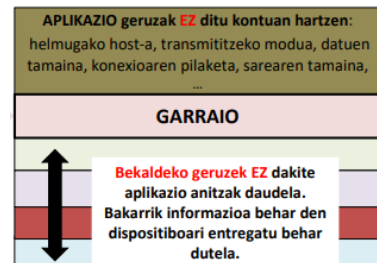


GARRAIO GERUZA

SARRERA

- Aplikazio geruzako datuak onartu eta prestatu sare geruzara bidaltzeko.
 - Paketatu, garraiatu eta entregatu.
 - Bidali host-etik host-era.
- Eskainitako konexioa orientatua edo ez orientatua.



FUNTZIOAK

- Aplikazio 1+ egon daiteke → Aplikazioen multiplexazioa.
 - Helmugako hostean paketeak aplikazio zuzenera bideratu.
 - Garraio geruzaren ardura → Aplikazioetan “komunikazio multiple” mantentzea.
 - Garraio geruzak ID zenbaki bat esleitu aplikazioari.
 - ATAKA ZENBAKI bakarra host horretan dagoen aplikazioarentzat → Goiburuan (header) erabili.
- Protokolo datu unitate batean sartu ahal diren datu kantitate limitea → PDU (Protocol Data Unit).

2 PROZESU

- Segmentazioa (jatorri)
 - Tamaina onartuan zatitu datuak.
 - G. geruzako protokoloek deskribatu zein zerbitzu zatitu.
 - Datu segmentu bakoitzean beharrezko ENKAPSULAZIOA egin.
- Birmuntaketa (helmuga)
 - Aplikazio geruzara bidali baino lehen.
 - G. geruzako protokoloek zehaztu nola birmuntatu header-ko informazioa.

DATUAK HONDATU EDO GUZTIZ GALDU DAITEKE

- Datuen bidaltze fidagarria ziurtatzeko ekintza basikoak:
 - Datuen JARRAIPENA.
 - Jasotako datuen BAIEZTAPENA.
 - Baieztapen gabeko edozein datuaren BIRBIDALTZEA.
- Jatorrizko garraio geruza → Jarraipen + baieztapen gabeko paketearen birbidaltzea.
- Helmugako garraio geruza → Paketeak jasotzen diren ahala jarraitu + jasotzea antzeman + pakete errepikatuak detektatu.

ORDEN OKERREAN IRITSI DAITEZKE

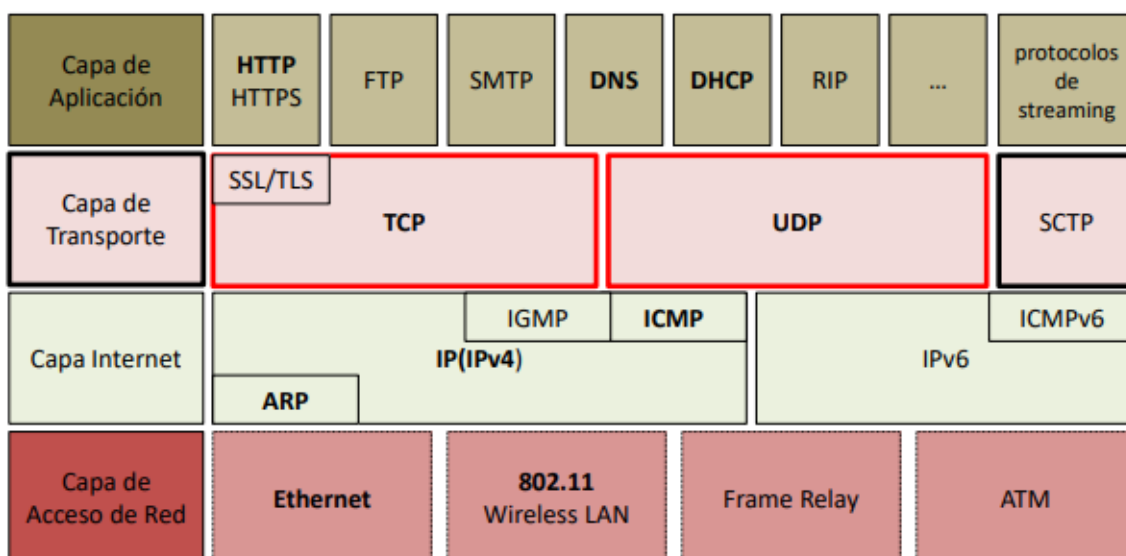
- Sareak transmisio denbora ezberdinak izan ditzake → Bide ezberdinak daude.
- EMATE ORDENATUA
 - Segmentuak zenbakitu eta birmuntaketan ordenatu direla ziurta dezake.

HOST-EK BALIABIDE MUGATUAK (MEMORIA EDO BANDA ZABALERA)

- FLUXU KONTROLA
 - G. geruzak bere baliabideak sobrekargatuak izatekotan → Protokoloen bidez aplikazioari eskatu bidaltze abiadura edo fluxua murriztea.
 - Segmentuen galera saihestu dezake (birbidalketak murriztu).

TCP/IP

- Aplikazio ezberdinak behar ezberdinak dituzte:
 - Segmentuak ordenatuta.
 - Datu guztiak iritsi ondo ulertzeko.
 - Datuen galera onartu.
- Horretarako PROTOKOLO EZBERDINAK



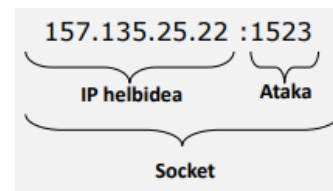
PROTOKOLOAK TCP/UDP

- TCP/IP ereduko garraio geruzaren protokolo garrantzitsuenak dira:
 - TCP (Transmission Control Protocol)
 - UDP (User Datagram Protocol)
- Aplikazio anitzeko komunikazioa kudeatu.
- Funtzio espezifikoetan ezberdintasunak:

TCP	UDP
<ul style="list-style-type: none">• Aplikazioen Multiplexazioa.• Segmentazioa (Datu zatiketa).• Akatsen Kontrola.• Fluxu Kontrola.• Itomen edo Kongestio Kontrola.• Galdutako datuen Birbidaltzea.• Konexio/Deskonexioa.	<ul style="list-style-type: none">• Aplikazioen Multiplexazioa.• Segmentazioa (Datu zatiketa).• Akatsen egiaztapena (Aukerazkoa).

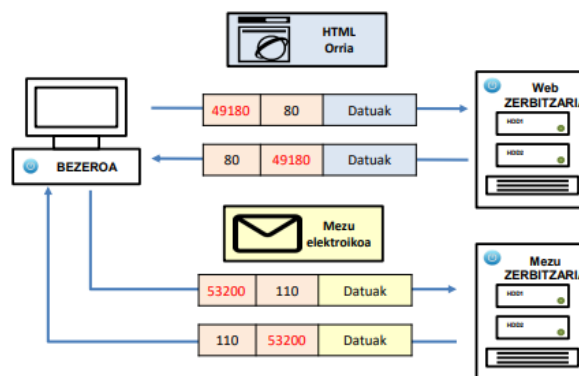
PROTOKOLOAK: APLIKAZIOEN MULTIPLEXAZIOA

- Biak aplikazioaren jarraipena mantendu.
- Aplikazioetara joandako segmentu (TCP) eta datagramak (UDP) bereizteko, hauen goiburuetan aplikazio hauek identifikatzeko informazioa dute.
 - ID horiek ataka zenbakiak dira.
- Multiplexazioa atakaren bidez egiten da (paketearen jatorri eta helmuga app adierazten duen helbide antzekoa).
- Socket-a ataka zenbakia eta sareko geruzako IP helbidearen arteko konbinazioa da. Socket pare bat (jatorri eta helmugakoak) bi host-en arteko elkarrizketa identifikatu.
- Ataka zenbakia: 16 bit (0 eta 65535 balioen artean).
- Atakak 3 tarteetan banatu:
 - **0 - 1023 (well known ports).** Protokolo estandar zerbitzarientzat (HTTP → 80. ataka)
 - Bakarrik supererabiltzaile moduko baimenekin erabili.
 - **1024 - 49151:** ataka ERREGISTRATUAK.
 - Edozein aplikazioak erabili.
 - IANA-ren Web orrian bakoitzak erabiltzen duen protokoloen zerrenda eskuratu.
 - **49152 - 65535:** ataka DINAMIKOAK/PRIBATUAK/IRAGANKOR.
 - Bakarrik bezeroak.
 - Dinamikoki esleitzen dira aplikazio bezero batek konexioa hastean.
- Bezero-zerbitzari eredua erabili → Konexioa bezeroak zein zerbitzariak bukatu.



ATAKA ZENBAKIA ESLEITU

- Esleipen modua aldatu (bezero edo zerbitzari bada).
- **Zerbitzariak ataka ESTATIKOAK esleituta duten bitartean, bezeroak DINAMIKOKI hautatu elkarrizketa bakoitzean.**
- Bezeroak urruneko zerbitzariari dagokion ataka ezagutu.
- Helmugako garraio geruzak JARRAIPENA mantendu, → Erantzun bat itzultzean aplikazio zuzenera eraman. Eskaera egiten duen aplikazioaren portu zenbakia zerbitzariak itzultzen duen erantzunaren helmuga portu zenbaki bezala erabili.



PROTOKOLOAK: SEGMENTAZIOA

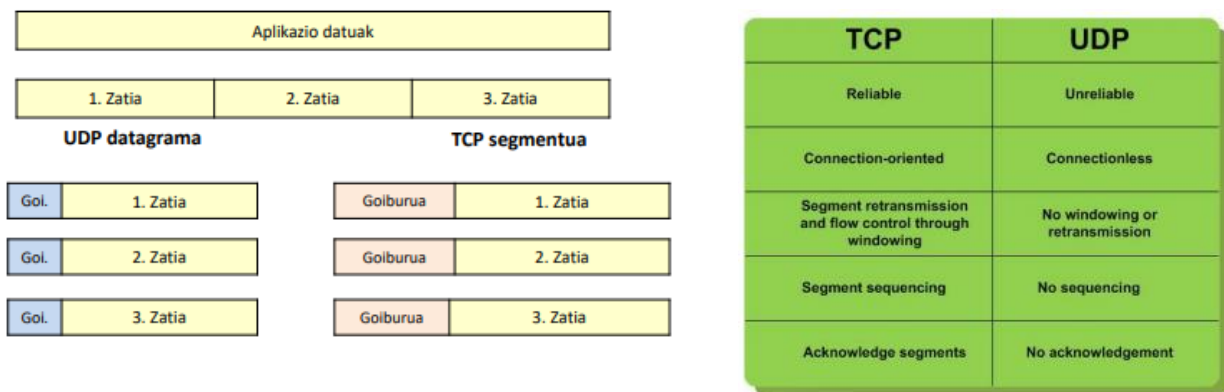
- Aplikazioaren datuak zatietan banatu → Bidearen mugapenen barruan transmititu + Bidean multiplexatu ahal izateko.

TCP

- Honen goiburuan sekuentzia zenbakia → Helmugako garraio geruzan birmuntaketa ordenean egin dezan.

UDP

- Ordena edo konexioa mantentzeaz ez da arduratzen → Sekuentzia zenbakirik ez.
- Honekin funtzionatzen duen aplikazioek ordena ez mantentzea bidalitako datuetan onartu behar du.

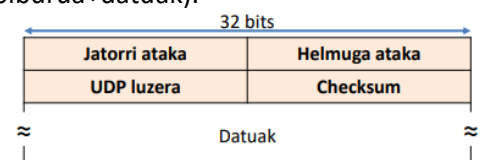


UDP PROTOKOLOA

- Garraiorako oinarrizko ezaugarriak.
- Konexiorik gabeko protokoloa → Aplikazio batek datuak partekatu behar dituen zuzenki egin.
- Erabilitako datu unitateak → DATAGRAMAK.
- Datuen tamaina datagrama bat baino handiago → Datagrama anitzetan zatitu.
- Datagramak ibilbide desberdinak egin + ordenatuta → Ezin ditu helmugan datagramak berrantolatu, jasotako ordenan hartu eta bidali.
 - Ordena aplikaziorako garrantzitsua → Honek kudeatu.
- Ez ditu baliabide asko erabiltzen.
- Datuak "AHALEGIN HOBERENA" ("best-effort") bezala bidali.

UDP DATAGRAMA FORMATUA

- Jatorri ataka → Jatorrizko aplikazio ID (aukerazkoa): 16 bit.
- Helmuga ataka → Helmugako aplikazio ID: 16 bit.
- Luzera → Datagrama osoaren luzera bytetan (goiburua+datuak).
- Checksum → Erroreak kontrolatzeko atala: 16 bit.
- Hautazkoa datu → Aplikazio geruzako datuak.



CHECKSUM

- Ethernet frame formatua (kanpoen tamaina byte-etan). ✓
- IPv4 header-aren formatua. ✓
- UDP header-aren formatua. ✓

Offset: Hexadecimal

```

-----
0: 0260 8ce8 5533 0800 2073 5ec6 0800 4500
16: 0047 adaa 4000 ff11 784f a8b0 0319 a8b0
32: 0132 e573 0035 0033 39f9 0100 0001
48: 0000 0000 0000 0235 3001 3103 3137 3603
64: 3136 3807 696e 2d61 6464 7204 6172 7061
80: 0000 0c00 01
  
```

seudoheader

```

a8b0 0319 ==> Jatorri IP helbidea
a8b0 0132 ==> Helmuga IP helbidea
0011 0033 ==> gero + protokolo zenbakia
(17 hamartar) + UDP luzera (51 hamartar, header eta
datuak barne)
  
```

- Orain batu behar dira, unitatera osagarri (complemento a uno) erabiltzen, pseudo-header + UDP header + UDP datuak, 16 bits-eko hitzak bezala.
 - Operazio batek carry bat sortzen badu bit esanguratsuenean, emaitzari gehitu behar zaio.
 - Aurrekoan checksum zelaia 0x00 bezala hartuz:

[Hex]	Bitar
[e573]	1110010101110011
[0035]	00000000000110101
[0033]	00000000000110011
[0000]	0000000000000000
[E5DB]	1110010111011011

Informazioa:

[Hex]	Bitar
[39f9]	0011100111111001
[0100]	0000000100000000
[0001]	0000000000000001
[0000]	0000000000000000
[0000]	0000000000000000
[0000]	0000000000000000
[0235]	0000001000110101
[3001]	0011000000000001
[3103]	0011000100000011
[3137]	0011000100110111
[3603]	0011011000000011
[1056D]	10000010101101101 -->

0000010101101101 + 1 = 0000010101101110 [56E]

[3136]	0011000100110110
[3807]	00111000000000111
[696e]	0110100101101110
[2d61]	0010110101100001
[1000C]	10000000000001100 -->
00000000000001100 + 1 = 00000000000001101 [D]	
[6464]	0110010001100100
[7204]	0111001000000100
[6172]	0110000101110010
[137DA]	1001101111011010 -->
0011011111011010 + 1 = 0011011111011011 [37DB]	
[7061]	0111000001100001
[0000]	0000000000000000
[0c00]	0000110000000000
[0100]	0000000100000000
[7D61]	0111110101100001

Orain header-aren totalak batzen dira

[Hex]	Bitar
[56E]	0000010101101110
[D]	00000000000001101
[37DB]	0011011111011011
[7D61]	0111110101100001
[BAB7]	1011101010110111

- Checksum-a kalkulatzeko guztiak batu eta bateko osagarrian ipini:

[55F0] 0101010111110000
 [E5DB] 1110010111011011
 [BAB7] 1011101010110111
 [1F682] 11111011010000010 -->

1111011010000010 + 1 = 1111011010000011 [F683]
 0000100101111100 097C

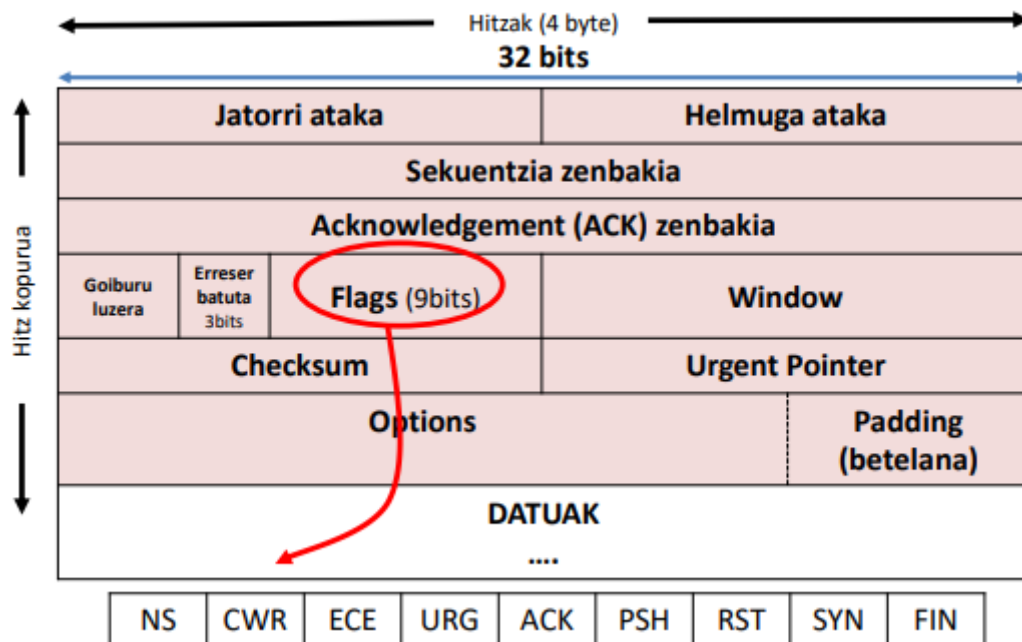
TCP PROTOKOLOA

- UDP baino fidagarriagoa eta konplexuagoa.
- Konexiora zuzenduta.
- TCP paketeak → SEGMENTUAK (informazioa zatitzen da sarean bidaiatzeko).
- TCP gainontzeko baliabideak erabiltzen ditu hurrengo funtzioak betetzeko:
 - Emate fidagarria.
 - Sekuentziaren birmuntatze ordenatuta.
 - Fluxua kontrolatu.
 - Etab.
- TCP segmentu bakoitzean datuak enkapsulatzean goiburuan 20 byte agertu.
- Web arakatzailak, posta elektronikoa, fitxategi transferentzia...

TCP FIDAGARRITASUNA

- Konexiora orientatutako saioak: Transmisio prozesua baino lehen saioa ezarri.
- Jasotako segmentuen baieztapenak egin.
- Datuen birbidaltzea, segmentuak ez badira jaso edota galdutzat eman badira.
 - Saioen ezarpena gehiegizko segmentuen trukaketa + baieztapenak + birbidaltzeak direla eta karga handia sortu sarean.
 - Host-etan ere gainezko karga sortzen da baieztapen zain dauden segmentuen erregistroa mantendu behar delako + birbidaltze prozesuagatik.

TCP GOIBURUA



FLAG-AK

FLAG (1 bit)	ESANAHI
NS	Nonce Sum. Aukerazkoa → Hartzaileak igorlearen segmentuak ezagutzen dituela erakusteko aukera eman.
CWR	Congestion Window Reduced. TCE segmentu bat jaso ECE bandarekin jarrita eta pilaketak kontrolatzeko mekanismoarekin erantzun duela adierazteko.
ECE	Explicit Congestion Notification. Pilaketen jakinarazpena.
URG	Urgent. Premiazko datu bloke bat definitzeko. Erakuslearen eremua baliozkoa dela adierazi → Premiazko datuak non amaitu adierazi erakusleak.
ACK	Acknowledgement.
PSH	Push. Hartzaileak datuak ahalik eta azkarren pasatu behar app-ri datu gehiago itxaron gabe.
RST	Reset. Errorea → Konexioa itxi.
SYN	Synchronise. Sinkronizatu sekuentzia zenbakiak konexioa hasteko. Konexio hasiera adierazi.
FIN	Eskatu konexio amaiera.

TCP IZENBURUA

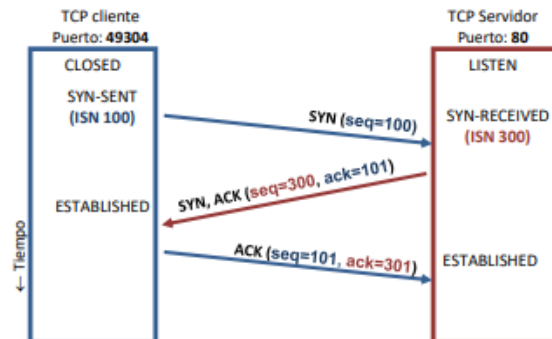
- Aukeren eremua (Options) hiru zati:
 - Aukeraren eremuaren luzera.
 - Erabiltzen den aukera.
 - Erabiltzen diren aukerak.
 - Aukeretako bat MSS da (segmentu gehieneko tamaina).

Kind	Length	Meaning	Reference
0	-	End of Option List	[RFC793]
1	-	No-Operation	[RFC793]
2	4	Maximum Segment Size	[RFC793]
3	3	Window Scale	[RFC7323]
4	2	SACK Permitted	[RFC2018]
5	N	SACK	[RFC2018]
6	6	Echo (obsoleted by option 8)	[RFC1072][RFC6247]
7	6	Echo Reply (obsoleted by option 8)	[RFC1072][RFC6247]
8	10	Timestamps	[RFC7323]
9	2	Partial Order Connection Permitted (obsolete)	[RFC1693][RFC6247]
10	3	Partial Order Service Profile (obsolete)	[RFC1693][RFC6247]
11		CC (obsolete)	[RFC1644][RFC6247]
12		CC.NEW (obsolete)	[RFC1644][RFC6247]
13		CC.ECHO (obsolete)	[RFC1644][RFC6247]
14	3	TCP Alternate Checksum Request (obsolete)	[RFC1146][RFC6247]
15	N	TCP Alternate Checksum Data (obsolete)	[RFC1146][RFC6247]
16		Skeeter	[Stev_Knowles]
17		Bubba	[Stev_Knowles]
---	---	---	---

TCP KONEXIOA EZARRI (3 WAY HANDSHAKE)

- Konexioa ezartzeko TCP protokoloan gehien erabiltzen den mekanismoa → 3 mezuen trukaketan oinarrituta.
 - Bezeroak zerbitzariari konektatzeko gonbidapena → 'Active open' (SYN flag = 1).
 - Gonbidapena jasotzean onartze mezu bat bidali zerbitzariak → 'Passive open' (SYN flag = 1 eta ACK = 1).
 - Erantzuna jasotzean bezeroak konexioa ezarritzat hartu eta hirugarren mezu bat bidali, aurrekoaren baieztapen (ACK = 1).

- TCP konexioa ezartzean ID bat aukeratzen da (ISN - Initial Sequence Number).
- 2 nodo parte hartu. Bakoitzak bere kabuz konexiorako erabiliko duen ISN-a aukeratu, beraz konexio baterako 2 ISN daude.
- ISN-a aurretik ezarritako konexio baten atzeratutako bikoizturen bat iristea ekiditu, honek konexio engainakorra gauzatuko luke.

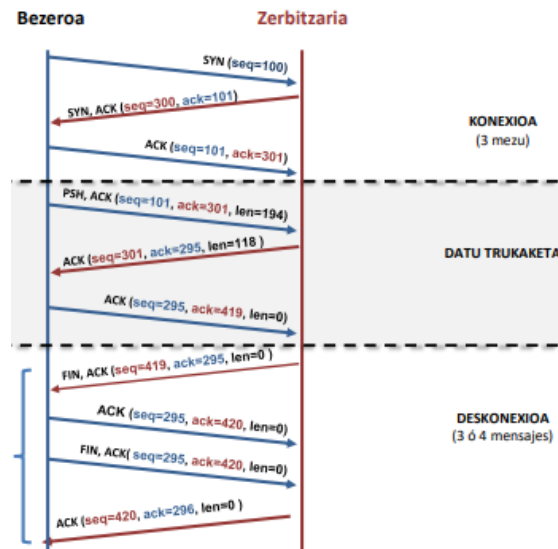


ISN HAUTAKETA

- 32 bit-eko zenbaki oso bat, zeinurik gabekoa eta 4 mikrosegundora unitate batean handitu.
- ISN berdina 4 orduetara agertu daiteke (ISN berdina duen edozein atzeratutako bikoizturen agerpena ekiditeko).
- Sistema eragileak algoritmo errazak erabili hauek eraikitzeke, CPU gutxiago kontsumitzeko.

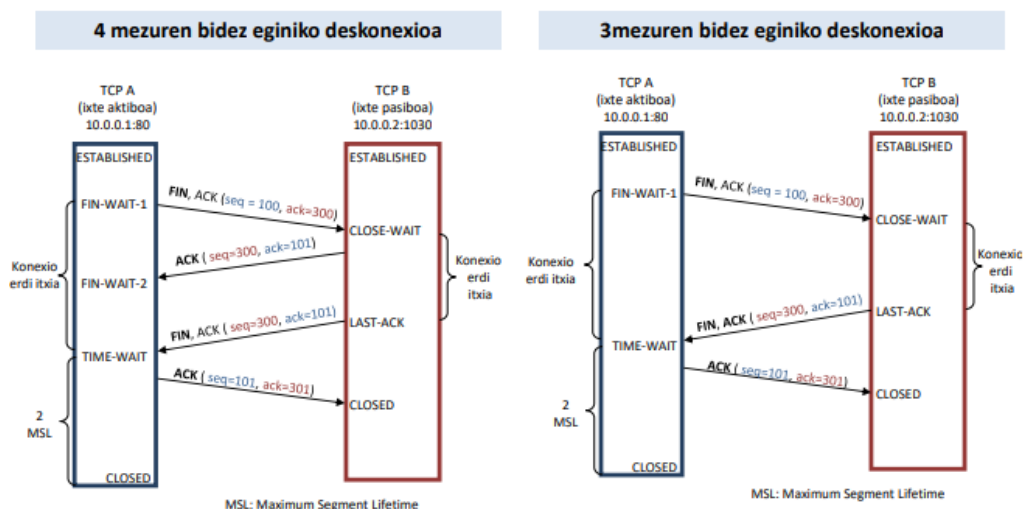
TCP DATU TRUKAKETA

- Zekuentzia zenbakiak eta SYN flag:
 - Konexioan transmititutako byte guztiak zenbatu.
 - SEKUENTZIA ZENBAKIA (SEQ) atala bidalitako datuen lehenengo byte-a identifikatu. Ez ditu segmentuak zenbatzen, byte-ak baizik.
 - ISN TCP protokoloak aukeratzen du SYN bidaltzen duenean, konexioaren ID bezala erabili hiru-bideko agurra konexioan.
 - Sekuentzia zenbakia datuen lehenengo byte identifikatu.
 - Flag SYN jarrita dagoenean sekuentzia zenbakia unitate 1-ean handitu.
 - SYN segmentua datu "birtual" bat bezala ikusi dezakegu.
- Hartze-agiriak eta ACK flag:
 - Baieztapenak beste TCP-ri datuak zuzen jaso direla adierazteko.
 - Flag ACK zentzuduna dela adierazi → Hurrengo TCP segmentuan jasotzea espero den lehen byte-a zehaztu. Hau da, ACK -1 jaso eta antzemandako azken byte-ren zenbakia adierazi.
 - TCP-k piggybacking teknika erabili datuen antzematea egiteko.
 - Baieztapen bat trama bakar batean bidali ordez (ACK mezua soilik) bidaliko den hurrengo paketearen amaieran jarriko du informazioa.
 - Mezu guztiek, lehenengoa izan ezik, ACK Flag jarrita dute.
 - ACK banderaren presentziak EZ du sekuentzia kopurua handitzen.



TCP DESKONEXIOA

- Bezeroa zein zerbitzariak konexioa eten edo itxi dezakete.
- 4 mezu trukaketaren bidezko prozesua.
- Bezeroa deskonexioa hasten badu:
 - FIN flag aktibatuta duen segmentu bat bidali.
 - Zerbitzariak ACK bidali FIN hori jaso duela adierazteko.
 - Zerbitzariak datu gehiago ez duenean bidaltzeko, FIN bat bidali saioa bukatzeko.
 - Bezeroak ACK batekin erantzun zerbitzariaren FIN jaso duela adierazteko.
- Hiru-bideko lotura baten bidez konexioa bukatu:
 - Bezeroak FIN bat bidali.
 - Zerbitzariak datu gehiago ez baditu FIN eta ACK-rekin erantzungo dio biak pakete bakarrean konbinatuz.
 - Bezeroak ACK batekin erantzungo dio.



TCP SEGMENTU REERAIKIERA

- Datu segmentuak BIRMUNTATU hasierako ordenean.
- Horretarako SEKUENTZIA ZENBAKIA (SEQ):
 - Saioaren konfigurazioan ISN bat ezarri.
 - Honek saio horretan bidaliko diren byten hasiera adierazi.
 - Saioan zehar datuak transmititu ahala SEKUENTZIA ZENBAKIA TRANSMITITU eta jakinarazi DIREN BYTETan INKREMENTATU.
 - Byten jarraipen hori SEGMENTU BAKOITZA IDENTIFIKATU ETA BERE BAIEZTAPENA (ACK) BIDALTZEA modu bakar batean egitea posible egin → Galdutako segmentuak identifikatu.
- SEQ segmentua nola birmuntatu eta nola ordenatu behar diren adierazi.
- TCP hartzaileak datu segmentuak hartze buffer batean sartu. Segmentuak bere SEQ arabera ordena zuzenean jarri eta aplikazio mailara pasatu birmuntatuta.

TCP RETRANSMISIOA

- Jatorri edo helmuga buffera beteta → Segmentuak galdu.
- Segmentu galduak kudeatzeko → Baieztapena ez duten segmentuak birbidali.
- Jatorri hosteko TCP denboraldi batean baieztapen bat jasotzen ez duenean, jaso duen azkeneko hartze-agiria itzuli eta puntu hortik aurrera berriro bidali.
- Host-ak “hartze agiri selektiboa” deritzon hautazko funtzio bat ere erabili dezakete.
 - Hau onartzen badute, hartzaileak segmentu ez jarraituen byteak antzeman ditzake eta orduan bakarrik galdutako datuak birbidali.

TCP FLUXU KONTROLA

- Helmuga app-ren datuak irakurtzeko abiadura igorlearen transmisio abiadura baino txikiagoa → Harrera buffera saturatu eta segmentuak gal ditzake.
- TCP Goiburuko WINDOW elementua baieztapena jaso baino lehen transmititu daitekeen datu kopurua ezarri.
 - Hasierako leihoaren tamaina saioaren hasieran ezarri hiru-bideko agurran.
 - Adibidea: Hasierako tamaina 3000 byte. Igorleak 3000 byte bidali eta baieztapen zain. Behin hau jasota beste 3000 byte bidali ditzake.
- LEIHOAREN NEURRI DINAMIKOAK:
 - Helmugak komunikazio abiadura moteldu behar badu memoria buffer mugatua duelako, leihoaren tainaren balio txikiagoa bidal diezaioke iturriari jakinarazpen batean.
 - Transmisioaren abiadura modu eraginkorrean murriztu, iturriak datuak maizago aitortzea espero duelako.
 - Datu galerarik edo baliabide murriztik gabeko transmisio aldien ondoren, helmugak leihoaren tamaina handitu dezake.
 - Aitorpen gutxiago eskatzeak sareko gainkarga murriztu.
 - Leihoaren tamaina handitzen datuak galdu arte, eta horrek leihoaren tamaina txikitu.

Host 1

Host 2

Aplikazioak idazten du 1 KB

Byte 999 arte ondo jaso ditut. Bidalazen didazun hurrengoa 1000 izan beharko luke.

Aplikazioak idazten du 4 KB

Adibide hauetan 1000 byteko segmentuak hartu dira. Zenbakia segmentu bakoitzaren posizio erlatiboa adierazten du.

Blokeatuta

Ack=0, Win=4000

Seq=0 0-999

Ack=1000, Win=3000

Seq=1000 1000-1999

Seq=2000 2000-2999

Seq=3000 3000-3999

Ack=4000, Win=0

Ack=4000, Win=2000

Seq=4000 4000-4999

Ack=5000, Win=3000

4 KB libre

3 KB libre

2 KB libre

1 KB libre

Buffer betea

Aplikazioak irakurtzen du 2 KB

2 KB libre

1 KB libre

Aplikazioak irakurtzen du 2 KB

3 KB libre