

Contents

1	TCP	2
1.1	UDP Protocol	2
1.1.1	UDP Header	2
1.2	ARQ Protocols	3
1.2.1	Stop and Wait	3
1.2.2	Go Back N	4
1.2.3	Selective Retransmission	4
1.3	TCP Protocol	4
1.3.1	TCP Header	5
1.3.2	TCP Flags	5
1.3.3	TCP Options	5
1.3.4	Sequence Numbers	6
1.3.5	Inici i Fi de la connexió	6
1.3.6	State Diagram	6
1.3.7	Congestion Control	6
1.3.8	Retransmission time-out	7
2	Network Applications	8
2.1	DNS	8
2.1.1	Protocol	8
2.1.2	Message Format	8
2.2	Email	10
2.2.1	SMTP protocol	10
2.2.2	MIME	11
2.2.3	Retrieval Protocols	11
2.3	Web	11
2.3.1	HTTP Messages	12

1 TCP

1.1 UDP Protocol

UDP significa User Datagram Protocol, es un "protocol" basat en datagrames. La capa de transport està identificada per ports amb numeros de 16 bits, per exemple 1245.

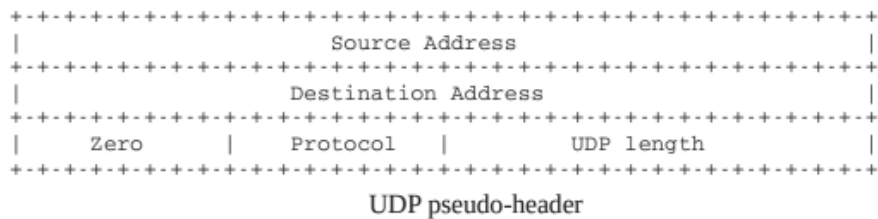
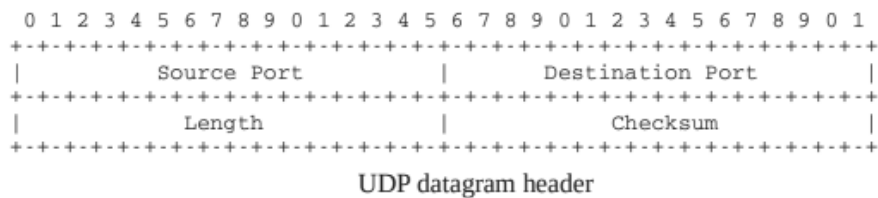
Datagram Service:

- No reliable
- No error recovery
- No ack
- Connectionless
- No flow control

UDP PDU es coneix com a datagrama UDP UDP s'utilitza principalment, quan s'han d'enviar missatges curts i en aplicacions en temps real com VoIP, video-conferencies, streaming...

1.1.1 UDP Header

El header de UDP conté 8 bytes, el checksum es calcula utilitzant el header, el pseudo-header i el payload. S'ha de tenir en compte que si al pseudo-header s'utilitza el PAT el checksum s'ha d'actualitzar.



1.2 ARQ Protocols

ARQ significa Automatic Repeat reQuest. Aquest protocol estableix comunicació entre dos endpoints amb detecció d'errors, recuperació d'errors i control de flux.

Hi ha tres tipus bàsics de protocol ARQ:

- Stop and Wait
- Go Back N
- Selective Retransmission

Coses bàsiques del ARQ:

- Connection oriented
- Tx/Rx buffers
- ack
- Retransmission Timeout
- Sequence Numbers

ARQ també s'utilitza pel flow control. Flow control consisteix en evitar que el remitent que transmeti a un rati de PDU superior al que el receptor pot treballar.

Amb Stop and Wait, si el receptor és més lent, els acks s'alenteixen i el remitent disminueix el rendiment.

Amb Continuous Transmission protocols, s'utilitza una finestra de transmissió. Aquesta finestra té el número màxim de PDUs sense confirmació que es poden transmetre. Si es sobrepassa aquesta finestra, el remitent deixa de transmetre. Cal afegir que el Stop and Wait es un protocol de finestra amb Tx window = 1 PDU.

Existeix el concepte de finestra òptima que consisteix en la finestra mínima que permet el màxim rendiment.

1.2.1 Stop and Wait

Quan el remitent està preparat, envia un paquet al receptor i Tx s'inicia.

Quan el paquet arriba complet al receptor, aquest genera un ack que envia al remitent.

Quan el remitent rep el ack envia el següent paquet.

Cada cop que s'envia un datagrama, s'inicia un RTO. Si no arriba o arriba amb errors, no s'envia l'ack d'aquest datagrama i quan expiri el RTO es tornarà a enviar el mateix datagrama.

S'utilitzen numeros de seqüència per a numerar els PDU i l'ack de cada PDU per saber quin paquet s'ha enviat i confirmat.

En termes d'eficiència, si no hi ha errors, l'objectiu és enviar els paquets seguits independentment del retard de propagació.

Per a calcular l'eficiència s'utilitza la següent formula $\frac{1}{1+2*a}$, on $a = \frac{t_p}{t_t}$

1.2.2 Go Back N

En aquest protocol, l'emissor va enviant paquets i va rebent els acks, però si el receptor rep un error o una PDU fora d'ordre, aquest no envia acks i descarta totes les PDU fins que li arriba la que toca.

Quan es produeix un RTO, el remitent va enrere i torna a enviar tots els PDU a partir de l'últim ack que ha rebut.

1.2.3 Selective Retransmission

Es pràcticament igual al Go Back N, però el remitent només reenvia PDU quan es produeix un RTO i el receptor guarda PDU fora d'ordre i els ack corresponents a aquestes PDUs.

1.3 TCP Protocol

És un protocol fiable ja que disposa de:

- Error Recovery
- ack
- Connection oriented
- Flow Control

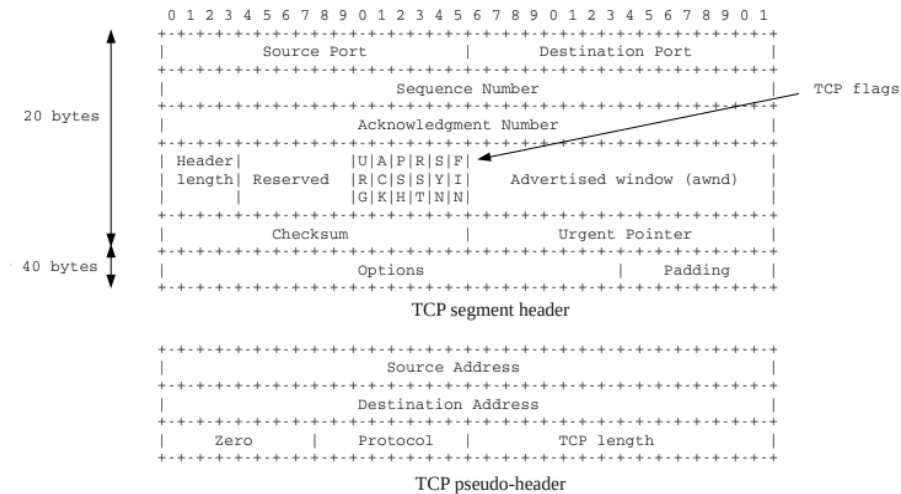
El TCP PDU és coneix com a segment TCP. Disposava de control de congestió, és a dir, adapta el rendiment TCP a les condicions de la xarxa. Els segments tenen la mida òptima ja que la mida màxima d'un segment és variable MSS (Maximum Segment Size). Per acabar aquests conceptes bàsics de TCP, aquest s'usa en aplicacions que requereixin fiabilitat com pot ser una web, ssh, ftp...

El funcionament del TCP és el següent:

- Protocol de finestra variable: $wnd = \min(awnd, cwnd)$
- Sempre que arriba un segment, s'envia un ack sense haver de processar les dades
- La finestra anunciada ($awnd$) s'utilitza per al control flow
- La finestra de congestió ($cwnd$) s'utilitza pel control de congestió

1.3.1 TCP Header

El header de TCP té una mida variable, però sabem segur que hi ha 20 bytes fixes i després totes les opcions que es vulguin respectant el tamany màxim de 60 bytes. Com a UDP també hi ha checksum que es calcula utilitzant el header, el pseudo-header i el payload. També s'ha de tenir en compte si s'utilitza el PAT.



1.3.2 TCP Flags

- URG (Urgent): S'utilitza el Urgent Pointer. Apunta al primer byte urgent. Per exemple Ctrl+C.
- ACK: S'utilitza el camp ack. Aquest flags s'usa sempre excepte el primer segment enviat pel client.
- PSH (Push): el remitent indica de fer push de totes les dades del buffer a l'aplicació.
- RST (Reset): Interromp la connexió.
- SYN: S'utilitza per establir connexió.
- FIN: S'utilitza per acabar la connexió.

1.3.3 TCP Options

- Maximum Segment Size (MSS): S'utilitza en el TWH (three-way handshaking) per inicialitzar el mss. En IPv4 s'inicialitza a MTU-40(mida de les capçaleres d'IPv4 i TCP sense opcions).

- Window Scale factor: S'utilitza en el TWH. La awnd es multiplica per $2^{\text{WindowScale}}$. Això permet tenir una awnd més gran que 2^{16} bytes.
- Timestamp: S'utilitza per calcular el Round Trip Time (RTT).
- SACK: En cas d'errors, indica quins blocs s'han rebut correctament pel Selective ReTx.

1.3.4 Sequence Numbers

El número de seqüència identifica el primer byte del payload. I el número del ack identifica el proper byte que espera rebre.

1.3.5 Inici i Fi de la connexió

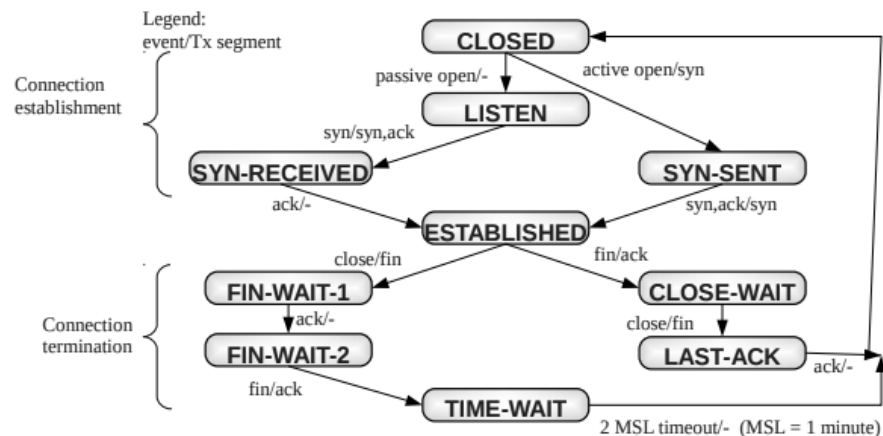
El client sempre envia el primer segment.

Els segments de TWH tenen payload = 0.

Els segments SYN i FIN consumeixen 1 número de seqüència.

El número de seqüència inicial és random per temes de seguretat.

1.3.6 State Diagram



1.3.7 Congestion Control

El protocol TCP interpreta les pèrdues de paquets com a congestió. Per a la congestió, s'aplica l'algorisme de Slow Start/Congestion Avoidance.

Aquest algorisme funciona de la següent manera:

```

Initialization:
    cwnd = MSS ; NOTE: RFC 2581 allows an initial window of 2 segments.
    ssthresh = infinity ;

Each time an ack confirming new data is received:
    if (cwnd < ssthresh) {
        cwnd += MSS ; /* Slow Start */
    } else {
        cwnd += MSS * MSS / cwnd ; /* Congestion Avoidance */
    }

When there is a time-out:
    Retransmit snd_una ;
    ssthresh = max(min(awnd, cwnd) / 2, 2 MSS) ;
    cwnd = MSS ;

```

On `snd_una` es el primer segment sense ack i `ssthresh` és el threshold entre Slow Start i Congestion Avoidance.

Durant SS `cwnd` augmenta ràpidament mentre que durant CA augmenta més lentament buscant el punt òptim d'ample de banda.

1.3.8 Retransmission time-out

El RTO s'activa sempre que hi ha algun ack pendent. Quan està activat va decrementat i quan arriba a 0 es produeix una retransmissió. Quan arriba un ack confirmant nous paquets, és calcula el RTO i fa reset si hi havia acks pendents però si no n'hi havia cap el RTO es para.

Per calcular el RTO, el que envia els segments TCP calcula el RTT mitjà (`srtt`) i la variança (`rttvar`), que ens permeten calcular el RTO com

$$RTO = srtt + 4 * rttvar$$

Cada cop que es retransmet un segment es duplica el RTO.

2 Network Applications

2.1 DNS

DNS (domain name system) permet als usuaris fer servir noms en comptes d'adreces IP. Per exemple `pornhub.com` en comptes de `66.254.114.41`.

El DNS consisteix en una base de dades a nivell mundial on cada entrada es coneix com a Resource Records (RR). I cada nom està associat a 1 o mes RRs. A més, cal afegir que el DNS no fa distinció de majúscules i minúscules.

Per accedir a la base de dades DNS s'utilitzen Name Servers (NS) que poden contenir RR permanents i en cache. Les RR en cache són eliminades després d'un cert temps.

Un root server és un entry point a la jerarquia del domini. Aquests root servers estan distribuïts per tot el món i n'hi ha de 13 tipus.

Per a "traduir" de nom a IP en les màquines Unix, primer és mira el fitxer `/etc/hosts` per mirar si hi ha una entrada amb el nom que es busca, si no el troba utilitza el NS que te al `/etc/resolv.conf`

2.1.1 Protocol

El protocol DNS és un paradigma client-servidor. S'utilitza UDP en missatges curts, altrament s'utilitza TCP. Aquestes consultes es fan a través del port 53.

Per a que no es saturi un servidor s'utilitza load balancing, que consisteix en tenir varies adreces IP associades a un mateix nom i a cada consulta es tradueix una IP diferent per evitar sobrecarregar un servidor.

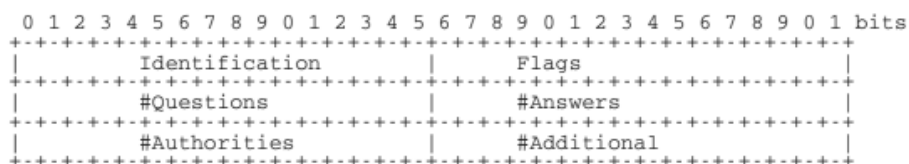
Si la resolució de nom és produeix de manera local, diem que s'ha fet una resolució recursiva; mentre que si s'accedeix a internet llavors passa a ser una resolució iterativa

2.1.2 Message Format

	Header (12 bytes)	
/	Question (variable)	/
/	Answer (variable)	/
/	Authority (variable)	/
/	Additional (variable)	/

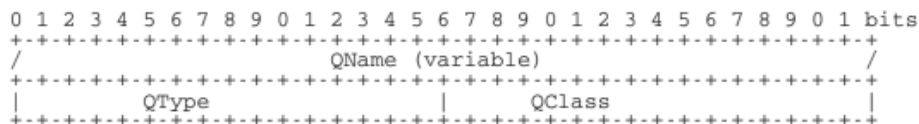
Header:

- Identification: 16 bits aleatoris per identificar la query i la resposta.
- Alguns flags
 - Query-Response, QR: 0 per query, 1 per resposta.
 - Authoritative Answer, AA: quan s'activa indica una resposta autoritativa.
 - Recursion Desired, RD: quan s'activa, indica que es prefereix la resolució recursiva.
- Els altres camps indiquen el número de preguntes, respostes, "autoritats" i camps addicionals del missatge.



Question:

- QName: s'indica el nom a resoldre.
- QType: s'indica el tipus de pregunta
 - Address, A.
 - Name Server, NS.
 - Pointer, PTR: per una resolució inversa.
 - Mail eXchange, MX: Domain Mail Server address.
- Qclass: per adreces d'Internet és 1.



Resource Records:

Els camps Answer, Authority i Additional estan composts per RRs:

- Name, Type, Class: igual que al camp Question.
- TTL (Time To Live): numero de segons que la RR pot estar en cache.
- RDLenth: mida de RR en bytes.
- Rdata: per exemple una adreça IP si el Type és 'A' o un nom si el Type és 'NS', 'MX' o 'CNAME'.

2.2.2 MIME

MIME Significa Multipurpose Internet Mail Extensions.

A diferència de SMTP aquest protocol permet enviar objectes multimedia. A més, el format intern es invisible pels usuaris.

En mails seguint aquest protocol es poden enviar un o més objectes, texts en diferents alfabetes, objectes grans entre altres coses.

Diferents maneres de fer el encoding:

- Quoted-Printable
- Base64
- Binary
- 7Bit
- 8Bit
- A la capçalera en format ascii

2.2.3 Retrieval Protocols

- Post Office Protocol (POP)
 - Utilitza el port 110
 - L'usuari normalment elimina els missatges quan els rep
- Internet Message Access Protocol (IMAP)
 - Utilitza el port 143
 - Els missatges es mantenen al servidor fins que l'usuari decideix borrarlos.
 - Proporcionan comandes per crear directoris, moure missatges, baixar algunes parts dels missatges.
- Web based Email (HTTP)
 - Un servidor web proporciona un mailbox. S'utilitza en un navegador i amb HTTP s'envien i es reben els emails.

2.3 Web

URI (Uniform Resource Identifier): sintaxi genèrica per identificar un recurs.

URL (Uniform Resource Locator): subset d'URIs que identifiquen la localització d'un recurs a Internet.

2.3.1 HTTP Messages

- Header: Permet al client donar informació addicional sobre la request i el propi client.
- Methods:
 - GET: comanda típica per demanar un objecte.
 - POST: demana un objecte segons les dades que es posen en el body. Aquestes dades es el contingut dels camps del "qüestionari" HTML.
 - HEAD: el servidor retorna el header
 - OPTIONS: demana opcions de comunicació
 - PUT emmagatzema una entitat
 - PATCH modifica un recurs existent
 - DELETE elimina una entitat
 - TRACE el receptor far un echo del missatge que ha rebut
 - CONNECT: s'utilitza amb un proxy

Els mètodes mñes utilitzats són GET i POST. POST utilitza els tipus MIME per enviar objectes.