C

Gabriele Dragotto
gabriele.dragotto@studenti.polito.it

Computer Networking I

1 Introduction

MESSAGE

*Meaningful expression with **well defined bounds** such size.*

1. **BOUNDS**
   It has specific length
2. **INDEPENDENT ON THE WAY YOU RECIVE IT**
3. **NON-PHYSICAL**
4. **REPRESENTATION**
   You can change it without compromising the meaning and content
   PAYLOAD is the part of transmitted data that is the actual intended message.

INFORMATION

*How **much you do not know** from the sender of the message before you have read the message.*

PAYLOAD

$$l = log_b N$$

BITS

EXAMPLE: THE SUN

Sun rises in east and sets in west. This message is carrying **zero** amount of information.

SYMBOLS AND ALPHABETS

Each letter is called a symbol and belongs to a set of possible letters called the alphabet

EXAMPLE: A WORD IN THE ALPHABET

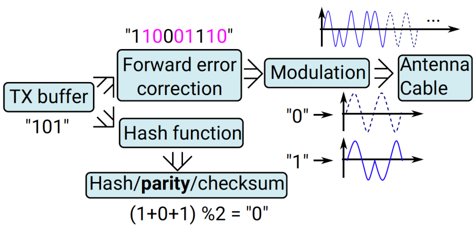
Let's computer the amount of information carried by a word of length L

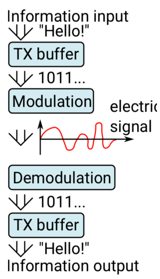
$l = log_2(Alphabet) \cdot L$

FRAME

*A frame is a digital **data transmission unit** and includes **headers and control parts***

ELECTRONIC TRASMISSIONS





MULTICAST

*IP multicast is a method of sending IP datagrams to a **group of interested receivers in a single transmission.***

DIGITAL AND ANALOG

*Each **digital** communication has a **finite alphabet**, while **analog has not***
Both in modulation and demodulation, the states are finite.

ERROR CORRECTION

*Error control are techniques that enable **reliable delivery** of digital data over **unreliable channels.***

HASH CHECKSUM **INTEGRITY CHECK**

ARQ AUTOREPEAT

*The receiver asks for **fragments** of messages to be re-sent because **integrity checks failed.***

FRAGMENTATION

INTEGRITY AND ACKs

REASSEMBLY

*Information are rearranged together as of the **fragmentation process** needed for the **integrity check.***

- OVERHEAD
  Data sent with the purpose of **controlling the transfer** of user information or the **detection and correction of errors.**

STREAMS

Message with no specific size.

- FRAGMENTATIONS
  Streams in nowadays networks are possible because of **fragmentation of the informations flow.**

DATAGRAM MODE

*Datagram is the natural way of communication between machines, with **finite packet sizes.***

Sender	Receiver
Representation in machine-readable format	Playback to the final consumer
Fragmentation into smaller packets	Reconstruction of sequence, retransmission requests
Adding error-correction information	Error-correction, error-checking
Modulation	Demodulation and decoding
Transmission via ether/cable	Detection of incoming signal

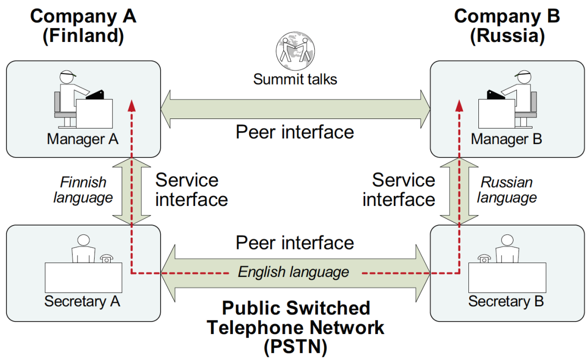
OSI

*The Open Systems Interconnection model is a **conceptual model** that standardizes the communication functions of a telecommunication system without regard to their underlying internal structure and technology.*

INTEROPERABILITY

Layer	Function	Example
Application (7)	Services that are used with end user applications	SMTP,
Presentation (6)	Formats the data so that it can be viewed by the user Encrypt and decrypt	JPG, GIF, HTTPS, SSL, TLS
Session (5)	Establishes/ends connections between two hosts	NetBIOS, PPTP
Transport (4)	Responsible for the transport protocol and error handling	TCP, UDP
Network (3)	Reads the IP address form the packet.	Routers, Layer 3 Switches
Data Link (2)	Reads the MAC address from the data packet	Switches
Physical (1)	Send data on to the physical wire.	Hubs, NICS, Cable

- **LAYER**
  A layer serves the **layer above** it and **is served by the layer below it.**
  The **service interface UP** serves the user.
  The **service interface DOWN** is interacting with whatever is needed
  The **peer interface** is interacting with the same layer on the **opposite side.**

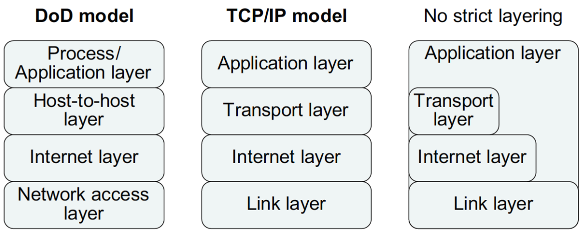


- **HEADER**
  Is a piece of information attached to the information by a **single layer.**

- **RELAY**
  Network, data and physical layers serve as **relay system** as to communicate between **end open systems.** It can operate in **one or more of these layers.**

LESS STRUCTURED OSI

OSI is a strict model, therefore is not actually used in telecommunications.



- **APPLICATION**
  Specifies the **shared protocols and interface methods** used by hosts in a communications network. DOS TELNET BROWSER
  OVERRIDE: Some or all functions of the lower layers. **NO PHY**
- **TRANSPORT**
  Ensure that the data is delivered exactly the way it was sent, handling **fragmentation and reassembling**. TCP UDP
- **INTERNET/NETWORK**
  Methods, protools and specification that are used to **transport and deliver datagrams.** IP ICMP IPv6
- **LINK**
  Transfer data between **nodes and network elements**.
  ETHERNET MAC PHY

MAC

*A media access control address of a computer is a **unique identifier assigned to network interfaces** for communications at the data link layer of a network segment.*

1. ARBITRATION IN SHARED CHANNELS
2. SYGNAL POWER PHY
3. IMPLEMENTS ARQ

PHY

*Provides the **mechanical, electrical, functional, and procedural means** to activate, maintain, and deactivate physical connections for bit transmission between data link entities*

1. COMMUNICATE WITH MAC
2. ELECTRICAL MODULATION

2 LAN & Ethernet & Internet

LAN

*A local area network is a computer network that interconnects computers within a **limited area***

5 Components of the LAN:

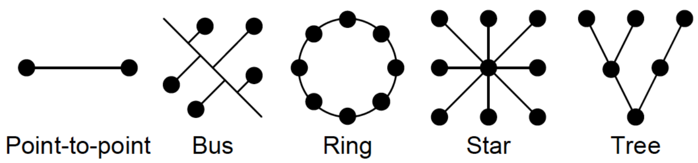
1. STATIONS
   Workstations running **a software** as to access to LAN
2. LAN INTERFACE
   Hardware for **connecting** different workstations
3. PHYSICAL TRANSMISSION MEDIUM
   Device used to connect interfaces: **ethernet cable**
4. PHYSICAL INTERFACE UNITS
   provides an **interface** between the station **hardware and the PTM**
5. INTERCONNECTING DEVICE
   repeaters, connectors and switches

LAN PHYSICAL TOPOLOGY

Shape of the wire used to build up the LAN

PHISICAL INDIPENDENT LOGICAL

vLAN



1. BUS
   Frame is transmitted in the entire network. **Terminators** remove headers.
2. STAR AND TREE TOPOLOGY
   Operates through forwarding of packets based on their **destinations**.
   1. BROADCAST
   2. FRAME-SWITCHED

LAN TYPE

A. NON-BROADCAST (SWITCHED)
   ADJACENCY: nodes can only communicate with nodes they are next to
B. BROADCAST (SHARED MEDIUM)
   COLLISION-DOMAIN: LAN or a part of a LAN in which there will be a collision if multiple stations transmit at the same time

MAC AS TRAFFICLIGHT

Broadcast networks need MAC for the same reason streets **need traffic** lights and rules of the road to prevent collisions. When **2 or more stations** transmit simultaneously, their signals **will collide and interfere with each other**

COLLISION DETECTION

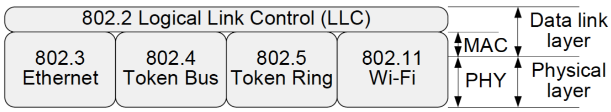
1. **SIMPLE AVOIDANCE**
   Transmitter starts sending while there is **silence** with **permission.** Other don't transmit.
2. **SIMPLE AVOIDANCE WITH COLLISION DETECTION**
   Transmitter starts sending while there is **silence without permission.** If 2 talks, they both stop.
3. **SCHEDULED ACCESS**
   Central authority decides who's going to send data.
4. **TOKEN PASSING**

ETHERNET

Ethernet is a family of computer networking technologies commonly used in LAN, MAN and WAN

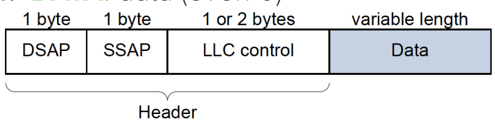
IEEE 802. PHYSICAL+DATALINK

ETHERNET DATALINK

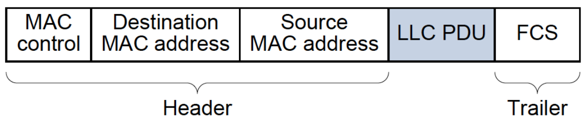


Divided into **2 components**:

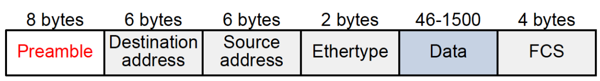
1. **LLC**
   1. **DSAP:** RX link to **network layer** protocol.
   2. **SSAP:** TX link to **network layer** protocol.
   3. **LLC CONTROL: control information** (ack, command, responses)
   4. **DATA:** data (even 0)



2. **MAC**
   MAC CONTROL: contains any control information needed for the functioning of the MAC protocol.
   DEST/SOURCE MAC
   LLC PDU: data from the LLC layer
   FCS: frame checksum.The MAC layer **is responsible for detecting errors & discarding** while LLC **keeps track of discarded and ask RT.**



ETHETNET FRAME



1. **PREAMBLE 8B**
   it consists of 8 bytes of alternating "1"s and "0"s, ending in 11, as to **synchronise clocks**
2. **DEST/SOURCE MAC 6Bx2**
3. **ETHERTYPE 2B**
   Defines versioning **(IPv4)**

LAN SWITCH

*Multiport node that allow stations to attach directly and **forward incoming packet to their correct MAC destination or broadcast***

COMMUNICATION NETWORK

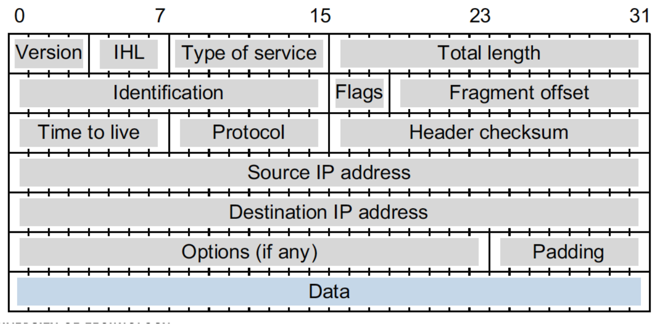
A system of ***interconnected intermediate systems****, end systems, and other equipment allowing information to be exchanged*

SUBNET: *small part of the network*

INTRA-EXTRA NET

CONNECTIONLESS: cheapest path available

INTERNET LAYER



HEADER $(20 \pm 40) Bytes$

The header has a **fixed-length component of 20 bytes** plus a **variable-length** component consisting of options that can be up to 40 bytes

1. **4b. VERSION:** Indicates **version number**
2. **4b. IHL: Length of the header MIN 5** (20 bytes) **MAX 15** (60 bytes)
3. **8b. TOS:** Species **priority, delay, throughput,** reliability…
4. **16b. TOTAL LENGTH: Total length MIN 28B MAX 15 65535B**
5. **8b. TTL:** Maximum hops to pass, **decrements it by 1. 6°OF SEP**
6. **8b. PROTOCOL: TCP = 6; UDP = 17; ICMP = 1**
7. **16b. HEADER CHECKSUM:** Verifies the **integrity of the header** of the IP packet.Since some header fields change, the **header checksum is recomputed and verified** at each point that the IP header is processed
8. **32b. SOURCE/DEST ADD:** Written down in binary or dotted-decimal.
9. **(40Bytes). OPTIONS:** Allows the packet to **request special treatment** such as route to be taken by the packet, timestamp at each router, etc.

DATA MIN 8B MAX 65KB

Must contain an **integer number of bytes.**

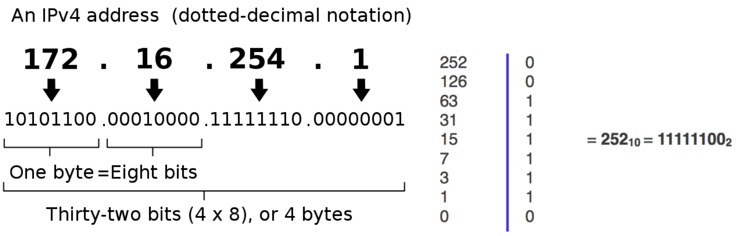
BIT ERROR

1. **BER(ate)**
   number of bit errors per **unit time.**
2. **BER(atio)**
   number of **bit errors divided by the total number** of transferred bits during a **studied time interval**
3. P_e **BIT ERROR PROBABILITY**
   is the **expectation** value of the **bit error ratio**
   $P_e = b \cdot BER_{atio}$ $\quad$ $P_e = 1 - (1 - BER_{atio})^b$

IP ADDRESS

IPv4 addresses may be ***represented in any notation expressing a 32-bit integer value. They are most often written in the dot-decimal notation, which*** *consists of four octets of the address expressed individually in decimal numbers and separated by periods.*

A.B.NETWORK_ID[C].HOST_ID[D]



RESERVER ADRESSES

0.0.0.0. **NULL ADDRESS**

172.16.0.0 to 172.31.255.255 **LAN**

192.168.0.0 to 192.168.255.255 **LAN**

10.0.0.0 to 10.255.255.255 **LAN**

ROUTING TABLE

Is a data table stored in a router or a networked computer that ***lists the routes*** *to particular network destinations*

To forward a packet, the routing node does the following:

- Look at the destination address of the packet, find the network part of that address.
- Look at the routing table, line by line, and see if a given network address is handled by that line
  - If it is - forward the packet to the specified gateway through specified interface
  - Otherwise - continue to next line
- If no rule matches - drop the packet (yes **THAT** simple)

Example routing table:

network address	interface	gateway
192.168.10.*	1	10.2.4.7
10.2.4.*	1	0.0.0.0 (myself)
..*.*	2	130.230.0.4(default gateway)
Discard packet		

Question 4
Correct
Mark 1.00 out of 1.00
Flag question

Mark the options that indicate why very small packet size is not always the best solution.

Select one or more:

- [x] a. Smaller packets will more likely require fragmentation of data, which forces creation of reassembly logic on receiver ✓
- [x] b. It is harder to encrypt small packets ✗
- [ ] c. It is easier to retransmit one big packet rather than a bunch of small packets
- [x] d. Smaller packets are harder to represent on physical layer in a way that is resistant to errors. ✓
- [x] e. Relative overhead of the headers grows as packet size decreases ✓

CIDR BLOCK

CIDR is principally a bitwise, prefix-based standard for the representation of IP addresses and their routing properties.

A.B.C.D/X

The number following the slash is the prefix length, **the number of shared initial bits**, counting from the most-significant bit of the address.

Address format	Difference to last address	Mask	Addresses		Relative to class A, B, C	Restrictions on *a*, *b*, *c* and *d* (0..255 unless noted)	Typical use
			Decimal	2^n			
a.b.c.d / 32	+0.0.0.0	255.255.255.255	1	2^0	1/ 256 C		Host route
a.b.c.d / 31	+0.0.0.1	255.255.255.254	2	2^1	1/ 128 C	*d* = 0 ... (2*n*) ... 254	Point to point links (RFC 3021)
a.b.c.d / 30	+0.0.0.3	255.255.255.252	4	2^2	1/ 64 C	*d* = 0 ... (4*n*) ... 252	Point to point links (glue network)
a.b.c.d / 29	+0.0.0.7	255.255.255.248	8	2^3	1/ 32 C	*d* = 0 ... (8*n*) ... 248	Smallest multi-host network
a.b.c.d / 28	+0.0.0.15	255.255.255.240	16	2^4	1/ 16 C	*d* = 0 ... (16*n*) ... 240	
a.b.c.d / 27	+0.0.0.31	255.255.255.224	32	2^5	⅛ C	*d* = 0 ... (32*n*) ... 224	Small LAN

ICMP

It is used by network devices, including routers, to send ***error messages and operational information.***

LIBRARY OF PREDEFINED MESSAGES

INTERNET/NETWORK

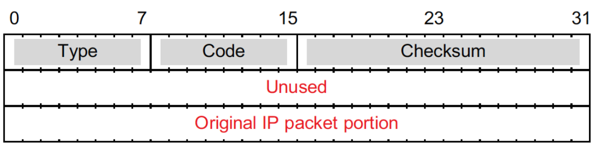
1. **ENCAPSULATED IN IP**
   Protocol field is set to 1
2. **PING + TRACEROUTE**
   Incremental TTL **trace route the way message is delivered.**
3. **NO SWITCHES AND ROUTERS**
   Can't be pinged

Sometimes the ip return **failures**

A. **TRANSIENT FAILURES**
   Such invalid checksum, are **generally ignored**

B. **SEMI-PERMANENT FAILURES**
   Need to **be reported immediately:** TTL=0, destination unreachable, etc

ICMPV4 FORMAT



1. 8b. TYPE: Indicates **message's type**
2. 8b. CODE: Describe the **purpose of message**.
3. 16b. CHECKSUM: Detect errors in the ICMP message, similar to ipv4
4. 32b. UNUSED: Contains all zero
5. ?. IP PACKET PORTION: Contains original **IP header and 8B of data**

PING

*ping is a **software utility** used to test the reachability of a host on an Internet Protocol (IP) network*

RFC 1122 states that "every host must implement an ICMP Echo server"

- **2 QUERY MESSAGES**
  An **ICMP Echo Request** message is a probe sent by a user to a destination system, which responds with an **ICMP Echo Reply message**

MTU DISCOVERY

Maximum transferable units you can send trough your network.

- **1500bytes**
  20B for **Network**, 20B for **Transport, 1460 Bytes for data**
- **TOO BIG**
  Packets are too large - return ICMP **Destination Unreachable messages with a code meaning "fragmentation needed and DF set"**
- **TCP**
  Will take care of transportation, or in case **the app**

TRACEROUTE

*Traceroute is a diagnostic tool for **displaying the route** (path) and measuring **transit delays of packets** across an Internet Protocol (IP) network.*

- **INCREASING TTL**
- **3 ICMP PER HOP**
- **PATH MAY CHANGES DURING THE PROCESS**

3 Planning and deploying network protocols

SUBNETTING

SUBNET: *small part of the network*

$$2^{HostBits} - 2$$

1. **ACTUAL HOST ADDRESS**
   Actual host acting
2. **NETWORK IP**
   Network IP address with **host bits set to 0**
3. **BROADCAST IP ADDRESS**
   Network IP address with all **host bits set to 1**
4. **MASK IP ADDRESS**
   All **network address are 1, all host are 0**

HOW TO PLAN

- **DOCUMENTS**
- **FUTURE GROWTH**
- **PHYSICAL ACCESSIBLE?**

LAYERED RESPONSIBILITIES

- **CORE DISTRIBUTION LAYER**
  High throughput devices, high processing power, forward between subnets.
- **ACCESS LAYER**
  Provide connectivity, **isolate subnets from each other.**

```
– 10.0.0.0/24
——— 10.0.0.0/25
————— 10.0.0.0/26 – available
————— 10.0.0.64/26
——————— 10.0.0.64/27 – available
——————— 10.0.0.96/27 – servers
———— 10.0.0.128/25
————— 10.0.0.128/26 – developers
————— 10.0.0.192/26 – accounting
```

Network Bits	Subnet Mask	Bits Borrowed	Subnets	Hosts/Subnet
8	255.0.0.0	0	1	16777214
9	255.128.0.0	1	2	8388606
10	255.192.0.0	2	4	4194302
11	255.224.0.0	3	8	2097150
12	255.240.0.0	4	16	1048574
13	255.248.0.0	5	32	524286
14	255.252.0.0	6	64	262142
15	255.254.0.0	7	128	131070
16	255.255.0.0	8	256	65534
17	255.255.128.0	9	512	32766
18	255.255.192.0	10	1024	16382
19	255.255.224.0	11	2048	8190
20	255.255.240.0	12	4096	4094
21	255.255.248.0	13	8192	2046
22	255.255.252.0	14	16384	1022
23	255.255.254.0	15	32768	510
24	255.255.255.0	16	65536	254
25	255.255.255.128	17	131072	126
26	255.255.255.192	18	262144	62
27	255.255.255.224	19	524288	30
28	255.255.255.240	20	1048576	14
29	255.255.255.248	21	2097152	6
30	255.255.255.252	22	4194304	2

4 ARP, DHCP and NAT

ARP LAYER

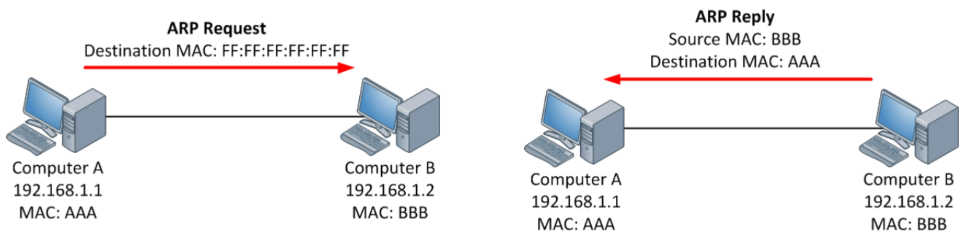
The Address Resolution Protocol is a request and response protocol whose **messages are encapsulated by a link layer protocol.**

LINK LAYER

- **LOCAL NETWORK**
- **LIMITING FACTORS**
- **MINIMISES THE OVERHEAD WITH CACHE**

ARP PROCEDURE

- **2 COMPUTER CONNECTED BY ETHERNET**
- **LOOK FOR DESTINATION IP'S MAC**
- **IF IP IN CACHE STOP; ELSE BROADCAST ARP REQUEST**
  To MAC address **FF:FF:FF:FF:FF:FF** asking a **reply only rom the specified ip to the source IP and Mac address** with a **payload asking for Mac.**
- **GET ARP RESPONSE**
  With the Mac address



DHCP

*A DHCP server enables computers to **request IP addresses** and networking parameters automatically, reducing the need for a network administrator or a user to configure these settings manually.*

APPLICATION LAYER

- **IP - GATEWAY - SUBMASK**
- **DNS**
- **NTP**
- **NO SECURITY**
- **ARBITRATION BASED ON DELAYS**

DHCP WORKING

- IP: 0-0-0-0/0 - MAC: VENDOR ONE
- BROADCAST ANYONE FOR DHCP ADDRESS
- OFFER FROM DHCP SERVERS
- CLIENT REQUEST TO A SINGLE SERVER
- ACKNOWLEDGMENT OF ADDRESS

DISCOVERY

PACKET HEADERS (UDP)

Source: **IP=0.0.0.0** and **Port:68**
Destination: **IP=255.255.255.255** and **Port=67**

DHCP HEADER

Operation code = **1 - discovery**
Client Hardware **MAC=VENDOR ONE**

OFFERING

PACKET HEADERS (UDP)

Source: **IP=DHCP-SERVER** and **Port:67**
Destination: **IP=255.255.255.255** and **Port=68**

DHCP HEADER

Operation code = **0x02**
Ip adders: **Your IP**
Server IP: **Your IP.254**
Client MAC: **Your MAC**

REQUEST

PACKET HEADERS (UDP)

Source: **IP=0.0.0.0** and **Port:68**
Destination: **IP=255.255.255.255** and **Port=67**

DHCP HEADER

Server ip adders: **Your IP.254**
Client MAC: **Your MAC**

ACKNOWLEDGMENT

PACKET HEADERS (UDP)

Source: **IP=ip.254** and **Port:67**
Destination: **IP=255.255.255.255** and **Port=68**

DHCP HEADER

Ip adders: **Your IP**
Server ip adders: **Your IP.254**
Client MAC: **Your MAC**

SOCKET

SOCKET

A network socket is an ***internal endpoint for sending or receiving data*** *at a single node in a computer network. Concretely, it is a representation of this endpoint in networking software*

IP + PORT + PROTOCOL

- LISTEN ON MULTIPLE INTERFACE
- TRANSMIT TO ON SPECIFIC INTERFACE

PORT

PORT

In the internet protocol suite, a port is an ***endpoint of communication*** *in an operating system.*

NAT TECHNIQUES

*Network address translation is a method of **remapping one IP address space into another by modifying network address information in Internet Protocol (IP) datagram packet headers while **they are in transit across a traffic routing device.***

NETWORK AND TRANSPORT LAYERS

- IP, PORT, BOTH
- 1 TO 1
  Used as to **change ip address without breaking the connections.**
  Useful to change **servers**
- N TO M
  Used from **ISP** to allocate address to **users when network doesn't have enough ip addresses.** COSTs AND IPs SAVING
- N TO 1
  Map **each internal socket** into an **outgoing socket.**
  Whenever a request is coming on **the outside to the outgoing socket, reroute it to the internal socket.**

  INT SOCKET: 10.0.0.0 port A —> EXT_SOCKET: a.b.c.d PORT B
  Request to a.b.c.d:B is routed to 10.0.0.0:A

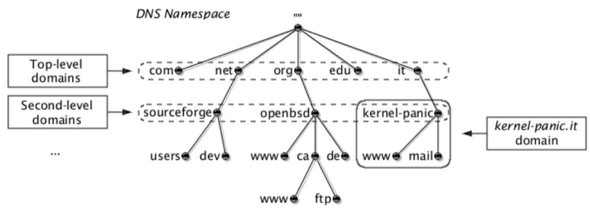
5 DNS

NAT TECHNIQUES

***Hierarchical decentralized naming system** for computers, services, or other resources connected to the Internet or a private network.*

NETWORK AND TRANSPORT LAYERS

- HIERARCHICAL OF DNS SERVERS
- REGULATED BY IANA



RESOURCE RECORDS

NAME - TTL - CLASS - TYPE - VALUE

Type = '**A**' - Address
Maps the hostname in Name filed to the IPv4 address in Value

Type = '**NS**' - Name Service
Name field contains the domain name, and Value is the hostname of the Authoritative Name server for the domain.

Type = '**CNAME**' - Canonical name
Defines an alias hostname

Type = '**MX**' – Mail Exchange
Links the domain name with the mail server for that domain.

Type = '**TXT**' - Text
Auxiliary record attaches a text string to the hostname.

EXAMPLE: TUT

Example of reply – dig MX tut.fi

```
;; ANSWER SECTION:
tut.fi.         415 IN  MX  10 mail2.tut.fi.
tut.fi.         415 IN  MX  0 mail.tut.fi.
tut.fi.         415 IN  MX  10 mail1.tut.fi.

;; AUTHORITY SECTION:
tut.fi.         172617 IN  NS  kaustinen.cc.tut.fi.
tut.fi.         172617 IN  NS  ns-secondary.funet.fi.
tut.fi.         172617 IN  NS  ressu.cc.tut.fi.

;; ADDITIONAL SECTION:
mail.tut.fi.    160 IN  A   130.230.162.19
mail.tut.fi.    160 IN  A   130.230.162.20
mail1.tut.fi.   415 IN  A   130.230.162.19
mail2.tut.fi.   415 IN  A   130.230.162.20
```

SOURCE OF AUTHORITY RECORDS

SOA

NAME TTL CLASS TYPE NAME-SERVER EMAIL-ADDR (SN REF RET EX MIN)

```
example.com.    IN    SOA    ns.example.com. hostmaster.example.com. (
                              2003080800 ; sn = serial number
                              172800     ; ref = refresh = 2d
                              900        ; ret = update retry = 15m
                              1209600    ; ex = expiry = 2w
                              3600       ; nx = nxdomain ttl = 1h
                              )
```

GLUE RECORDS

*A glue record is simply the **association of a hostname** (nameserver, or DNS) with an **IP address** at the registry.*

- **CLIENT BASED QUERIES**
  Every time you ask for a server, the **root server, secondary root server,** will give you **only a piece** as to resolve address.

TYPES OF QUERIES

RECURSIVE QUERY
With a recursive name query, client requires that the DNS server **respond** to the client with either the requested **resource record or an error** **no redirects**

ITERATIVE QUERY
Client allows the DNS server to **return the best answer** it can give based on its cache or zone data. No answer: the **best possible information it can return is a referral**

ADVANCED DNS

- **DNS CLUSTERING**
- **DYNAMIC DNS**
- **GEO DNS**

SECURITY

- **MITM ATTACK**
  Build a fake, take over a **node between DNS** endpoint and respond with the fake ip address
  DNSSEC - IETF Specs - AUTH
- **DNS SPOOFING**
  Send email from a different **SMTP** server.
  SPF: v=spf1 a mx include: mail.dragotto.net -all
  DKIM: Add DNS **fingerprint in email.**
  DMARC: Check that SPF and DKIM rules are followed.

6 TCP & UDP

TRANSPORT LAYERS

Transport layer protocols have some characteristics in common

1. **USABLE PRIMITIVES**
   For the **app layer.** Abstract the connection and its problems
2. **MULTIPLEX CONNECTIONS**
   With different ports
3. **LISTENING SOCKETS**
   Allow accepting connections in a **unified manner**
4. **END-TO-END**
   They are only implemented at **end systems.**

UNIQUE IDENTIFIER

SOURCE DESTINATION IP	PROTOCOL FIELD VALUE	SOURCE AND DESTINATION PORT

DEMULTIPLEXING

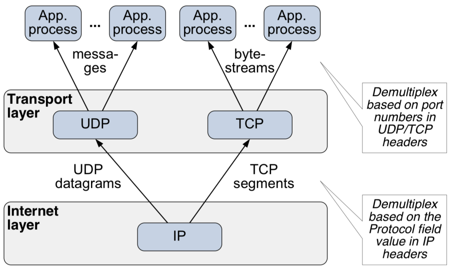
DEMUX** is the **reverse of the multiplex (MUX) process** – which split the unique signal input into **different streams

LOWER TO HIGHER

MULTIPLEXING

***MUX** or Multiplexing is the process in which multiple Data Streams, coming from different Sources, are combined and Transmitted **over a Single Data Channel** or Data Stream.*

HIGHER TO LOWER



PORT NUMBERS

*Port is an **endpoint of communication** in an OS.*

2^{16} = 65, 536

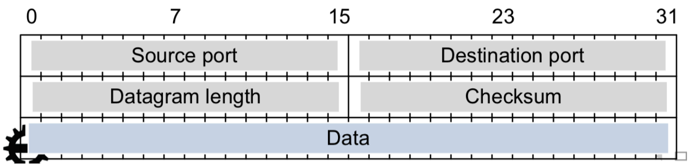
1. **0-1023**
   Well knowns
2. **1024-49151**
   Registered ports
3. **49151-65536**
   Dynamic ports
4. **EPHEMERAL PORTS**
   Port dynamically assigned to client and freed up when no longer needed

UDP

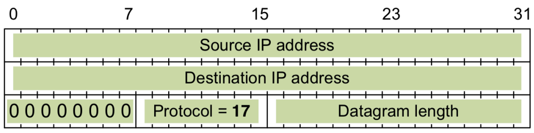
A. **MESSAGE DATAGRAM ORIENTED**
   Small messages (eg: **DNS, DHCP**)
B. **CONNECTIONLESS**
   Establishing a connection before sending data is not required
C. **STATELESS**
   Neither side keeps track of the connection
D. **UNRELIABLE**
   No **ACK** or **retransmissions.**

1. UNRELIABLE
2. ERROR CONTROL (opt)
3. DATA INTEGRITY VERIFICATION
   UDP checksum applies to the entire UDP datagram plus a pseudo header pre fixed at the time of checksum computation
4. NO FLOW-CONGESTION CONTROL
5. NO FEEDBACK MESSAGE

UDP DATAGRAM STRUCTURE



1. 16bits. SRC & DST PORTS
2. 16bits. DATAGRAM LENGTH
3. 16bits. CHECKSUM
   If the length of the datagram is not a multiple of 16 bits, the datagram will be padded out with "0"s to make it a multiple of 16 bits.
   PSEUDO-HEADER DURING COMPUTATION



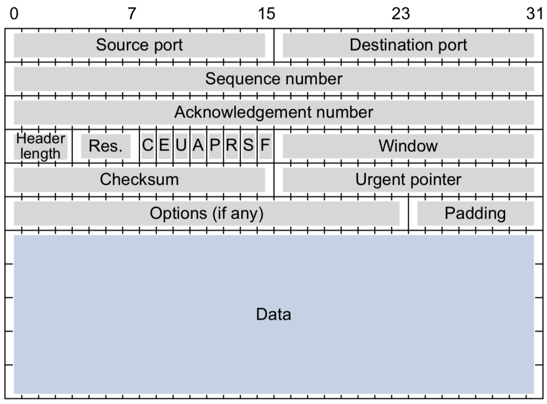
A. CORRUPTED
   Notify via ICMP
B. NO COMPUTATION OF HEADER
   Fill with all 0 the **checksum field. Then set to all 1**

TCP

A. BYTE STREAMS ORIENDED
   Data bytes are **delivered in-order** to an application process
B. CONNECTION ORIENTED
   A connection must be **established** between hosts
C. STATEFUL
   Both sender and receiver **keep track of the state** of the session
D. RELIABLE
E. FULL-DUPLEX
   Both hosts can send infos in the **same channel**

1. FLOW AND CONGESTION CONTROL
   TCP regulates the rate at which the sending host transmits data
2. ERROR CONTROL (mandatory)
   TCP checksum applies to the entire TCP segment plus a pseudo header pre fixed at the time of checksum computation. **Trigger resending when not passed**
3. FEEDBACK BASED

TCP PACKET



1. **16b.** **SRC & DST PORTS**
2. **32b.** **SEQUENCE NUMBER**
   Identifies the position of the first data byte of this segment in the sender's byte stream. **IF SYN=1 THEN SN=ISN+1 with ISN** $2^{32} - 1$

3. **32b. ACK NUMBER**
   If the **ACK bit is set to 1**, identifies the **sequence number of the next data byte** that the sender expects to receive Also indicates that the sender has **successfully received all data up to** (*but not including*) this value
4. **4bits. HEADER LENGTH**
   Specifies the length of the TCP header in 32-bit words
5. **9bits. CONTROL BITS**
   ECN Explicit congestion **notification.**
   CWR Sending host has received a TCP **segment with ECE=1.**
   ECE Host is Congestion-capable
   URG Urgent data
   ACK Ack number is correct
   PSH Pass the already received data to the application
   RST drop all buffers and reset the connections
   SYN used to establish a TCP connection
   FIN end the connection
6. **16bits. WINDOW**
   Bytes the receiver of this segment is **ready to accept**
7. **16bits. CHECKSUM**
8. **16bits. URGENT POINTER**
   If the URG bit is set to 1, **specifies a positive offset that must be added** to the **Sequence number** field value of the segment to yield the sequence number of the last byte of urgent data
9. **?. OPTIONS**
10. **?. PADDING**

MAXIMUM SEGMENT SIZE

$$MSS = MTU - Headers$$

A. IPv4 MSS=MTU - 40Bytes = 1460Bytes
B. IPv6 MSS=MTU - 60Bytes
C. Ethernet2 MTU= 1500Bytes

ACKS

A. **PIGGYBACKED**
A data segment from host A to host B can also contain an ACK for data sent in the direction from B to A. **REDUCE HEADERS AND TRAFFIC**

B. **CUMULATIVE**
ACKs for complex packets can be sent together when **everything has been received.**

C. **DELAYED**
3:Sent when no ACK for the **previous segment**, or no message in the last **500ms** or there is a gap in **SN.**

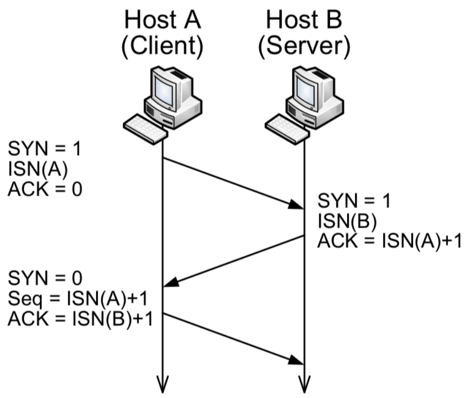
D. **DUPLICATE**
1:Out of **order** packet. Ack signals the **expected packet**

3WAY HANDSHAKE

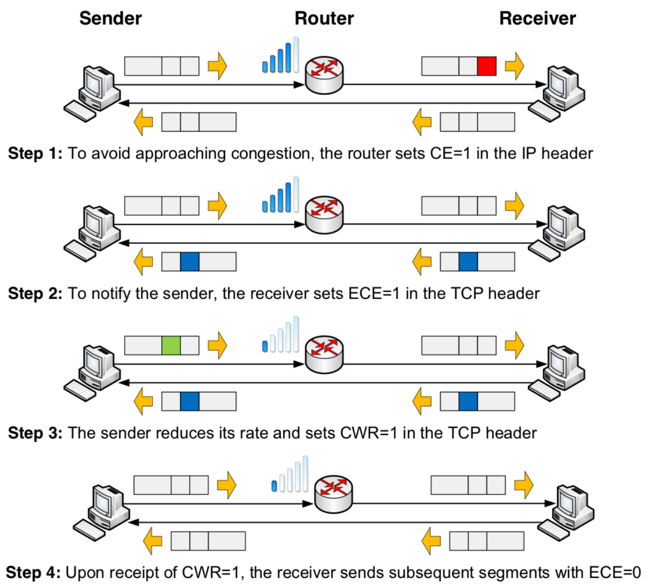
A. **A->B SYN SEGMENT**
With no app-data and **SYN=1 and ISN(A)**
Client: active open

B. **B->A SYN/ACK SEGMENT**
With no app-data and **SYN=1 and ACK=ISN(A)+1 and ISN(B).**
Server: passive open

C. **A->B ACK SEGMENT**
With no app-data and **SYN=0 and ACK=ISN(B)+1**



ECN - CE



A. **TOS IPv4 FIELDS**
ECT capable and **CE** experienced

CONNECTION CLOSE

A. B->A FIN=1
   Server active close
B. A->B FIN=1
   Client: passive close
C. B->A ACK FIN SEGMENT
D. TIME_WAIT=2*MSL
   ACK Received and **Buffering** period

FEATURE	IMPLEMENTATION
MULTI/DEMULTI PLEXING	Port numbers
ORDER + SEGMENTATION	1. CONN ESTABLISHMENT/TERMINATION 2. MSS OPTION 3. PATH MTU DISCOVERY
ERROR CONTROL	1. CHECKSUM 2. SEQ NUMBERS 3. ACKs 4. RETRANSMISSION AND TIMED RETRANS
FLOW CONTROL	1. RECIVE WINDOW 2. SILLY WINDOW AVOIDANCE 3. NAGLE ALGORITHM 4. WINDOW SCALE OPTION
CONGESTION CONTROL	1. KARN'S ALGORITHM 2. INITIAL WINDOW 3. SLOW START 4. CONGESTION AVOIDANCE 5. FAST RETRANSMIT AND RECOVERY 6. ECN-SUPPORT

7a Flow control

FLOW CONTROL

*Process of managing the **rate of data transmission** between two nodes, providing a mechanism for the **receiver** to **control the transmission speed***
HW/SW E2E/H2H

- **CONGESTION CONTROL**
  Prevents overloading by acting on **middle-point nodes and the sender**
- **COMPROMISE**
  High throughput, resource utilisation and low control overhead
- **SYNCHRONISES DIFFERENT SPEEDS**

CONTROL SYSTEMS

A. OPEN CONTROL *A PRIORI*
   Guessing the rate by estimating. **No feedback loop.**
   Initial negotiation then agreement.
B. CLOSE CONTROL
   Adjust the estimation. **Use a feedback loop**
   Used in TCP controls.
   SIGNALING: **In-band or out-of-band**

ON/OFF

*Two signals allow the **receiver** to signal wether is able or not to **accept data.***

ON=READY OFF=NOT READY

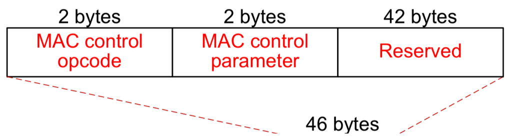
- XON/XOFF
  Allow the receiver to signal is state / software. **In-band signaling**

PAUSE

*Signal the sender to pause the data stream for a **defined timespan***

- IN BAND
- FULL-DUPLEX
  Computer/Computer — Computer/Switch — Switch/Switch
- NEW PAUSE
  Replaces the old ones
- ALLOWED MESSAGES
  Are only pause between endpoints.

MAC FRAME



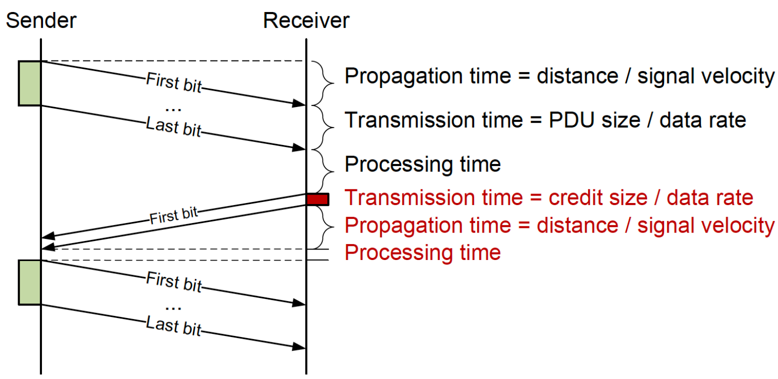
- MAC-CONTROL OPCODE
  PAUSE – 0x0001
- MAC CONTROL PARAMETER
  Specifies the duration of the **pause in BIT-TIME**
- RESERVED
  Filled with 0s-

STOP AND WAIT

*Stop-and-wait flow control is the simplest form of flow control. In this method, the **receiver indicates its readiness** to receive data for each frame, the message is broken into multiple frames.*

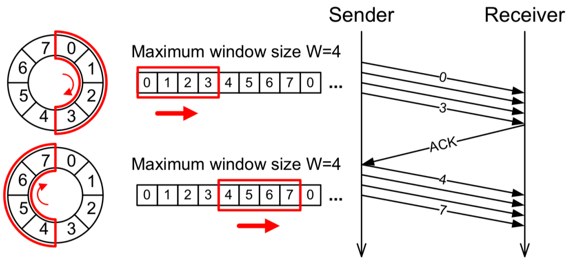
SEND FRAMES AFTER ACKs (Credit)

A. PROPAGATION TIME
B. TRANSMISSION TIME
C. PROCESSING TIME
D. SAME FOR ACK



SLIDING WINDOW

Generalisation of ***stop-and-wait*** *for* ***more than 1 PDU***



- **Credits issued at the end of windows**
  - Data rate in bits/s $= R$
  - Data PDU transmission time $= T_{tr,d}$
  - Credit transmission time $= T_{tr,c}$
  - Propagation delay $= T_{pr}$
  - Window size in PDUs $= W$

$$R_{effective} = \frac{WT_{tr,d}}{WT_{tr,d} + T_{tr,c} + 2T_{pr}} R$$

- **Credits issued after each PDU, advancing the window by 1**
  - Data rate in bits/s $= R$
  - Data PDU transmission time $= T_{tr,d}$
  - Credit transmission time $= T_{tr,c}$
  - Propagation delay $= T_{pr}$
  - Window size in PDUs $= W$

$$R_{effective} = \min\left(\frac{WT_{tr,d}}{T_{tr,d} + T_{tr,c} + 2T_{pr}} R, R\right)$$

TCP FLOW CONTROL

TCP receiver sends ***window size*** *and* ***the ACK.***

$$[ACK, ACK + WNDW - 1]$$

- **GENERALISATION**
  $min(rwnd, cwnd)$ between **receiver window** and **congestion window.**
- **RWND=0 - ACK= SQN(A)-1**
  Sender asks to **wait before sending data.**

BDP IN TCP

In data communications, ***bandwidth-delay product*** *is the product of a data link's capacity (in bits per second) and its round-trip delay time*

$$BDP = B_{dwith} \cdot R_{TTime}$$

$$LinkUtilisation = \frac{RWDW}{BDP}$$

- Consider a 1000 km fiber link has a 5 ms one-way delay
  - The velocity of signal propagation in optical fiber is about 200,000 km/s
- The RTT (i.e., the two-way propagation delay) $= 2 * 5$ ms $= 10$ ms
- When operating at 10 Gbits/s, the BDP $= 100 * 10^6$ bits or $12.5 * 10^6$ bytes
- The upper bound on the link utilization is

$$\frac{\text{rwnd}}{\text{BDP}} * 100\% = \frac{65,535}{12.5 * 10^6} * 100\% = 0.52\%$$

- To improve efficiency, the receive window size should be increased

SILLY WINDOW SYNDROME

each ACK advertises a ***small amount of space available*** *and each segment carries a small amount of data*

- **RECEIVER HEURISTIC**
  ACK WITH ON: Instead of sending a window advertisement immediately, the receiver waits until the available space **reaches either 50% of the total buffer size or a maximum-sized segment**
- **SENDER HEURISTIC**
  CLUMPING: collect the data transferred in each call before transmitting it in a **single, large segment.** NAGLE ALGORITHM

7b Congestion Control

CONGESTION

*Congestion is the state of a network in which the **incoming load exceeds the network capacity** for a **period of time**, large enough for the queues in the network to grow over their normal size*

CONTROL VS OVERPROVISIONING

- **COLLAPSE**
  is the situation in which an **increase in the offered load** results in a **decrease in capacity** of the network to react to traffic

BOTTLENECK

*Performance or capacity of an entire system is **severely limited by a single component***

METHODS

1. **RATE-BASED CONTROL**
   the sender is aware of a **specific data rate,** and the receiver or a intermediate system informs the sender of a new rate that it must not exceed
2. **WINDOW-BASED CONTROL**
   the sender keeps **track of the window** – – a c**ertain amount** of data that it is allowed to send before new feedback arrives

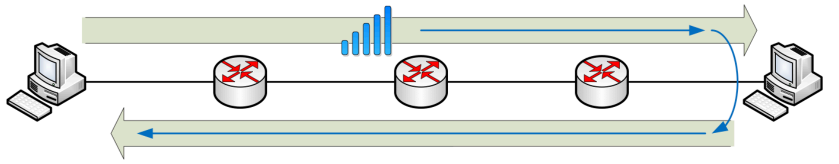
ICMP SOURCE QUENCH

*In IPv4, a device that is **forced to drop packets** due to congestion provides feedback to the senders that overwhelmed it by sending them ICMPv4 Source Quench messages*

DEPRECATED

1. **FAST**
2. **EXPLICIT FEEDBACK**
3. **HOW TO SIGNAL CONGESTION STOP?**
4. **BANDWIDTH CONSUMING**
5. **WHEN SEND THE QUENCH?**

ECN



1. **ECN-CAPABLE**
   Hosts transmit their willingness to **accept.**
   SYN SEGMENT: ECN + WDW
   SYN/ACK SEGMENT: ECN
2. **CE**
   Congestion experienced **signals** the problem.

AIMD

*AIMD combines **linear growth of the congestion** window with an **exponential reduction** when a congestion takes place.*

KRAN ALGORITHM

RTO is estimated with the KRAN Algorithm based on the ***RTT***

1. RESEND QUEUE
   A generic packet is kept in the retransmission queue before **being deleted**
2. RTT AND VARIANCE
   The algorithm sums to the RTT the **medium variance experienced.**
   $min(RTO) = 1s$

A. RETRANSMISSION AMBIGUITY
   ACKs from a retransmission or original transmission?
B. KRAN SOLUTION
   Round trip time estimation is based only on unambiguous ACKs, for **segments that were sent only once.** On successive retransmissions, set **each timeout to twice the previous** one

INITIAL WINDOW

1. ssthresh
   is used to determine whether the **slow start or congestion avoidance algorithm** is used to control data transmission
2. cwnd
   is a **sender-side limit** on the amount of data the sender can transmit into the network before receiving an ACK

SLOW START

To probe the network path and to determine ***how much bandwidth is available,*** *TCP uses an algorithm called slow start*

1. CWND=IWindow
2. INCREMENT
   For every ACK received that acknowledges new data, the **cwnd is incremented** by the number of **bytes in the sender's MSS**
3. CWND>SSTRESH || PACKET LOSS
   Congestion avoidance: linear increment over exponential.

LOSS DETECTION

1. DUPLICATE ACKs
2. TIMEOUTS - FAST RETRANSMIT
   After **4 identical ACKs** TCP performs a **retransmission** of what appears to be the missing segment, without waiting for the retransmission timer to expire

TCP-RENO

Avoid duplicating a ***slow start*** *by continuing transmitting datas. Fast recovery* ***helps*** *recovery the* ***data sending*** *after a congestion.*

1. SSTRESH = $max(\frac{F_{light}}{2}, 2 \cdot MSS)$
2. SEND LOCAL SEGMENT
   $CWND = sstresh + 3 \cdot MSS$
3. EACH DUPLICATE ACK
   cwnd is incremented by 1 **full-sized segment**

TCP NEW-RENO

1. PARTIAL ACKs
   an ACK that acknowledges **some but not all** of the segments sent before fast retransmit
2. cwnd
   is a s**ender-side limit** on the amount of data the sender can transmit into the network before receiving an ACK

8 Application Layer

APP LAYER

*Application Layers interact with the **network layer** by telling where to forward the message and with the **transportation layer protocols.***

STREAM vs MESSAGE

1. **ASYMMETRICAL DESIGN**
   Client request server replies or vice versa
2. **P2P SYSTEM**
3. **HYBRID SYSTEMS**
4. **HIGH LEVEL INTERACTION**
5. **NOT INCLUDED**
   Encryption (usually), error correction, identification of connections

POSIX SOCKETS

*Defines the way that **applications** should interact with the **operating system** TCP/IP with OS*

1. CREATE **socket()**
2. ATTACK TO INTERFACE **bind()**
3. WAITING **listen()**
4. ACCEPT **accept()**
5. ESTABLISH **connect()**
6. SEND AND RECIVE **write() read()**
7. CLOSE **close()**

HTTP PROTOCOL

POST HEAD GET OPTIONS (PUT DELETE PATCH)

Request_type=one of <GETIPOSTITRACEIDELETE...> **Target** /HTTP <version>
Options (option=value)
empty_line
Message body empty_line

SMTP

Simple mail transfer protocol sends email through internet.

1. MAIL
   Specifies the **return path**
2. RCPT
   Specifies the **recipients**
3. DATA
   Specifies the **message**
4. AUTH
   Authentication in **plaintext.** In **SMTPs** connection is **TSL encrypted** and auth are **base64 encoded**

- **BINARY ARE WEIGHTY (ATTACHMENTS)**
- **NO ENCRYPTION**
- **SECURITY CONCERNS**

POP3 *Post Office Protocol version 3 - used to manipulate emails*

1. **LIST, QUIT, DELETE, MOVE**

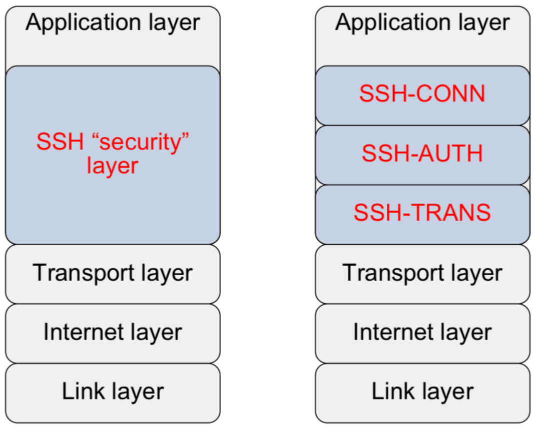
- **CAN'T FETCH HEADER**
  In IMAP you can just fetch the header.
- **READ/NOT READ**
- **NO FOLDERS**
- **NO FILTERS**
-

FTP *File transfer protocol is used to manipulate files*

1. **CONTROL PORT (21)**
   Carries control messages. **Plaintext unencrypted protocol**
2. **DATA PORT**
   Carries data transfer. **Active / passive** *(preferred)*

9a SSH and TSL

SSL *Protocol that operates in TCP as to provide secure connections*



1. **BETWEEN APPLICATION AND TRANSPORT**
   Encrypt and compress
2. **AUTHENTICATION**
3. **CONNECTION**
   Multiplexing of several logical channels into a single tunnel

TSL *Protocol that operates between **app and transport***

1. **HTTPS= HTTP + TSL**
2. **RECORD PROTOCOL**
   Symmetric encryption with **integrity checks.**
3. **HANDSHAKE PROTOCOL**
   Second layer with **asymmetric public private key. Attacks are detected.**

9b Security engineering

DEFINITION

*Security engineering is a specialized field of engineering that **focuses on the security aspects in the design of systems** that need to be able to deal robustly with possible sources of disruption, ranging from natural disasters to malicious acts.*

1. **POLICY**
   How your system works.
2. **INCENTIVES**
   List of **reasons why** you would want the program to operate as it should
3. **MECHANISM**
   Combination of all mechanisms in order to **implement the policy, incentives and assurances**
4. **ASSURANCES**
   Provide assurances to operators of the platform **that the security is implemented, can be negative!**

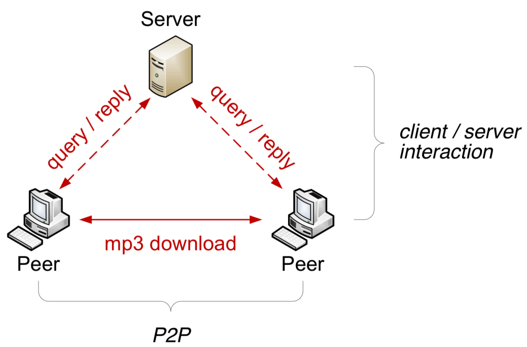
P2P and Overlay

WWW

- **UBIQUITOUS**
- **ASYMMETRICAL**
  Low rate links in up, huge **links in download**
- **FIREWALL**
- **NETWORK ADDRESS TRANSLATOR (NAT)**
- **DYNAMICAL IP ASSIGNMENT**
- **HUGE PROBLEM FOR EVERYTHING DIFFERENT THAN WWW**

P2P SYSTEM: NAPSTER

- **SEPARATION OF SIGNALING AND DATA**
  CENTRAL INDEX SERVER
  Lists all the data. **Single point of failure: NAPSTER was shut down.**
- **HANDSHAKE**
  Central server exchange **information to and from the peers.**
- **SCALABILITY OF CENTRAL SERVER**
- **RELIABILITY**
  dDos attack, legal prosecution
- **SECURITY**
  mp3 authenticity and protocol **encryptions.**



GNUTELLA

*Gnutella is a large peer-to-peer network. It was the first **decentralized peer-to-peer network** of its kind, leading to other, later networks adopting the model*

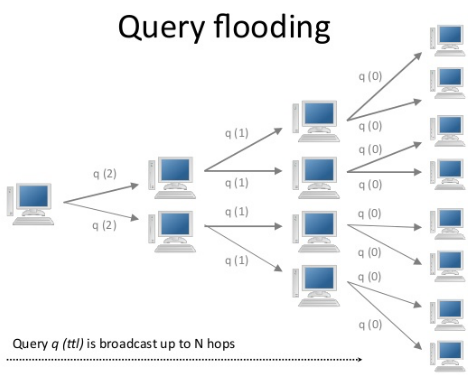
- DECENTRALISATION
- SEARCHING STRATEGY
  Flooding: a simple computer network routing algorithm in which every incoming packet is **sent through every outgoing link** except the one it arrived on.
- SUPERNODES
  Supernode is any node that also serves as **one of that network's relayers and proxy servers,** handling data flow and connections for other users. This semi-distributed architecture allows data to be decentralized without requiring excessive overhead at every node.

- OPEN SOURCE
- AUTONOMOUS, HARD TO SHUT DOWN
  PING - PONG - QUERY - QUERY_HIT - PUSH
- GNUTELLA 2
  Users are **leaf,** more experienced users become **hubs**

- NO FULLY DECENTRALISED
- FULLY DECENTRALISED ARE SLOW

QUERY FLOODING

Query flooding is a method to search for a resource on a P2P network. It is simple but scales very poorly and thus is rarely used. Early versions of the Gnutella protocol operated by query flooding; newer versions use more efficient search algorithms.



BOOTSTRAPING

*A bootstrapping node, also known as a rendezvous host, is a node in an overlay network that **provides initial configuration information** to newly joining nodes so that they may successfully join the overlay network.*

vLAN and Tunneling

CONGESTION

A virtual private network (VPN) ***extends a private network across a public network****, and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network.*

IMPROVED WAN CONNECTING VLANS

- **COST SAVING**
- **SCALABILITY**
- **FLEXIBILIY**

- **ELEMENTS**
  Client, server, tunnel, endpoints, protocol, **edge devices**
  P stuff: devices and network set up by **provider.**
  C stuff: devices and network set up by **customer (island).**

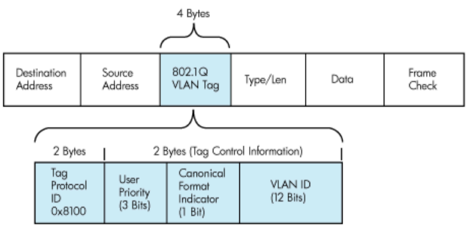
TYPES OF VPN

1. **REMOTE ACCESS VPN**
   connect individual remote **users to corporate networks**
2. **INTRANET VPN**
   connect a number of **LANs (Intranets) located in multiple geographic areas** over the shared network infrastructure
3. **EXTRANET VPN**
   limited access of corporate resources is **given to business partners**, such as customers or suppliers, enabling them to access shared information

4. **PE BASED**
   The provider setup the VPN
5. **CE BASED**
   The VPN is setup by customers' devices.

ETHERNET VLAN

A virtual LAN (VLAN) is any ***broadcast domain that is partitioned and isolated*** *in a computer network at the data link layer*



TUNELING PROTOCOLS

- **PRIVACY**
- **INTEGRITY**
- **AUTHENTICATION**
- **CERTIFICATION**
- **ACCESS CONTROL**
- **KEY MANAGEMENT**

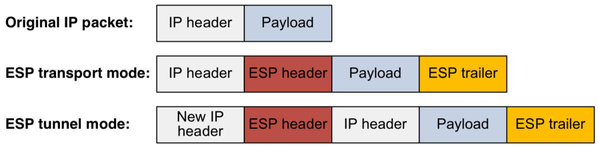
1. **L2 - PPTP**
   DES - 3DES - PAP
2. **L3 - GRE**
   Packet inside the packet

1. **L3 - IPSEC**

   Site-site and user-site with **strong encryption.** The project is **modular and has 3 main components.**

   AH - Auth Header: ensures integrity and authentication of the packet. Comes with a **new header** and the information about **the algorithm used.**

   ESP - Encapsulated Security Payload: ensures **data privacy** by encryption in addition to integrity and authentication. Contains **header + authentication data.**



   IKE and SA (Security Association)