, client is altive close and server is passive close

Reliable Data Transfer (RDT)



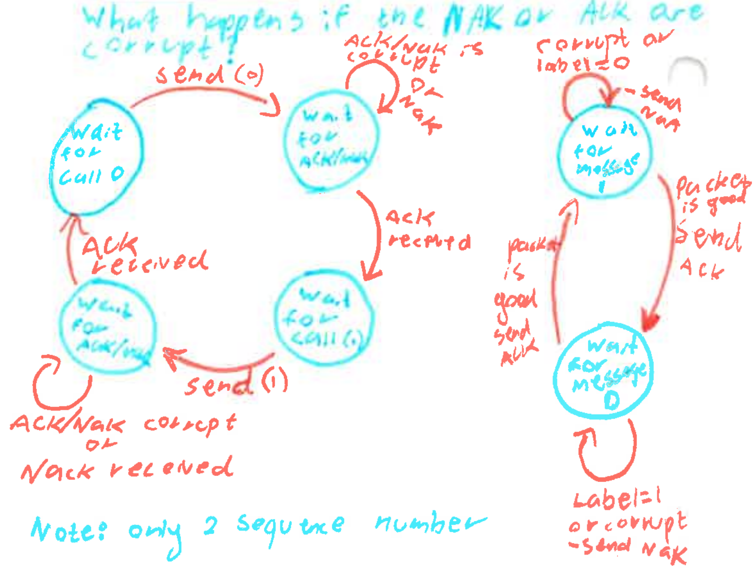
RDT 1 assumes pakets are always sent Sucsessfully. Not much more to say.

RDT 2.0

Accounts for bitflips & lost packets.
Send ACK when receiving
Send NAK when error recieve

RDT 2.1

What happens if the NAK or ACK are corrupt?

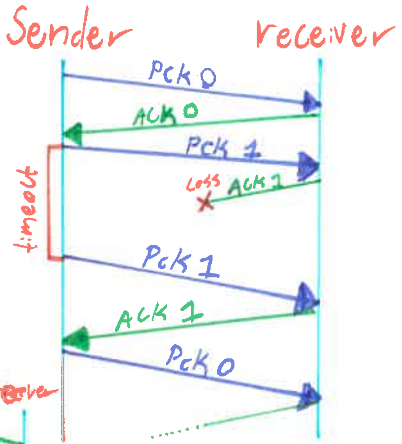


Note: only 2 sequence number

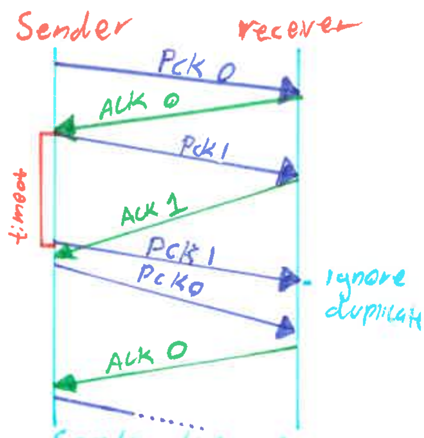
RDT 3.0

Timeout that causes data to be resent.
Acks specify sequence number.

The Graph looks like RDT 2.1 but it also resends in the event of a timeout.



The same would happen if pck 1 was lost as well.

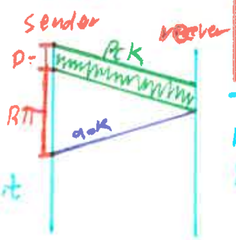


Sender does not know that pck 1 actually arrived.

RDT 3.0 works great but the performance is awful.

$$Utility = \frac{transmission}{RTT + transmission}$$

It is basically stop & wait

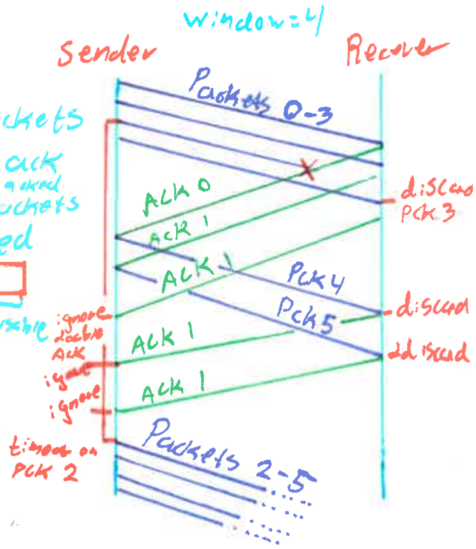


Pipelined protocols

Allow multiple packages to be sent before the first Ack is received.
Allowing Better utilization.
(go back N)

Go Back N:

- Sender sends N packets before waiting for ack
- timeout, resend all unacked packets when timeout is reached



Maximum Segment Size

- Related to TCP not RDT.
- Largest chunk of data that TCP will send. Announced in options field.
- Default value = 536
- Large MSS is good until fragmentation.

Flow control

- The amount of data sent before the sender expects an ACK.
- Balance speed and the receivers ability to process data.
- Uses sliding window protocol.
- Duplex (both sender & receiver are seperate)

When sender sends many small packages, the receiver may wait and send a comulative Ack to prevent too many messages from sending

Optimal window size:

Capacity (bits) = bandwidth (bits/sec) × RTT (sec)

window size field is 16 bit ⇒ max possible window size = 65535

Congestion

Despite what the receiver says, there can be other factors that limit the transfer-window other than the receivers read speed. If a router buffer is overflown, sending more data after a timeout would only worsen the problem, causing further congestion.

Congestion Control

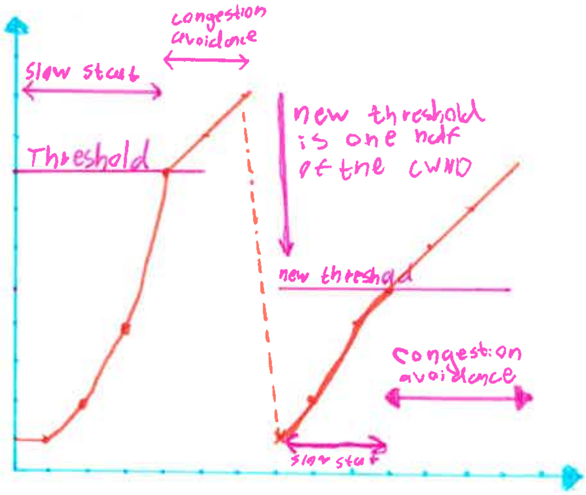
Since the receiver is not the only Agent that can determine the throughput through the window size, a congestion window (CWND) is used.

⇒ window size = min(rcv window, CWND)

CWND (slow start)(tahoe)

CWND starts smal but increases exponentially.

- At begining CWND = MSS
- for each ACK increase CWND by 2 times until CWND reaches a threshold value
- Thereafter increase by one segment for each ACK.
- Continues until congestion or CWND = WND
- If congestion occur threshold is set to ½ last CWND. & CWND restarts from MSS



Chapter 4 Network Layer

Forwarding & Routing

Forwarding:
Move packet from router input to router output.

Routing:
Determine the best route for packets to take in network

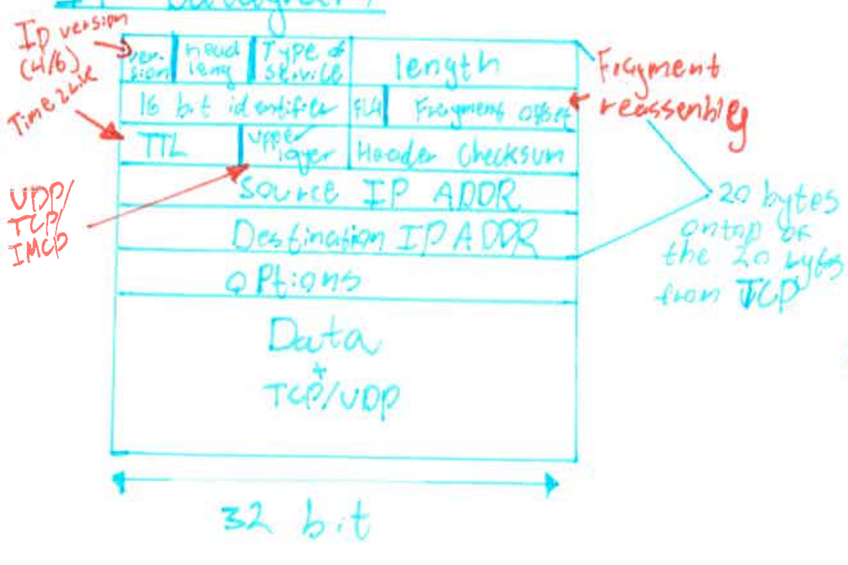
Forwarding Table

ADDRESS	Link
10011011 01 ****	0
01111011 0011 **	1
01111011 00111 *	2
Otherwise	5

Longest match possible

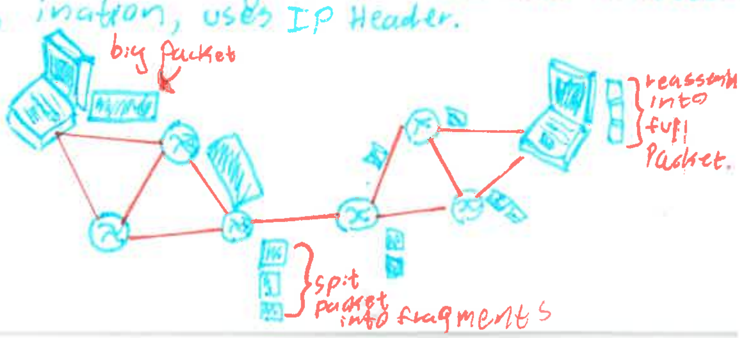
Table is created by the Routing protocol

IP Datagram



Ip Fragmentation

Some links have a Maximum Transmission Unit (MTU). Ethernet is 1500 byte. Datagrams larger than MTU are broken up into fragments, Reassembled at destination, uses IP Header.

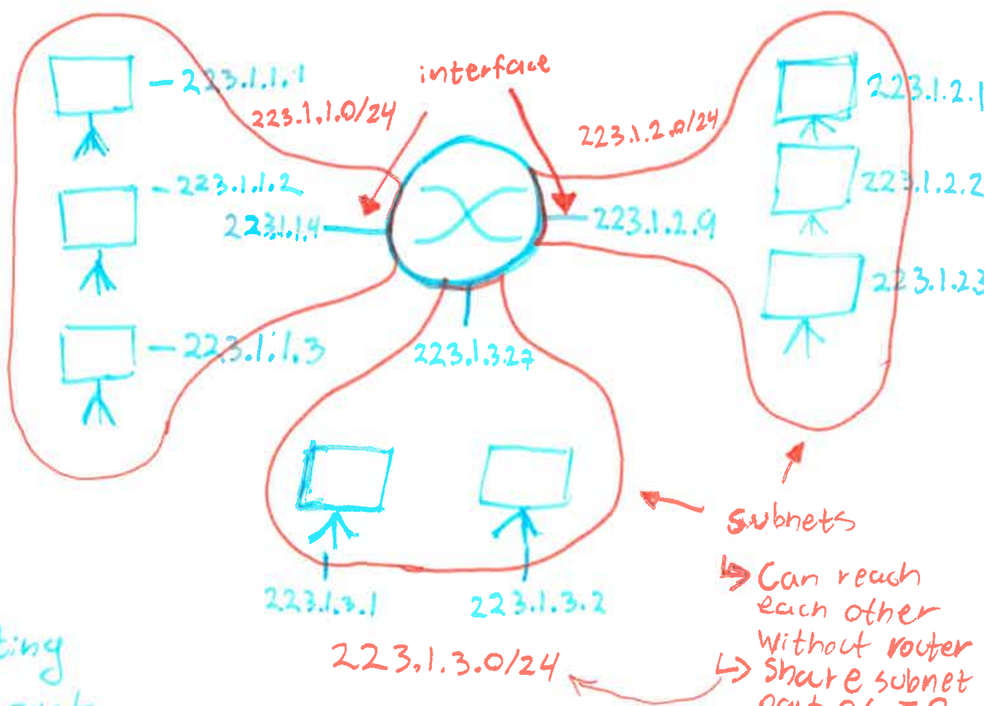


IP Address (version 4)

32 bit identifier
↳ 4.3 Billion addresses
(not enough)

Associated with
An interface on a
machine.

11011111 00000001 00000011 00000010
223 . 1 . 3 . 2



Subnets
↳ Can reach each other without router
↳ Share subnet part of IP

CIDR

Classless InterDomain Routing

↳ Defines the subnet part of an IP address.

a.b.c.d/x ← x is the number of bits of subnet address

11001000 00010111 00010000 00000000 (Host address)

200.23.16.0/23 → first 23 bits represent the subnet

Every IP from 200.23.16.0 to 200.23.17.255 are in the same Subnet

Who owns an IP Address

ICANN gives blocks of IP Addresses to an ISP.

↳ An Isp give a smaller block of addresses to a subnet organization (Kth)

↳ KTH gives its subnets a range of IP addresses.

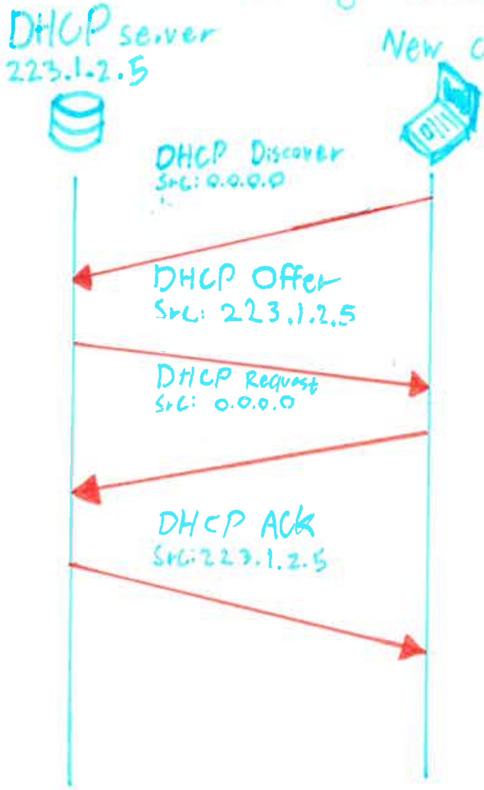
↳ A subnet has a DHCP Server that gives IP addresses to all its devices.

DHCP

Dynamic Host Configuration Protocol.

↳ A DHCP Server leases IP addresses dynamically to devices on a network.

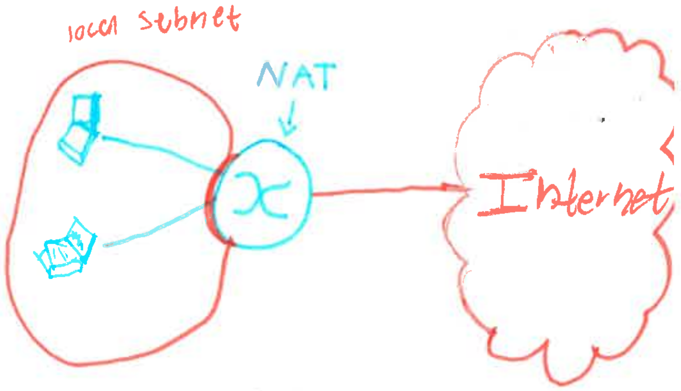
↳ reuses addresses when they are Not in use any more.



Network Address Translation (NAT)

↳ Local networks use a single Ip from the outside and port numbers are used to select which device inside the network is used in the connection.

↳ Adds security. Every new device does not need a new global ip, ISP can change without changing local devices. Inside computers are not visible to outside world.



↳ Port field is 16 bit, Allowing 65000 connections.

Internal IP	Port	External IP	Port
138.76.29.7	5000	1.1.1.1	80
138.76.29.7	2000	129.68.1.1	80
138.76.29.6	8000	129.69.1.1	25
138.76.29.6	6000	130.11.1.2	80

↳ contraversal since it mixes layers, P2P connections are limited (port forward) Should use IPv6 instead.

NAT Problems

A server behind a nat cannot get connection requests from outside and only be seen within subnet. (Minecraft)

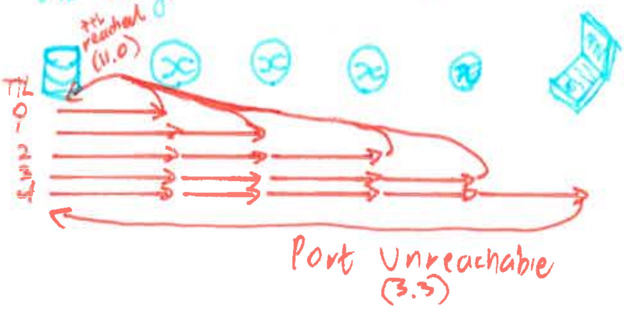
Solution 1:
Port forward, bind a specific Port on router to a server. The server's global IP becomes The IP of the NAT

Solution 2:
Universal plug & Play
Make server IP public

Solution 3:
Relay
Place server outside NAT and all traffic goes through the external server.

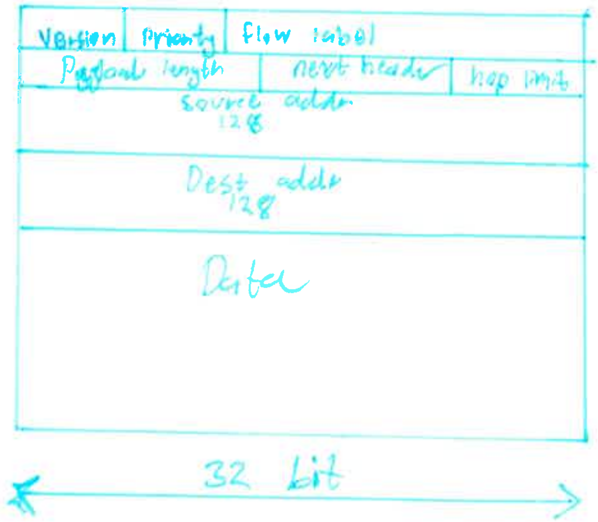
Traceroute & ICMP

Send UDP messages to destination with increasing TTL starting from zero. With an unused port number. Wait for "port unreachable" reply. TTL messages will inform of the routers on the way.



IPv6

Allows a lot more IP addresses 32→128 bit
has no fragmentation, header is 40 byte always



Routing

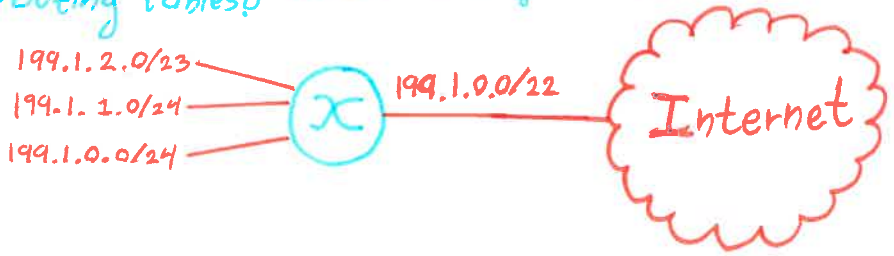
Finding the best path to a distination.
- Distance-Vectors (bellman ford)
- Link-State (Dijkstra)

What is the "best" Path?
- Fewest Hops
- bandwidth
- buissiness relation

RIP/OSPF/BGP

Aggregation

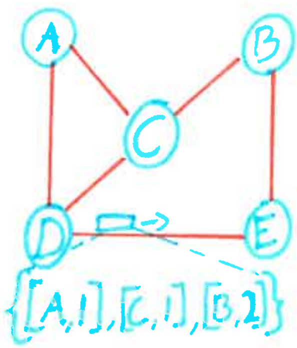
Routers can combine IP Addresses to make a bigger net with a smaller prefix. This is usually done manually and leads to smaller routing tables.



Routing Information Protocol

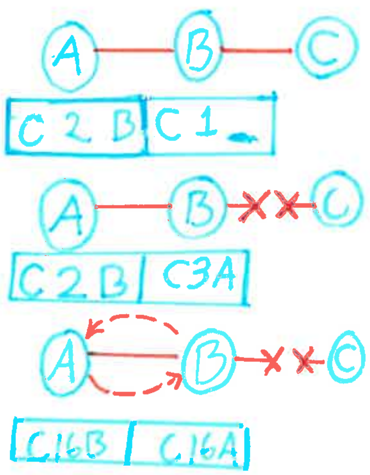
(RIP) - RFC 1058 & RFC 2453
- 1 hop ⇒ directly connected
- 16 hops ⇒ infinity
  - ↳ cannot support network with diameter greater than 15.

Nodes send vectors (Arrays) of the number of hops and direction of all known nodes. (via UDP) All nodes update their table if a shorter route is found.



IF E knew a longer Path to a node, it will be updated with the value from D+1

RIP Problem: Count to infinity.



B is directly connected to C. A knows of this And routes data to C through B

IF the connection B↔C is broken B May be informed from A that

A and B will loop packages until eventually the cost of both nodes reaches 16, but this may take several minutes.

Solution: Poision Reverse

When A advertices to B its routes it sets distance of all routes that go past B as 16. This means B will never think it can reach C over A and eventually A will be informed of the disconnect to C.

RIP Disadvantages:

- ↳ Slow convergence (info spreads slowly)
  - ↳ every neighbour only speak every ~30 seconds.
- ↳ Instability it takes long to fix broken connections.
- ↳ Uses high bandwidth.
- ↳ Max diameter is 15

Advantages:

- ↳ Easy to set up
- ↳ Widely avalible

Open Shortest path First

OSPF RFC 2328

build link state advertisements (LSA) and distribute them to all other routers. Then each router uses Dijkstras algorithm to find the best path.

uses the IP Protocol directly and not UDP/TCP. (Protocol field = 89)

3 protocols:

Hello:

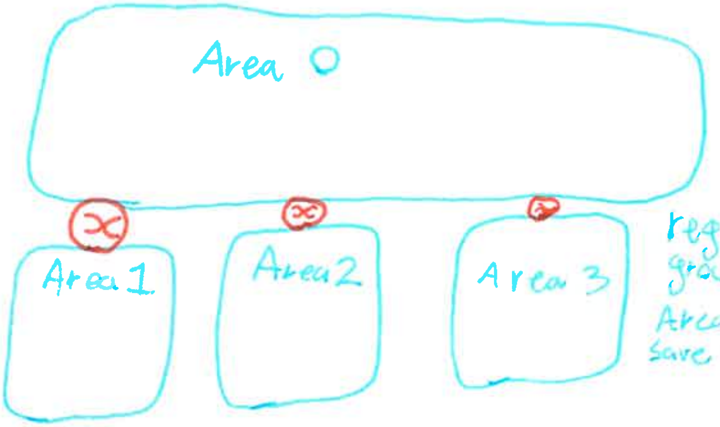
Check & authenticate routers.

Exchange:

Exchange LSA with neighbours

Flood:

When changes are made flood every router with new information



regions are grouped into Areas to save computing

Advantages:

↳ no dependencies
↳ full topology
↳ easy troubleshooting
↳ Fast convergence

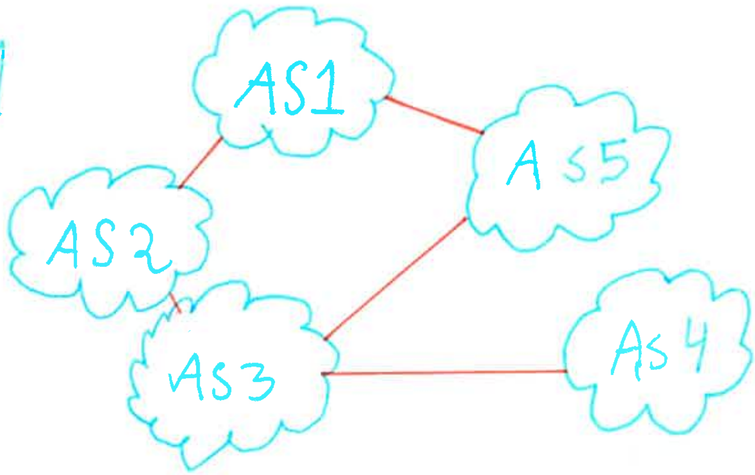
Disadvantages:

↳ uses lots of memory.

Border gateway protocol

Reachability over Autonomus Systems. Uses TCP.

AS number	Network
3	Stanford
32	MIT
2839	KTH
1653	SUNET



Chapter 6 - Link Layer

Terminology

Hosts & routers - Nodes
Wires/wireless - Link
Packets - Frame

Mac Protocol

Channel Partitioning:

↳ Divided chanel into pieces:
time/frequency/colour
Exclusive access

Random Access

↳ Collisions may happen but they may recover.
Ethernet

Taking Turns

↳ coordinated access.
Bluetooth.

Mac Address

48 bit addres that is hardware encoded in network chip
eg: 0A-24-BB-76-09-AD

Each device has a unique Address

ARP

Look up to pair IP with MAC address. When a MAC address is missing from table, broadcast to FF-FF-FF-FF-FF with the matching IP address and the device will reply with its MAC Address.

A sends IP Packet with B's IP Address to R's Mac address. R knows B's Mac-IP Pair and sends message to B's Mac address

CSMA (carrier send multiple Access)

1 Listen
2 If nobody is sending{
    send
  }else{
    wait
  }
3 goto 1

Collisions may still occur.

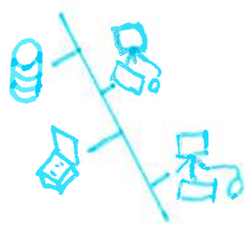
Collision detection:
When sender detects collision, stop sending and wait a random amount of time before sending again

Ethernet

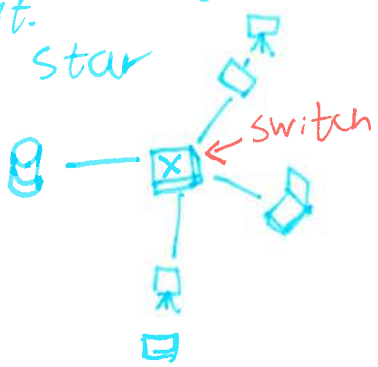
Bus: (outdated)
Nodes share connections, but allow collissions.

Star:
switch in center that organizes where data is sent.

Bus



Star



Ethernet Frame

Premable	Dest ADDR	Src ADDR	type	Data	CRC

Premable: 8byte
Seven bytes of 10101010
One byte of 10101011

Synch sender & receiver clocks

Address: 2×6 byte
48 bit MAC addresses.

Type:
Indicate higher level Protocol.
IPV4/IPV6/ARP...

CRC:
Eror checsum.

NO handshake/No Acks

Ethernet Switch

Stores & forwards Ethernet frames using CSMA/CD. Invisible to hosts and do not require configuration

When it does not know of the Dest Address it floods network. Learns who Senders are.

Load Balancing

An intermediary that balances which server that handles requests.