ISO-OSI medel

Datagr.	**ISO-OSI**	**TCP-IP**	*prot.*
podatki	aplikacijska	aplikacijska	HTTP, DNS, POP3, SSH
podatki	predstavitvena		
podatki	sejna		
segmenti	prenosna	transportna	TCP, UDP
paketi	omrežna	internetna	IP, ICMP
okvirji	povezavna	povezovalna	MAC, PPP
biti	fizična		

Fizična plast

Naloge:

- **kodiranje bitov** z neko fizikalno veličino (*signalom*) za prenos po mediju (*baker, optika, radijski, IR, ...*)
- **prenos posameznih bitov** v analogni ali digitalni obliki
- **prenos celotnega signala** (zaporedja bitov po mediju)
- **pretvorba signala** v obliko, ki je primerna za prenos po mediju

Modulacija

- *amplitudna* (1: velika amplituda, 0: majhna ampituda)
- *frekvenčna* (1: visoka frekvenca, 0: nizka frekvenca)
- *fazna* (1: zakodiramo z fazo 0°, 0: z fazo 180°)
- *kvadratna*
  - kombinacija amplitudne in fazne modulacije
  - 4 fazni koti 0°, 90°, 180° 270°
  - 2 nivoja amplitude
  - 8 kombinacij faze in amplitude → 3 bite
  - lahko imamo tudi več faznih kotov in amplitud in naenkrat zakodiramo do 6 bitov

Povezavna plast

Lahko (ne pa njuno) opravlja naloge: okvirajanje datagramov, zaznavanje in odpravljanje napak, dostop do medija, zagotavljanje zanesljive dostave, kontrola pretoka.

Okvirjanje datagramov

Okvir je *enota* na povezavni plasti. Določa začetek in konec prenesenih podatkov.

Zaznavanje in odpravljanje napak

dodamo bite za preverjanje pravilnosti (EDC - Error Detection Code)

- **soda/liha pariteta** vseh 1 skupaj s pariteto je sodo/liho *(omogoča le zaznavanje lihega števila napak)*
- **dvo dimenzionalna pariteta** dodamo praitetni bit za vsako vrstico in vsak stolpec + kumulativni/meta paritetni bit je pariteta paritetnih bitiv *(omogoča zaznavanje in odpravljanje enojnih napak in zaznavanje dvojnih napak)*
- **Hammingova koda**
  Paritetni biti p imajo indekse 2^i in jih postavimo na mesta, ki jih določajo indeksi. Vmes vstavimo bite podatkov d.
  Vsak paritetni bit p_i pokriva i bitov od svoje pozicije naprej, nato preskoči i bitiov in tako naprej. Število 1, ki jih paritetni bit pokriva mora biti sodo.

	1.	2.	3.	4.	5.	6.	7.
	p_1	p_2	d_1	p_4	d_2	d_3	d_4
p_1	✓		✓		✓		✓
p_2		✓	✓			✓	✓
p_4				✓	✓	✓	✓

  Ko prejmomo sporočio, preverimo paritetne bite in definiramo:

$$p_i' = \begin{cases} 1 & \text{paritetni bit } p_i \text{ napačen} \\ 0 & \text{sicer} \end{cases}$$

  Nato izračunamo sindrom:

  $[\dots p_4' \; p_2' \; p_1']_{(2)} = \#$ bita, ki ga je treba obrniti
- **Cyclic Redundancy Check - CRC** *z r paritetnimi biti lahko popravi do $r+1$ napak*
- **Internet checksum** *uporablja se na transportni plasti*

Dostop do skupinskega medija

Imamo dve vrsti povezav: **dvotočkovna** (le en pošiljatel in prejemnik - PPP, DHLC) in **oddajna** (več vozlišč komunicira naenkrat - Ethernet, Wireless LAN).

Kanal lahko delimo na več načinov:

Delitev kanala

Time Division Multiple Access TDMA, Frequency Devision Multiple Access FDMA *Ni kolizij*

Sočasni/naključni dostop (kolizijski)

- **ALOHA** Če pride do kolizije, se paket po naključnem času ponovno pošlje.
- **razsekana ALOHA** Čas je razsekan na časovne intervale. V vsakem intervalu se lahko pošlje en paket. Če pride do kolizije, se v nasldnjem intervalu pošlje ponovno z verjetnostjo p.
- **Carier Sense Multiple Access - CSMA** Poslušaj ali že kdo oddaja preden začeneš oddajati sam.
- **Carier Sense Multiple Accerss Collision Detection - CSMA/CD** Če zaznaš kolizijo nehaj oddajati in pošlji motilni siganal.

Izmenični dostop

- **Rezervacija s centralnim vozliščim (pooling)**: centralno vozlišče zaporedno dodeluje medij posameznim vozliščem (*zakasnitev, ena točka odpovedi*)
- **Rezervacija z žetonom, ki kroži (token)** (*zakasnitev, ena točka odpovedi - žeton*)

Dostop do brezžičnega medija

Protokol 802.11 (WiFi) za dostop do medija uporablja **Carier Sense Multiple Accerss Collision Avoidance - CSMA-CA**.

Uporablja signala:

- **Request To Send - RTS**
- **Clear To Send - CTS**

Terminal, ki želi pošiljati najprej pošlje RTS in čaka na CTS. Ko terminal prejme CTS, ne bo oddajal toliko časa kot traja, da se prenese en paket.

Protokoli na povezavni plasti

Ethernet

Okvir ethernet:

preambula	*ponorni naslov*	*izvorni naslov*	*tip*	*podatki*	*CRC*
8B	6B	6B	2B	1500B	4B

Ethernet ponuja:

- nepovezavna storitev
- nezanesljiva storitev (*kontrola pravilnosti se izvaja s CRC a brez popravljanja, potrjevanje in ponovno pošiljanje se ne uporabljata*)
- uporablja CSMA/CD (*pri vsaki koliziji se čakanje eksponentno poveča* $0-1, 0-3, 0-7, 0-15, \dots$)

Point to Point Protocol - PPP

Povezavni protokol; imamo le enega pošiljatelja in prejemnika.

Okvir PPP:

zastavica za začetek	*address*	*control*	*protokol*	*podatki*	*CRC*	*zastavica za konec*
1B	1B	1B	1B/2B	$\leq$1500B	2B/4B	1B

Poja "address" in "control" se ne uporabljata.

Pred niz `01111110` (zastavica) urivamo *ubežno kodo* `01111101`.

Naslavljanje naprav na povezavni plasti

Vsaka naprava ima fizični naslov (Media Access Control - MAC) iz 48 bitov.

Naslov `FF-FF-FF-FF-FF-FF` je **brodcast** (prejemniki so vsi).

Address Resolution Protocol - ARP

Vsaka naprava ima ARP tabelo [IP | MAC | TTL]

Če naprava želi ugotoviti MAC druge naprave, pošlje **ARP poizvedbo** (*Moj max je y, kdo ima IP x?*) na brodcast naslov. Naparava, ki ima ta IP odgovori z **ARP odgovorom** (*Jaz imam IP x in oj MAC je z.*). Nato si obe napravi shranita MAC naslove v ARP tabelo.

Stikalo

Stikalo uporablja **stikalno tabelo** [MAC | vrata | TTL], da se odloči, na katera vrata poslati okvir. Ko prejme kak okvir, si zapomni lokacijo pošiljatelja in okvir:

- **poplavi** na vsa vrata razen na tista iz katerih je prišel okvir (*če ne ve, na katerih vratih je prejemnik*)
- **posreduje** na izbrana vrata (*če ve, na katerih vratih je prejemnik*)
- **filtriria** / zavrže okvir (*če je namenjen istim vratom iz katerih je prišel*)

Omrežna plast

Lahko omogoča:

1. zagotovljena dostava paketov
2. dostava v zagotovljenem času
3. dostava v pravem vrstnem redu
4. zagotovljena spodnja meja pasovne širine
5. največja dovoljena varianca zakasnitve (jitter)
6. varno komunikacijo (zaupnost integriteto podatkov, avtentikacijo)

Storitve interneta

		zagotovljene storitve				
Omrežje	Model	pas. širina	brez izgub	vr. red	čas	obv. o zamaš.
Internet	best effort	×	×	×	×	×izguba
ATM	CBR constant bit rate	const	✓	✓	✓	ni zamaš.
ATM	ABR available bit rate	min	×	✓	×	✓

Usmerjevalnik

Naprava, ki deluje na omrežni plasti in skrbi za transport datagrama po *jedru omrežja*. Povezuje različne medije in portokole.

Funkcije usmerjevalnika:

- **Posredovanje paketov**/forwarding: prenos na pravi izhodni vmesnik glede na destinacijo paketa in **posredovalno tabelo**.
- **Usmerjanje**/routing: določanje poti paketov od izvora do cilja. Usmerjevalniki s poočjo usmerjevalnih protokolov najdejo najcenejšo pot.

Povezavna omrežja / navidiezni vodi

Faze pri vzpostavitvi navideznega voda: *vzpostavitev*, *tok podatkov*, *rušenje*

Usmerjevalniki usmerjajo pakete glede na *št. voda* v paketu in posredovalno tabelo [vhodni vmesnik | vhodna št. voda | izhodni vmesnik | izhodna št. voda].

Nepovezavna / datagramska omrežja

Ni faze vspostavljanja povezave. Paket lahko do cilja pride po različnih poteh.

Usmerjevalniki usmerjajo glede na ciljni (IP) naslov in usmerjevalno tabelo [predpona naslova | vmesnik].

Internet Protocol - IP

Paket IPv4

Dolžina glave brez "možnosti" je 20B.

4b	4b	4b	4b	4b	4b	4b	4b
Verzija	*Dolžina glave*	*Tip storitve*		*Skupna dolžina*			
Identifikacija				*Zastav.*	*Zamik fragmenta*		
TTL		*Protokol*		*Kontrolna vsota*			
Izvorni IP naslov 32b							
Ponorni IP naslov 32b							
Možnosti							*Zpolnjevanje*
Podatki							

Fragmentacija

IP datagram se mora enkapsulirati v omejene okvirje (MTU) povezavne plasti. Zato se IP paket razbije na več manjših. Da fragmente lahko sestavimo nazaj se uporabljajo polja v glavi:

- **ID**: pri vseh fragmentih enak
- **Zastavica MF**: 0, če je to zadnji fragment; 1, sicer
- **Offset**: enota za odmik je 8 Byte

Podomrežja

Razred	Začetek naslova	Maska
A	`0 xxxxxxx`	`255.0.0.0`
B	`10 xxxxxx`	`255.255.0.0`
C	`110 xxxxx`	`255.255.255.0`
D	`1110 xxxx`	-
E	`11110 xxx`	-

Posebni naslovi

- Privatni (24 bitni) `10.0.0.0/8`
- Privatni (20 bitni) `172.16.0.0/12`
- Privatni (16 bitni) `192.168.0.0/16`
- Povratna zanka (localhost) `127.0.0.0/8`
- Link-local `169.254.0.0/16`

Posebni naslovi v lokalnem omrežju

- Naslov omrežja: *1. naslov v omrežju (same ničle)*
- Hosts: *vsi naslovi vmes*
- Brodcast: *zadnji naslov v omrežju (same enke)*

Dinamično dodeljevanje naslovov - DHCP

Ko se naprava priključi v omrežje, pošlje **DHCP discover**, DHCP strežnik(i) odgovori(jo) z **DHCP offer**, naprava izbere IP iz ponudbe in pošlje **DHCP request**, končno strežnik potrdi z **DHCP ACK**.

Internet Control Message Protocol - ICMP

Sporočilo je enkapsulirano v paketu IP. Uporablja se za sporočanje napake, nedosegljivosti, protokola, vrat, ...

Traceroute pošlje vrsto paketov z TTL=1,2,3,... vsakič ko paketu poteče TTL router pošlje nazaj ICMP sporočilo.

Network Address Translation - NAT

Lokalne naslove slika v globalne in obratno.

To doseže z NAT preslikovalno tabelo [globalni naslov, port | zasebni naslov, port].

Vsak zahtevek ima izvorni in ponorni port/vrata.

IPv6

Naslovi so razdeljeni (:) na 8 blokov po 16 bitov. Zaporedje blokov samih ničel lahko krajšamo z "::",

Dolžina glave je vedno 40B.

4b	4b	4b	4b	4b	4b	4b	4b
Verzija	*Traffic class*		*Flow label*				
Payload len				*Next header*		*Hop limit*	
Izvorni IP naslov 128b							
Ponorni IP naslov 128b							
Podatki							

Novo polje "flow label" (vrsta toka) za zagotavljanje kakovosti storitev za posebne tokove podatkov.

Prehodni mehanizmi

- **Dvojni slkad**: Vse naprave imajo IPv4 in IPv6. Če je na poti kakšno IPv4 vozlišče, se bo proromet vmes pretvarjal v IPv4.
- **Tuneliranje**: Če je na poti kakpno IPv4 vozlišče se IPv6 paket enkapsulira v IPv4.

Usmerjanje

Usmerjevalni algoritmi so lahko:

- **centralizirani** (*imajo podatke o celem omrežju*) ali **decentralizirani** (*imajo podatke le o sosednjih vozliščih*)
- **prilagodljivi** (*lahko prilagajajo cene povezav glede na zasičenost*) ali **naprilagodljivi**

Decentralizirano usmerjanje

Vsako vozlišče pozna ceno povezave do svojih sosedov. Hrani **vektor razdalj**, **vektorje razdalj svojih sosedov** in **posredovalno tabelo** [cilj | cena | via sosed].

Ko vozlišče prejme vkektor razdalj svojega soseda, izračuna nov vektor razdalj *če lahko do nekega vozlišča pride ceneje prek nekega soseda, posodobi posredovalno tabelo.*

Usmerjevalniki so povezani v **avtonomne sisteme - AS** v katerih uporabljajo isti **INTRA-AS** usmerjevalni protokol.

- **Routing Information Protocol - RIP** usmerjanje z vekotrjem razdalj (na 30s, 180s+ → prekinjena povezava), cena je število skokov, max cena je 15M
- **Open Shortest Path First - OSPF** obvestila se s poplavljanjem posredujejo po celem sistemu, vsak potem preračuna najkrajše poti
- **Interior Gateway Routing Protocol - IGRP** Cisco-va izboljšava RIP; cena je odvisna od pasovne širine, zakasnitve, obremenitve, MTU in zanesljivosti.

Za povezovanje AS mad seboj se uporablja **INTER-AS** usmerjevalni protokol, ki pa morabiti enak v celem omrežju.

- **Border Gateway Protocol - BGP** omogoča, da omrežja oglašujejo svojo prisotnost drugim omrežjem

Transportna plast

Naloge transportne plasti so:

- povezovanje dveh oddaljenih *procesov*
- multipleksiranje/demultipleksiranje komunikacije med procesi
- zanesljiv prenos podatkov
- kontrola pretoka in zasičenja

Vrata - port

FTP-data: 20, FTP-cmd: 21, SSH: 22, Telnet: 23, SMTP: 25, DNS: 53, HTTP: 80, POP3: 110, IMAP: 143, IRC: 194, HTTPS: 443

User Datagram Protocol: UDP

Nepovezavna storitev, nudi le best-effort: ni zagotavljanja vrstnega reda, ni nadzora zamašitev, ...

Dolžina glave je le 8B.

16b	16b
Izvorna vrata	*Ponorna vrata*
Dolžina (skupaj z glavo) v B	*Internetna kontrolna vsota*

Internetna kontrolna vsota

Pošiljatelj: pošiljatelj sešteje 16 bitne besede in shrani eniški komplement = kontrolna vsota
Prejemnik: sešteje 16 bitne besede skupaj s kontrolno vsoto → dobiti mora same enice

Potrjevanje

sprotno potrjevanje: po vsakem pošiljanju počakaj na potrditev

tekoče pošiljanje: večji razpon za številčenje paketov, shranjevanje paketov na obeh straneh

- ponavljanje N potrjenih (*go-back-N*)
  - pošiljatelj hrani okno največ dovoljenih nepotrjenih paketov
  - ACK(n) potrdi vse pakete do vključno n
  - časovna kontrola za najstarejši paket, ko poteče pošlji vse nepotrjene pakete na novo
- ponavljanje izbranih (*selective repeat*)
  - prejemnik shranjuje sporočila in jih sortira pred dostavo
  - pošiljatel ponovno pošlje le tiste pakete za katere ni prejel potrdila
  - vsak paket ima svojo časovno kontrolo

neposredno: uporaba ACK in NCK **posredno**: samo ACK, namesto NCK se ponovi ACK zadnjega segmenta

Transfer Control Protocol: TCP

Povezavna storitev, full duplex, zanesljiv, kontrola pretoka, kontrola zasičenja, *tekoče pošiljanje in neposredno potrjevanje*

<table>
<tr><td colspan="3">16b</td><td colspan="3">16b</td></tr>
<tr><td colspan="3">Izvorna vrata</td><td colspan="3">Ponorna vrata</td></tr>
<tr><td colspan="3">Dolžina (skupaj z glavo) v B</td><td colspan="3">Internetna kontrolna vsota</td></tr>
<tr><td colspan="6">zaporedna št. (B)</td></tr>
<tr><td colspan="6">št. potrditve (B)</td></tr>
<tr><td>Dolzina glave</td><td>0</td><td>zastavice</td><td colspan="3">sprejemno okno</td></tr>
<tr><td colspan="3">Internetna kontrolna vsota</td><td colspan="3">karalec urg. data</td></tr>
<tr><td colspan="6">opcije (spremenljiva dolžina)</td></tr>
</table>

Dolžina glave (4b) je podana v številu 32-bitnih besede

Sprejemno okno (16b) št. bajtov, ki jih sprejemnik lahko sprejme

Zastavice (9b):

- URG*: urgentni podatki
- ACK: potrditev povezave
- PSH: takoj predaj aplikaciji
- RST, SYN, FIN: vzpostavljanje in rušenje povezave

Vzpostavitev povezave - trojno rokovanje

1. Odjemalec pošlje segment z zastavico `SYN` (sporoči začetno številko segmenta, ni podatkov)
2. Strežnik vrne segment `SYN ACK` (rezervira medpomnilnik, odgovori z začetno številko svojega segmenta)
3. Odjemalec vrne `ACK`, lahko že s podatki ("štuporama")

Rušenje povezave

1. odjemalec pošlje segment `TCP FIN` strežniku
2. strežnik potrdi z `ACK`, zapre povezavo, pošlje `FIN`
3. odjemalec prejme strežnikov `FIN`, potrdi ga z `ACK` (počaka časovni interval, da po potrebi ponovno pošlje `ACK`, če se ta izgubi)
4. strežnik sprejme `ACK`, končano

Nastavitev časovne kontrole

Na potrditev moramo čakati vsaj `RTT` (Round Trip Time).

$$\texttt{OcenRTT}[i] = (1-\alpha)\texttt{OcenRTT}[i-1] + \alpha\texttt{IzmerRTT}[i]$$

$$\texttt{DevRTT}[i] = (1-\beta)\texttt{DevRTT}[i-1] + \beta\left|\texttt{IzmerRTT}[i] - \texttt{OcenRTT}[i]\right|$$

$$\texttt{ČakalniInterval}[i] = \texttt{OcenRTT}[i] + 4\texttt{DevRTT}[i]$$

Način potrjevanja

TCP uporablja podobno strategijo kot *ponavljanje N nepotrjenih*. Pošiljatelj ima časovno kontrolo le za najstarejši nepotrjen segment, vendar ob poteku časovne kontrole ne pošlje vseh segmentov v oknu, temveč le najstarejši nepotrjen segment.

RFC2018 vpeljuje potrjevanje le izbranih paketov.

Dogodek pri prejemniku	*Odziv prejemnika*
Sprejem segmenta s pričakovano številko, vsi prejšnji že potrjeni.	Počakaj na naslednji segment max 500 ms. Če ta pride v tem intervalu, izvedi zakasnjeno potrditev obeh (*delayed ACK*). Če ne pride v tem intervalu, potrdi samo prejetega.
Isto kot zgoraj, a potrditev za prejšnji segment še ni bila poslana.	Takoj pošlji *kumulativno potrditev* za oba segmenta brez izvajanja zakasnjene potrditve.
Sprejem segmenta s previsoko številko (*zaznamo vrzel*)	Takoj potrdi zadnji še sprejeti segment (*pošlji duplikat ACK*).
Sprejem segmenta z najnižjo številko iz vrzeli (*polnjenje vrzeli*)	Takoj potrdi segment.

Hitro ponovno pošiljanje (fast retransmit)

Ponovno pošiljanje se običajno izvede po preteku časovne kontrole. Če pa pošiljatel prejme 3 potrditve za že potrjen paket, takoj izvede ponovno pošiljanje.

Kontrola pretoka

Usklajevanje hitrosti pošiljanja med pošiljateljem in prejemnikom.

Prejemnik sporoča razpoložljiv prostor v pomnilniku v polju 'sprejemno okno':

$$\texttt{rwnd} = \texttt{velikost} - [\texttt{lastByteRcvd} - \texttt{lastByteRead}]$$

Oznake

`rwnd`	...	recieve window
`cwnd`	...	congestion window
`MSS`	...	maximum segment size
`RTT`	...	round trip time
`ssthresh`	...	slow start threshold

Nadzor zasičenja

min(`rwnd`, `cwnd`) določa največ nepotrjenih B preden moramo ustaviti pošiljanje.

TCP Tahoe: osnovna verzija; 2 fazi: *počasen začetek* in *izogibanje zasičenju*; po izgubi paketa `cwnd` = 1 `MSS`.

TCP Reno: 3 faze: *počasen začetek*, *izogibanje zasičenju* in *hitra obnova*; če dobiš 3 kopije ACK že potrjenih podatkov, `cwnd` = `cwnd`/2 + 3 `MSS`.

TCP Vegas: dodano zaznavanje sitacij, ki vodijo v zasičenje in *linearno zmanjševanje hitrosti* ob zasičenju.

Na začetku: `cwnd` = 1`MSS`; faza počasnega začetka; `ssthresh` = ∞.

- **Faza počasnega začetka:**
  Za vsak prejet ACK: `cwnd` += min(`N`, `MSS`), kjer je `N` št. B, ki jih je potrdil ta ACK.
  Alt. rešitev: Za vsak prejet ACK: `cwnd` += `MSS`.

  Na ta način se skozi čas `cwnd` povečuje eksponentno.

  - Če je `cwnd` $\geq$ `ssthresh`, preidemo v fazo izogibanja zasičenju.
  - Če prejmemo 3 ACK, že potrjenega segmenta, gremo v *fazo hitre obnove* `cwnd` = ssthresh + 3`MSS`, `ssthresh` = `cwnd` / 2

- **Faza izogibanja zasičenju:**
  Števemo B, ki jih potrdijo ACK. Ko to število preseže `cwnd`, resetiramo števec in `cwnd` += MSS.
  Alt. rešitev: Za vsak ACK, ki potrdi nove podatke povečamo `cwnd`+ = `MSS` $\cdot$ `MSS`/`cwnd`

  Na ta način se skozi čas `cwnd` povečuje linearno.

  - Če poteče časovna pontrola za najstarejši segment, gremo v *fazo počasnega začetka* in `cwnd` = MSS, `ssthresh` = `cwnd` / 2.
  - Če prejmemo 3 ACK, že potrjenega segmenta, gremo v *fazo hitre obnove* `cwnd` = ssthresh + 3`MSS`, `ssthresh` = `cwnd` / 2

- **Faza hitre obnove:**
  Za vsak podvojeni ACK: `cwnd` += `MSS`.

  - Če dobimo ACK, ki potrjuje nove podatke, gremo v *fazo izogibanja zasičenju* in `cwnd` += `ssthresh`.
  - Če preteče nek timeout, gremo v *fazo počasnega začetka* in `cwnd` = MSS, `ssthresh` = `cwnd` / 2.

TCP konvergira k pravični delitvi pasovne širine med uporabniki.