# OSI, TCP/IP modeļi

ISO (*Internet Standards Organisation*) izveidoja OSI (*Open System Interconnection*) ir references modelis – veids kā zinātniski raudzīties uz datortīklos notiekošajiem procesiem. Radīts 1984. gadā, ir sintētisks. Tas kategorizē datortīklu procesus un tehnoloģijas 7 līmeņos un to savstarpējās saskarnēs:

1. **Fiziskais (*Physical*)**

Apraksta fiziskās, elektroniskās un funkcionālās metodes, kā tiek uzturēti, izveidoti un deaktivizēti savienojumi bitu pārsūtīšanai. Saņem no kanāla līmeņa kadrus un pārveido tos elektriskā impulsā.

1. **Kanāla (*Data Link*)**

Atbild par saziņu starp tīkla kartēm kopīgā vidē. Saņem no tīkla līmeņa paketes un iekapsulē tās kadros.

1. **Tīkla (*Network*)**

Apraksta metodes loģiskai adresācijai. Ļauj sazināties starp vairākiem tīkliem. Saņem no transporta līmeņa segmentus un iekapsulē tos paketēs.

1. **Transporta (*Transport*)**

Ļauj sadalīt, pārsūtīt un salikt kopā datus individuāliem savienojumiem starp iekārtām. Sadala datus segmentos, ievieš portu adresāciju.

1. **Sesijas (*Session*)**

Atbild par sesiju pārvaldību, palīdzot augstākiem slāņiem ar datu plūsmas organizēšanu.

1. **Prezentācijas (*Presentation*)**

Definē vienotu veidu kā reprezentēt datus, kas tiek sūtīti starp lietojumu līmeņa servisiem.

1. **Lietojumu (*Application*)**

Nodrošina tīkla pakalpojumus programmatūrai.

TCP/IP ir atvērtā standarta un standartā bāzēta protokolu saime, kas reizē ir arī references (konceptuāls) modelis, ko dažkārt sauc arī par interneta modeli. Tas ir senāks par OSI modeli. To izveidoja ASV aizsardzības departaments. Tajā ir 4 slāņi, vairāki no tiem apvieno dažus OSI slāņus:

1. **Tīkla piekļuves (*Network Access*)** – kontrolē fiziskās ierīces, kas veido tīklu.
2. **Interneta (*Internet*)** – nosaka labāko maršrutu pa tīklu. Tiešs analogs OSI tīkla slānim.
3. IPv4 un IPv6 protokols definē paketes kā info vienības, bet nenodrošina to drošu nonākšanu galamērķī. ICMP sūta kontroles ziņas, ARP atrod MAC adresi zināmai IP adresei un RARP atrod IP adresi zināmai MAC adresei.
4. **Transporta (*Transport*)** –tiešs analogs OSI transporta slānim. Nodrošina sūtīšanas integritāti. TCP bez zudumiem, UDP ar zudumiem.
5. **Lietojumu (*Application*)** – reprezentē datus lietotājam. Tajā apvienoti OSI lietojumu, prezentācijas un sesijas slāņi.

Pirms slāņu ieviešanas, datortīklu tehnoloģisko risinājumu savietojamība bija neveikla. Un radās, piemēram, situācijas, kad, izmainot vada struktūru, bija jāraksta jauna programmatūra.

Darbību sadalīšana neatkarīgos slāņos sniedz priekšrocības:

* Atvieglota jaunu protokolu izstrāde, jo katra slāņa protokoliem ir labi definēti dati ar kuriem tie strādā, kā arī interfeiss ar kaimiņu slāņiem.
* Sadala saziņu sīkākās, labāk pārvaldāmās un aprakstāmās daļās;
* Tiek standartizētas datortīkla komponentes, tādējādi atsevišķu komponenšu izstrādē var piedalīties vairāki ražotāji;
* Dažāda veida aparatūra un programmatūras ir savietojamas un izmaiņas vienā slānī nerada neparedzētas izmaiņas citos slāņos.

# Vītā pāra kabeļi

Vītā pāra kabeļi ir apakštips vara kabeļiem, kas ir mūsdienās visbiežāk lietotie tīkla kabeļi. Datu pārraide tajos notiek kā elektriskie impulsi. Varš ir lēts, un tam ir maza pretestība elektriskajai strāvai. Saņēmējam ir detektors, kas saņemto signālu var pārvērst digitālos datos, taču, jo tālāk šie dati ceļo, jo vājāks kļūst signāls.

Vara kabeļos elektriskā impulsa sprieguma vērtības ir jūtīgas pret interferenci no diviem galvenajiem avotiem:

1. **Elektromagnētiskā interference**

No ārējiem avotiem (elektriskie motori, fluorescentas gaismas). Pret tās efektiem cīnās, ekranējot vadus ar īpašu metāla apvalku, kā arī izvēlētos pareizo vada zemēšanas veidu;

1. **Šķērsrunu (*crosstalk*)**

Elektromagnētiskie trokšņi starp atsevišķajiem vadiņiem. Kad elektriskais impulss plūst pa vadu, tas ap sevi izraisa nelielu apļveida magnētisko lauku, kas traucē blakus esošam kabelim. Pret tās efektiem cīnās, signāla vadu savijot pārī ar pretēji vērstas elektriskās ķēdes vadu, tādējādi kompensējot traucējumus.

Pastāv divu veidu vītā pāra kabeļi, no kuriem izvēlas piemērotāko, atkarībā no elektriskajiem trokšņiem vidē:

1. **UTP jeb neekranētie (*Unshielded Twisted Pair*)**

Tos var diezgan daudz locīt, tie aizņem maz vietas. Tie ir ļoti ietekmējami ar elektromagnētiskajiem traucējumiem, taču tie ir visplašāk izmantoti, lai savienotu tīkla iekārtas ar virzi vai maini.

1. **STP jeb ekranētie** **(*Shielded Twisted Pair*)**

Iedalās vairākos apakštipos, kuri atšifrējami pēc tajos sastopamajiem burtiem, kur “U” nozīmē bez ekranējuma, “S” nozīmē režģa ekranējums un “F” nozīmē folijas ekranējums. Slīpsvītras kreisajā pusē ir vada kopējais ekranējuma tips un labajā pusē individuālo pāru ekranējums:

* U/UTP – nav nekāda izolējuma ne visiem pāriem kopā, ne individuāli katram pārim;
* F/FTP – ar foliju izolēti visi pāri kopā un ar foliju izolēts arī katrs kabeļu pāris;
* S/UTP – ar režģi izolēti visi pāri kopā un bez izolējuma katrs atsevišķais kabeļu pāris;
* SF/FTP – ar režģi un foliju izolēti visi pāri kopā un ar foliju izolēts katrs kabeļu pāris.

Ja STP kabelis ir nepareizi zemēts, tas var darboties kā antena un uztvert nevēlamus signālus.

Vītā pāra kabeļu standartā ir ieviests sadalījums pa kategorijām, kas norāda kabeļa atbilstību noteiktām prasībām (metāla blīvums, materiāls). Kategorijas numurējas no 1 līdz 8 un var saturēt papildus indeksus, kas precizē kategorijas kvalitāti. Tā piemēram, mūsdienās, ļoti populāra ir **5e kategorija**, kas nodrošina 100Mb/s joslu pilna dupleksa režīmā.

* Cat8 = 40 Gb/s
* Cat7 = 10 Gb/s
* Cat6 = 1000 Mb/s
* Cat5 = 100 Mb/s
* Cat3 = 10 Mb/s

A standarts un B standarts atšķiras ar vadu secību, to pielietojums atkarīgs no tā, kādas ierīces savienojamas. Izšķir **pārejas** jeb *crossover* (A-B) un **caurplūdes** jeb *straight-through* (A-A vai B-B) savienojumu.

Vītā pāra kabeļos ir 4 pāri: zaļš un zaļi-balts (saņem datus), oranžs un oranži-balts (sūta datus), zils un zili-balts (PoE), brūns un brūni-balts (PoE).

# Optisko kabeļu datu pārraides vide

Otrs tīklos lietotais kabeļu tips. Salīdzinoši dārgāki par vara kabeļiem, tehnoloģija prasa precizitāti. Taču tiem ir vairākas priekšrocības. Tie ļauj pārsūtīt signālu lielākos attālumos ar lielāku joslas platumu nekā jebkurš cits datu pārraides medijs. Tie sūta datus ar mazāku signāla nodilšanu nekā vara kabeļi, turklāt ir imūni pret EMI. Optiskie kabeļi satur elastīgas, ļoti smalkas stikla šķiedras, kas ir nedaudz biezākas par cilvēka matu. Tām ir stikla apšuvums un polimēra apvalks.

1. **Multimodu šķiedra**

Oranži vai jūras-zaļi. Izmanto LAN tīklos, jo tajos esošās LED gaismas ir lētas. Gaisma šķiedrā ieiet no vairākiem leņķiem, jo šķiedras kodols ir 50 mikroni. Tādējādi gaismas staram piemīt lielāka dispersija, kas rada signāla zudumu. Var nest signālu 550 metru attālumā, to joslas platums ir līdz pat 10Gb/s.

1. **Vienmoda šķiedra**

Dzelteni. Izmanto dārgu lāzera tehnoloģiju ar ļoti koncentrētu gaismas staru. Šķiedras kodols ir vien 9 mikroni, kas ir ļoti tuvs rādītājs gaismas viļņa garumam. Var nest signālu 100 kilometru attālumā. Ja šķiedras vide būtu vakuums un lāzers spētu pārslēgties pietiekami ātri, teorētiski sasniedzama pārraide gaismas ātrumā. Taču atbilstošās mūsdienu tehnoloģijas ļauj sasniegt 100Gb/s ātrumu.

Optiskos kabeļus lieto četros gadījumos:

* **Korporatīvajos tīklos** – izmanto uzņēmumu pamattīkliem, lai savienotu tīklu infrastruktūras iekārtas.
* **“*Fiber-to-the-Home*”** – izmanto platjoslas pieslēgumiem mājām un nelieliem uzņēmumiem.
* **Tālsatiksmes tīkli** – lieto interneta pakalpojumu sniedzēji starppilsētu un starpvalstu tīkliem.
* **Zemūdens tīkli** – lieto, lai piegādātu uzticamu un ātru savienojumu, kas ir spējīgs izturēt zemūdens vidē līdz pat starpokeānu distancēm.

Optiskajiem kabeļiem ir pieejami vairāki savienotājposmi, to uzstādīšana notiek ar īpašām iekārtām speciālistu izpildījumā, tā ietver metināšanu, jo kabeļa videi jābūt stingri slēgtai:

* **ST (“*Straight-Tip*”)** – vieni no pirmajiem, kas tika lietoti. Tie droši nofiksējas ar uzskrūvējamu bajonetes tipa uzgali.
* **SC (*Subscriber Connector*)** – bieži izmantoti LAN un WAN tīklos. Izmanto “*push-pull*” mehānismu.
* **LC (“Lucent-Connector”) Simplex** – mazāka SC versija, tādēļ to popularitāte pieaug.
* **Duplex Multimode LC** – līdzīgi *simplex* LC, bet izmanto *duplex* savienojumus.

# Bezvadu lokālie tīkli

Bezvadu lokālie tīkli pārraida datus ar elektromagnētisko viļņu (EMV) starpniecību. Tie izmanto radioviļņu vai mikroviļņu frekvences. Bezvadu tīklu lietošana sniedz ērtības, jo tīklam var piekļūt no jebkuras vietas pārklājuma diapazonā, tie sniedz mobilitāti un ietver vienkāršu instalācijas procesu. Bezvadu tīkli mūsdienās ir galvenais veids, kā lietotāji pieslēdzas mājas vai darba tīklam. Pastāv vairāki ierobežojumi:

1. **Pārklājuma diapazons**

Bezvadu tīkla tehnoloģijas labi funkcionē atvērtās vidēs bez šķēršļiem, taču konkrēti materiāli, kas lietoti ēku būvē var negatīvi ietekmēt pārklājuma kvalitāti.

1. **Elektromagnētiskā interference**

Bezvadu tīklus var traucēt blakus esošas iekārtas (bezvadu telefoni, fluorescējošās gaismas, mikroviļņu krāsnis, citas bezvadu iekārtas).

1. **Drošība**

Atšķirībā no tīkla kabeļiem, bezvadu medijam nav nepieciešams piekļūt fiziski. Tādēļ tiek izmantota autentifikācija (**WPA3 jeb *Wi-Fi Protected Access*)**, kas aizstāja novecojušo WEP standartu) un šifrēšana (**AES, Galuā šifrs**).

1. **Dalīts medijs**

WLAN operē pus-dupleksajā režīmā, kas nozīmē, ka tikai viena iekārta vienlaikus var sūtīt vai saņemt datus. Jo vairāk lietotāji cenšas piekļūt bezvadu tīklam, jo vairāk tiek samazināts katra individuālā lietotāja joslas platums.

Bezvadu medijiem ir izveidoti standarti, kas attiecas gan uz fizisko slāni, gan kanāla slāni, piemēram, datu kodēšana radio signālos, signālu frekvence un transmisijas jauda, signāla uztveršanas un atkodēšanas process, antenas dizains u.c.

* **Bluetooth (IEEE 802.15)** – bezvadu personālo tīklu (WPAN) standarts, kas ietver iekārtu sapārošanu, un datu sūtīšanu līdz pat 100 metru attālumā.
* **Zigbee (IEEE 802.15.4)** – specifikācija zema datu pārraides ātruma un jaudas komunikācijām. Paredzēta tuvas darbības iekārtām, kurām nepieciešams ilgs akumulatora darbības laiks.
* **Wi-Fi (IEEE 802.11)** standartu saime ietver vairākus standartus, kurus apzīmē viens līdz divi latīņu alfabēta burti. Atšķiras joslas platums un frekvence. 2024. gada jaunākais standarts ir IEEE 802.11be jeb Wi-Fi 7, kas ietver 6GHz, 5GHz un 2.4GHz frekvences un 30Gb/s joslas platumu. Izmanto nesējkontroles daudzpiekļuves sadursmju novēršanas protokolu. Bezvadu tīkla saskarnes kartei vispirms jāklausās, vai kanāls ir brīvs un vajadzības gadījumā jāgaida. *Wi-Fi Alliance* nosaka, kuras ierīces ir klasificējamas un sertificējamas ar Wi-Fi preču zīmi.
* **Satelītu komunikācijas u.c.**

Tīkla iekārtām ir definēts **frekvenču diapazons jeb josla**. Piemēram, WLAN iekārta ar 2.4 GHz antenu var izmantot frekvences no 2.40 līdz pat 2.48 GHz. Frekvenču joslas ir sadalītas kanālos, kurus apzīmē skaitlis un tam ir specifiska frekvence. Kanāli 5GHz joslā nepārklājas, taču 2.4 GHz joslā pārklājas – tur nepieciešams specifiski konfigurēt tīkla punktus, lai tie maksimāli izmantotu nepārklājošus kanālus.

WLAN tīklam ar **infrastruktūras topoloģiju** nepieciešams:

1. **Bezvadu piekļuves punkts** – koncentrē bezvadu signālus no lietotājiem un savienojas ar vara vai optiskajiem kabeļiem. Mājas vai nelielu uzņēmumu lietotie bezvadu virži pilda gan virža, gan maiņa, gan bezvadu piekļuves punkta funkcijas;
2. **Bezvadu tīkla kartes adapteri** – ļauj sazināties tīkla iekārtām.

WLAN tīklā ar **Ad Hoc jeb IBSS (*Independent Basic Service Set*)** topoloģiju vairākas ierīces komunicē bezvadu medijā bez piekļuves punkta, ar *peer-to-peer* savienojumu.

# Decimālā, binārā un heksadecimālā skaitīšanas sistēmas

1. **Binārā skaitīšanas sistēma**

Binārajai skaitīšanas sistēmai bāze ir 2. Šajā sistēmā izmanto ciparus 0 un 1. Binārā skaitīšanas sistēma ir vienīgais datoram saprotamais datu reprezentācijas veids. Vienu bināro ciparu sauc par bitu.

IPv4 adreses ir viens 32 bitus garš skaitlis, kuru cilvēku ērtības labad mēdz dalīt četros 8 bitu oktetos un reprezentēt kā četrus decimālus skaitļus, atdalītus ar punktiem.

1. **Heksadecimālā skaitīšanas sistēma**

Heksadecimālajai sistēmai bāze ir 16. Šajā sistēmā izmanto ciparus no 0 līdz 9 un burtus no A līdz F (angļu alfabētā). Binārā un heksadecimālā skaitīšanas sistēmas labi strādā kopā, jo viens heksadecimāls cipars ir izsakāms četros bitos.

Heksadecimālo skaitīšanas sistēmu izmanto, piemēram, lai reprezentētu IPv6 un MAC adreses. IPv6 adreses ir 128 bitus garas un katrus 4 bitus reprezentē viens heksadecimāls cipars, tātad kopumā 32 heksadecimāli cipari. MAC adreses ir 48 bitus jeb 12 heksadecimālus ciparus garas.

Papildus

ciparos -bāzes skaitīšanas sistēmā var uzglabāt skaitļus.

Lai pārveidotu decimālu skaitli binārajā, var izmantot divus paņēmienus:

* **Dalīšana**

Skaitli nepārtraukti dala ar divi, piefiksējot atlikumu. Bināro skaitli iegūst, atlikumus sakārtojot pretējā secībā. Piemērs ar 159:

159 : 2 79, atlikums 1

79 : 2 39, atlikums 1

39 : 2 19, atlikums 1

19 : 2 9, atlikums 1

9 : 2 4, atlikums 1

4 : 2 = 2, atlikums 0

2 : 2 = 1, atlikums 0

1 : 2 = 0, atlikums 1

Tātad 10011111. Ar heksadecimāliem skaitļiem rīkojas analogi, tikai, dalot ar sešpadsmit, nevis divi.

* **Salīdzināšana**

Skaitli salīdzina ar divnieka pakāpēm, sākot no lielākās. Ja skaitlis pārsniedz doto pakāpi, tas tiek reducēts uz tā moduli pēc šīs pakāpes un ir zināms, ka atbilstošā binārā skaitļa pieraksta pozīcijā būs 1, citādi 0. Piemērs ar 159:

159 128 27 – patiess (1)

159 (mod 128) 31

31 64 26 – aplams (0)

31 32 25 – aplams (0)

31 16 24 – patiess (1)

31 (mod 16) 15

15 8 23 – patiess (1)

15 (mod 8) 7

7 4 22 – patiess (1)

7 (mod 4) 3

3 2 21 – patiess (1)

3 (mod 2) 1

1 1 20 – patiess (1)

Tātad 15910 100111112

Ar heksadecimāliem skaitļiem rīkojas analogi, tikai, salīdzinot ar sešpadsmitnieka, nevis divnieka pakāpēm.

Negatīvus skaitļus attēlo ar divnieka papildkodu.

# Ethernet protokols

Ethernet ir viena no divām mūsdienās lietotajām LAN (*Local Area Network* ) tehnoloģijām. Ethernet izmanto tīkla kabeļus, t. sk. vītā pāra, koaksiālos un optiskās šķiedras kabeļus. Ethernet atbalsta šādus joslas platumus: 10 Mb/s (Classic Ethernet), 100 Mb/s (Fast Ethernet), 1000 Mb/s, 10 Gb/s, 40 Gb/s un 100 Gb/s.

Ethernet strādā OSI modeļa kanāla un fiziskajos slāņos. Ethernet, izmanto divus atsevišķus kanāla līmeņa apakšslāņus: **loģiskā posma vadības (*Logical Link Control*)** slānis un **medijpiekļuves vadības (*Media Access Control*)** slānis.

1. **LLC (IEEE 802.2)** – komunicē starp tīkla programmatūru augšējos slāņos un aparatūru zemākajos slāņos. Tā datu kadros ievieto informāciju, kas identificē, kurš tīkla slāņa protokols tiek lietots šim kadram. Šī informācija atļauj vairākiem 3. slāņa protokoliem, piemēram, IPv4 un IPv6, izmantot to pašu tīkla interfeisu un mediju. LLC galvene dod kanāla slānim informāciju par to, ko darīt ar paketi. Piemēram, ja iekārta saņem datu kadru, tā var apskatīt LLC galveni un noskaidrot, ka tā ir domāta IP protokolam tīkla slānī.
2. **MAC (IEEE 802.3)** – atbild par datu iekapsulēšanu un piekļuvi medijam. Tas ietver Ethernet kadru adresēšanu un Ethernet kļūdu noteikšanu.

**Mazākais pieļaujamais Ethernet kadra izmērs ir 64 baiti un lielākais pieļaujamais ir 1518 baiti.** Kadri, kuru izmērs ir zem 64 baitiem (“*runt frame*”) vai virs 1518 baitiem (“*baby giant frames*”) tiek uzskatīti par nederīgiem un tiek automātiski ignorēti. Ethernet kadru sastāvs:

* **Preambula** (8 baiti) - izmanto sinhronizācijai starp sūtītāju un saņēmēju
* **Galamērķa MAC adrese** (6 baiti)
* **Sūtītāja MAC adrese** (6 baiti)
* **Tips** (2 baiti) - identificē augstāko līmeņu protokolu, kurš iekapsulēts Ethernet kadrā, piemēram, 0x86DD apzīmē IPv6
* **Dati** (46 baiti - 1500 baiti)
* **Kontrolsumma** (FCS) (4 baiti)

**Nesēja jušanas un sadursmju atklāšanas daudzpiekļuve jeb CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*)** ir tehnoloģija, kas ļauj vairākām iekārtā izmantot vienu vidi pus-dupleksajā režīmā, nosakot sadursmes un definējot algoritmu retranslācijai.

Kad tīkla karte veic transmisiju, tās signāls ir nesējs. Tīkla karte nosaka nesēja esamību un konsekventi nosūta signālu. Ja nesējs netika noteikts, iekārta zina, ka medijs ir brīvs translācijai. Sākotnēji Ethernet tīkli izmantoja vienu kabeli (koaksiālo), kuram fiziski pieslēdzās pārējie. Vēlāk izmantoja centrmezglus (*hubs*), kurus mūsdienās aizstāja maiņi. Maiņi atbalsta pilno-dupleksa režīmu, tādēļ CSMA/CD nav nepieciešams.

Ethernet datus ir svarīgi sūtīt pareizajos brīžos:

* **Slota laiks** – minimālais transmisijas ilgums.
* **Starpkadru atstarpe** – laiks starp kadru transmisiju.
* **Atkāpes laiks** – laiks, kas iekārtām jāgaida, kad notikusi sadursme.

Tīklu komunikāciju veidi: **uniraide, apraide, multiraide**.

**MAC adrese** ir 48 bitus gara. Pusi sastāda organizatoriskais unikālais identifikātors (regulē IEEE, tajā ir apraides / multiraides bits un lokāli administrētās adreses bits) un otru pusi – ražotāja piešķirtā iekārtas adrese.

# Ethernet komutēšana (*switching*)

Jo lielāks ir Ethernet tīkla segments, jo lielāks ir **sadursmju domēns**, kas samazina tīkla datu apmaiņas efektivitāti. Tādēļ Ethernet tīklus ir labi sadalīt vairākos segmentos. To var darīt dažādos veidos:

1. **Tilts (bridge)** – tīkla iekārta, kas sadala vienu sadursmju domēnu divos. Kad tilts saņem datu kadru, tas pārbauda, vai galamērķa MAC adrese ir tā MAC adrešu tabulā un attiecīgi nosūta kadru pa vienu no tā diviem portiem.
2. **Mainis (switch)** – tilts ar vairākiem portiem. Pie katra porta pieslēgtās iekārtas ir nodalītas atsevišķā domēnā. Ja pie porta pieslēdz tikai vienu ierīci, tās kolīzijas domēnā ir tikai mainis un tā pati – to sauc par mikrosegmentu.

Mainis izmanto MAC adreses, lai veiktu lēmumus, kurus kadrus kur pārsūtīt. Ja galamērķa adrese ir uniraides adrese, mainis to meklēs savā **MAC adrešu tabulā**. Ja adrese tabulā ir, mainis kadru pārsūtīs uz tai atbilstošo portu. Ja adreses tabulā nav, mainis to pārsūtīs pa visiem portiem, izņemot avotu. Mainis analizē visus ienākošos kadrus, nosaka sūtītāju un tam atbilstošo portu. Ja sūtītāja MAC adrese nav maiņa tabulā, to pievieno kopā ar atbilstošo porta numuru. Ja sūtītāja MAC adrese jau ir tabulā, mainis atjaunina šī ieraksta taimeri. Noklusētu Ethernet maiņi patur ierakstus MAC adrešu tabulā 5 minūtes. Izšķir divas pāradresēšanas metodes:

* **Uzkrājnosūtnes komutēšana (store-and-forward switching)** – mainis sagaida visu kadru, aprēķina CRC jeb kļūdas testu, balstoties uz vieninieku skaitu kadrā. Tikai ja CRC ir derīgs, mainis pārsūta kadru uzmeklētajā portā. Tādējādi mazāka daļa joslas platuma tiek aizņemta ar kļūdainiem kadriem. Noderīga, kad servisa kvalitātes nolūkos jāveic datu klasifikācija un prioritātes noteikšana (piemēram, VoIP ir lielāka prioritāte par HTTP).
* **Caurtekas komutēšana (*cut-through switching*)** – mainis pārsūta kadru jau pirms tas ir pilnībā saņemts, sagaidot tikai galamērķa adresi. Izšķir ***Fast-forward Switching*** un **bezfragmentu komutēšanu**, kad komutators tomēr saglabā pirmos 64-baitus un veic tiem pārbaudi pirms pārsūtīšanas (kompromiss starp abiem veidiem).

Ir iespējams konfigurēt maini, lai tas pārslēgtos no caurtekas komutēšanas uz uzkrājnosūtnes komutēšanu, kad sasniegts noteikts lietotāju skaita slieksnis.

Maiņu atmiņas buferizācija ļauj saglabāt datus pirms to pārsūtīšanas (piemēram, galamērķa ports ir nepieejams pārslodzes dēļ). Pastāv divi atmiņas buferizācijas veidi:

* **Uz portiem balstītā atmiņa** – dati tiek glabāti rindās, kas ir piesaistītas attiecīgajam portam. Ir iespējams, ka viens kadrs novēlo visu rindā esošo kadru transmisiju.
* **Dalītā atmiņa** – visi kadri ievietoti kopīgā atmiņas buferī.

**Auto-MDIX** ir tehnoloģija, kas ļauj maiņiem automātiski noteikt portam pievienoto kabeļu tipu – caurplūdes vai pārejas, lai atbilstoši konfigurētu tīkla interfeisu.

# IPv4 protokols

Viens no galvenajiem tīkla līmeņa protokoliem. IP jeb interneta protokols ļauj jebkurām internetā pieslēgtajām iekārtām sazināties ar jebkurām citām internetam piesaistītām publiski pieejamām iekārtām. IP nodrošina globālu adresācijas sistēmu (IP adreses) un īpašu datu struktūru (paketes), kas standartizē datu apmaiņu.

**IPv4 paketēm** ir galvene, kas palīdz nodrošināt, ka konkrētā pakete tiek piegādāta uz nākamo pieturvietu ceļā uz galamērķa iekārtu. Daži no galvenes laukiem:

* **Versija** – izmantotā IP versija
* **Galvenes garums**
* **DSCP** – izmanto, lai noteiktu paketes prioritāti
* **Dzīvlaiks (TTL)** – katru reizi, kad paketi apstrādā virzis, tas tiek samazināts par 1. Ja lauks sasniedz nulli, virzis paketi likvidē un sūtītājam nosūta ICMP kļūdas ziņu
* **Protokols** – iekapsulēto datu protokols: ICMP, TCP vai UDP
* **Sūtītāja IPv4 adrese** – uniraides
* **Galamērķa IPv4 adrese** – uniraides, multiraides vai apraides.
* **Kontrolsumma**

**IPv4 adreses** ir 32 bitus garas. Tās attēlo kā četrus decimālus skaitļus (0-255) atdalītus ar punktiem. Piemēram, 216.58.255.0 ir derīga IPv4 adrese. Kopumā ir aptuveni 4 miljardi IPv4 adrešu, to skaita nepietiekamība pamanīta jau 90. gados. Turklāt, ne visas adreses izmantojamas publiskajā internetā. Daži no rezervēto IPv4 adrešu diapazoniem:

* 0.0.0.0/8 (0.0.0.0 – 0.255.255.255)

Lietotas pašreizējam tīklam

* 10.0.0.0/8 (10.0.0.0 – 10.255.255.255)

Lietotas privātajam tīklam

* 127.0.0.0/8 (127.0.0.0 – 127.255.255.255)

Atgriezeniskās jeb *loopback* adreses

**Noklusējuma vārteja (*Default Gateway*)** ir tīkla iekārta, kas var pārsūtīt paketes uz citiem tīkliem. Parasti noklusējuma vārtejai ir lokālā IP adrese tajā pašā adrešu diapazonā kā citām tīkla iekārtām un tā var pieņemt datus no ārējiem tīkliem un nosūtīt tos lokālajām iekārtām (un *vice versa*).

**Tīkla adrešu translēšana jeb *Network Address Translation* (NAT)** ir tehnoloģija IP adrešu saglabāšanai. Tā ļauj privātiem tīkliem, kas lieto nereģistrētas IP adreses pieslēgties internetam, translējot to adreses par legālām tīkla adresēm pirms to sūtīšanas ārējos tīklos. Tas nozīmē, ka ar virža starpniecību datoru grupu var reprezentēt tikai ar vienu unikālu IP adresi. NAT palīdz saglabāt izzūdošās IPv4 adreses.

Publiskajā tīklā lietotās IP adreses piešķir **IANA (*Internet Assigned Numbers Authority*)**. Par IP adresēm konkrētā pasaules reģionā atbild reģionālie interneta reģistri.

IPv4 adreses sastāv no **tīkla daļas** un **saimniekdatora adreses**.

Internetam pievienotie saimiekdatori var sūtīt informāciju citiem trīs veidos:

* **Uniraide (*unicast*)**

Adreses 0.0.0.0 – 223.255.255.255

* **Apraide (*broadcast*)**

Adrese 255.255.255.255

* **Multiraide (*multicast*)**

Adreses 224.0.0.0 – 239.255.255.255

# IPv6 protokols

Jaunākā interneta protokola versija, kas nākotnē pilnībā aizvietos IPv4, jo tam ir trūkumi:

* **Adrešu trūkums** – 4 miljadri unikālu adrešu nav pietiekami arvien popularitātes ziņā pieaugošajam lietu internetam (IoT), arvien lielākam vienmēr pieslēgto savienojumu pieprasījumam un augošajam interneta lietotāju skaitam.
* **“*End-to-End*” savienojumu trūkums** – izmantotā NAT tehnoloģija piešķir iekārtu grupai vienu publisku adresi, kas apslēpj gala iekārtas adresi.
* **Palielināta tīkla sarežģītība** – NAT ir paildzinājis IPv4 mūžu, taču jāatceras, ka tas ir ieviests kā pagaidu pārejas mehānisms uz IPv6. Tas padara tīklu sarežģītāku, veido papildu aiztures un padara traucējummeklēšanu sarežģītāku.

90. gadu sākumā sāka veidot IPv6 ar uzlabojumiem, kas iekļauj:

* **Palielināts adrešu skaits** – adreses garums ir 128 biti, kas sniedz aptuveni 340 undecilionu jeb 3,4 1038 adreses.
* **Uzlabota pakešu struktūra** – lauki maz.
* **Nav nepieciešams NAT** – vienkāršo tīklu.

Nav konkrēta datuma, līdz kuram plānots veikt pilnīgu pāreju uz IPv6, tādēļ šobrīd abi protokoli tiek lietoti paralēli, to panāk ar vairākām metodēm:

1. **Duālais steks** – iekārtas darbina abus protokolus vienlaikus
2. **Tunelēšana** – ļauj transportēt IPv6 paketes pa IPv4 tīklu, iekapsulējot tās.
3. **Translēšana** – NAT64 ļauj IPv6 iekārtām komunicēt ar IPv4 iekārtām ar tehniku, kas līdzīga NAT. Paketes tiek dinamiski pārveidotas par tīkla atbalstītajām paketēm. Strādā abos virzienos.

IPv6 adreses ir 128 bitus garas. Tās attēlo kā astoņus heksadecimālus skaitļus (0000-ffff) atdalītus ar koliem. Vadošās nulles katrā hekstetā (16 bitu fragmentā) tiek atmestas, tādējādi. Dubultais kols apzīmē virkni ar hekstetiem, kas sastāv tikai no nullēm, tādējādi, piemēram, 2001:db8:cafe:1:0:0:0:1 var tikt saīsināts kā 2001:db8:cafe:1::1.

Internetam pievienotie saimiekdatori var sūtīt informāciju citiem trīs veidos:

1. **Uniraide (*unicast*)**
2. **Multiraide (*multicast*)**
3. **Kādamraide (*anycast*)**

Kādamraide ir vairākām iekārtām piederoša viena IPv6 adrese, paketes tiek sūtītas uz vispieejamāko galamērķi.

IPv6 prefiksi var būt garumā no 0 līdz 128 bitiem, tos pieraksta kā slīpsvītru aiz adreses. Prefiksa garums nosaka, kura adreses daļa identificē tīklu, kura interfeisu. IPv6 adreses var iedalīt:

* **Globālās uniraides**

Analogas publiskajām IPv4 adresēm. Sastāv no globālā maršrutēšanas prefiksa, ko piešķir ISP. Parasti tas ir 48 bitus garš. Tam seko apakštīkla identifikators (Subnet ID), parasti 16 bitus garš. Visbeidzot ir interfeisa identifikators (Interface ID), kas ir ekvivalents IPv4 saimniekdatora daļai. Termins interfeiss tiek lietots, jo vienai tīkla iekārtai tie var būt vairāki, katrs ar savu IPv6 adresi.

* **Lokālā kanāla (*Link-Local Address*)**

Nepieciešama katrai IPv6 iekārtai. Izmantota, lai komunicētu ar citām iekārtām vienā apakštīklā. Virži nepārsūta paketes, kas satur kādu LLA adresi. Var ģenerēt manuāli vai dinamiski ar gadījumskaitļu un MAC adreses palīdzību.

* **Atgriezeniskā** – analoga IPv4 atgriezeniskajām adrešu diapozonam – ::1/128
* **Nenoteiktā** – adrese ::/128
* **Unikālās lokālās** – adreses diapazonā fc00::/7 – virza tikai privātajā tīklā. Var bez reģistrācijas tikt lietotas organizācijas ietvaros.
* **Iegultās IPv4** – izmanto tunelēšanai.

# IP virzīšana (routing)

Virzīšana ir virziena noteikšanas metode datu pārraides laikā, ko realizē, izmantojot maršrutēšanas tabulas. Kad tīkla iekārta vēlas nosūtīt datus citai tīkla iekārtai, tā izmanto savu maršrutēšanas tabulu, lai veiktu lēmumu, kā to nosūtīt. Ja galamērķa adrese atrodas citā tīklā, tad pakete tiek nosūtīta noklusētajai vārtejai (visbiežāk virzis). Virzis apskata paketes galamērķa IP adresi, meklē to savā maršrutēšanas tabulā, lai noteiktu kur paketi pārsūtīt. Maršrutēšanas tabula satur visas zināmās tīkla adreses un tām atbilstošos izejas portus. Šos ierakstus sauc par virzieniem. Virzis pārsūta paketi, izmantojot vislabāko (tādu, kurš visvairāk sakrīt) virzienu.

**Virzīšanas tabula** glabā 3 dažādu veidu ierakstus:

* **Tieši savienotie tīkli** – virzieni, kas ved uz virzim tieši pieslēgtajām iekārtām.
* **Tālie tīkli** – virzieni, kas ved uz iekārtām, kas ir citos tīklos.
* **Noklusētais maršruts** – arī viržiem ir noklusētā vārteja, ko tie izmanto gadījumos, kad tabulā nav labākā virziena.

Virži var uzzināt par tīkliem vienā no diviem veidiem:

1. **Manuāli** – tīkla administrators ievada tālos tīklus virzīšanas tabulā (statisks virziens) un tiem atbilstošo nākamā lēciena (“*next hop*”) adresi. Ja notiek izmaiņas tīkla topoloģijā, šos ierakstus nepieciešams manuāli atjaunināt. Statiskie virzieni piemēroti maziem tīkliem, kur ir maz redundantu posmu.
2. **Dinamiski** – tālie tīkli tiek automātiski atklāti, izmantojot dinamisko virzīšanas protokolu. Virži, kas izmanto dinamisko virzīšanas protokolu, automātiski dalās ar virzienu informāciju ar citiem viržiem. Arī tīkla topoloģijas izmaiņas tiek noteiktas automātiski.

Dinamiskie virži atklāj tālos tīklus, uztur aktuālu virzīšanas informāciju, izvēlas labāko virzienu uz galamērķa tīkliem un atrod jaunu virzienu, ja iepriekšējais vairs nav pieejams.

**Virzīšanas informācijas protokols jeb *Routing Information Protocol* (RIP)** izmanto lēcienu (hops) skaitu, lai noteiktu dažādu virzienu vērtības. Lēcienu skaits atbilst tam, cik iekārtām virziena maršruts ved cauri. RIP regulāros intervālos izsūta maršrutēšanas atjauninājumu ziņas, kuru sniegtā informācija ļauj tīkla iekārtām uzlabot to virzīšanas tabulas.

Lielākā RIP vērtība ir 15 (ja nesasniedzams galamērķis, tad 16), tātad tas nav piemērots lieliem tīkliem. Mūsdienās izplatītāks ir **Pirmā atvērtā īsākā veļa protokols jeb *Open Shortest Path First* (OSPF)**, kuru izveidoja IETF, tas atbalsta gan IPv4, gan IPv6.

# TCP protokols

Pārraides vadības protokols jeb Transmission Control Protocol (TCP) ir datortīklu protokols, kas strādā OSI modeļa 4. (transporta) slānī un nodrošina funkcijas:

* **Uzticama piegāde** – ievieš mehānismus gadījumiem, kas dati tiek sabojāti vai pazaudēti.
* **Plūsmas vadība** – nodrošina efektīvu datu plūsmu starp iekārtām.
* **Sekvencēšana** – nodrošina unikālu marķējumu katram datu segmentam, ko saņēmējiekārta izmanto, lai atgrieztu sākotnējo.

TCP segmentē augstāko slāņu protokolu datus sīkākās daļās – segmentos, tādējādi augstāko līmeņu protokoli var uzskatīt pārraidāmos datus par vienu kopumu un ļaut TCP organizēt to dalīšanu.

Lai uzturētu sarunas stāvokli un izsekotu informācijai, TCP izveido savienojumu starp sūtītāju un saņēmēju. Tādējādi TCP ir uz savienojumu orientēts protokols. Šo savienojumu sauc par sesiju un to izveido pirms datu transmisijas, nosakot satiksmes daudzumu, kas var tikt nosūtīts konkrētā laika posmā.

TCP uzticamības dēļ tas ir pieņemtais protokols, piemēram, mājaslapu un e-pastu datu plūsmām, kur datiem jāierodas tā, kā tie sākotnēji sūtīti. TCP galvene aizņem 160 bitus:

* **Sūtītāja ports**
* **Galamērķa ports**
* **Sekvences skaitlis**
* **Apstiprinājuma skaitlis**
* **Galvenes garums**
* **Rezervētais laiks**
* **Kontroles biti**
* **Loga izmērs**
* **Kontrolsumma**
* **Steidzamības lauks**

**“*Three way handshake*” jeb trīs virzienu rokasspiediens**. Izmantots TCP sesijas izveidei ar īpašām 3 kontroles ziņām:

1. **SYN** – savienojuma uzsākšanas pakete, kas satur nejauši ģenerētu sekvences skaitli (piemēram, 5432), kas apzīmē sākumu iekārtas A sūtītajiem datiem.
2. **SYN-ACK** – iekārta B atbild pati ar savu sekvences skaitli (piemēram, 9876), kā arī ar iekārtas A sūtīto sekvences skaitli, palielinātu par 1 (5433).
3. **ACK** – iekārta apstiprina iekārtas B sūtīto, atbildot ar iekārtas B sūtīto sekvences skaitli, palielinātu par 1 (9877).

Lai pārtrauktu TCP savienojumu, izmanto **FIN (*finish*)**, kuru otra iekārta apstiprina ar ACK.

**Ports ir skaitlis**, kas nosaka, kuram operētājsistēmas procesam jāpiegādā ziņa. Tādējādi iekārta var uzturēt vairākus savienojumus vienlaikus. 443 izmanto HTTPS, 25 izmanto SMTP, 80 izmanto HTTP, FTP, SSH utt.

TCP ir nederīgs gadījumos, kad nav pieļaujama aizkave (VoIP), taču dažreiz ugunsmūri bloķē UDP, tādēļ nākas izmantot TCP. TCP ir nederīgs gadījumos, kad jāveic multiraides ziņas (DCHP).

Aplikācijas, kas straumē uzglabātus audio un video ierakstus, izmanto TCP. Ja tīkls vairs pēkšņi nevar atbalstīt joslas platumu, programma iepauzē atskaņošanu. Pauzes laikā iespējams redzēt ielādes ekrānu. TCP pārnes aptuveni 80% interneta satiksmes.

TCP plūsmas kontrolē **loga izmērs** nosaka, cik baiti var tikt nosūtīti līdz sūtītājs sagaida apstiprinājumu (ACK). Loga izmērs ir iekļauts katrā pieprasījumā, tādēļ to ir iespējams dinamiski mainīt, atbildot uz pārslodzi. **Maksimālais segmenta izmērs** nosaka lielāko segmentu, ko galamērķa iekārta var saņemt vienlaikus. Parasti 1460 biti, neskaitot galvenes izmēru.

# UDP protokols

UDP (User Datagram Protocol) ir labākā mēģinājuma OSI modeļa transporta līmeņa protokols, Tas ir vienkāršāks nekā TCP:

* **Nenodrošina sekvencēšanu**, dati tiek salikti tādā secībā, kādā tie ir saņemti
* **Nepiegādātie segmenti netiek pārsūtīti**
* **Nenotiek sesijas izveide**

Ja nepieciešama uzticama piegāde ar UDP, tas jārealizē augstākajos slāņos. UDP ir **bezsavienojuma protokols**, tas neveido sesijas un tam nav rokasspiedienu mehānisma.

UDP ir ļoti atbilstošs gadījumos, kad **nepieciešama maza aizture**, piemēram, tiešsaistes video saziņā un VoIP zvanos. Dzīvā video un audio plūsmas **var tolerēt nelielus pakešu zudumus** bez pamanāma efekta, kas tos padara piemērotus UDP.

UDP komunikācijas bloki ir **datagrammas jeb segmenti**. Galvenē ir tikai 4 lauki, katrs 2 baitus liels:

* **Sūtītāja ports**
* **Galamērķa ports**
* **Garums**
* **Kontrolsumma**

Datagrammas ar nederīgu kontrolsummu tiek izmestas un par šo kļūdu netiek informēta programmatūra.

Uzsākot komunikāciju, UDP klients pieprasa datus no servera, UDP klienta process dinamiski izvēlas porta skaitli no skaitļu diapazona. To sauc par **sarunas sūtītāja portu**. **Galamērķa ports** parasti ir kāds no labi zināmajiem jeb reģistrētajiem portu skaitļiem. Klienta izvēlētie skaitļi tiek iekļauti visu datagrammu galvenēs šīs transakcijas ietvaros. Servera sūtītajās datagrammās tie ir apmainīti vietām.

Portus var iedalīt vairākās kategorijās:

1. **Labi zināmi porti** – no 0 līdz 1023. UNIX bāzētās sistēmās nepieciešamas superlietotāja tiesības, lai tos izmantotu. Definēti bieži izmantotiem servisiem.
2. **Reģistrētie porti** – no 1024 līdz 49151. Tos specifiskiem iemesliem atvēlē IANA.
3. **Dinamiskie porti** – no 49152 līdz 655353. Nav iespējams reģistrēt ar IANA. Lieto privātiem servisiem vai gadījumos, kad ports nepieciešams īslaicīgi.

UDP var tikt izmantots DDoS uzbrukumos, kur uzbrucēji, neveicot savienojuma izveidi, uz serveri nosūta lielu skaitu datagrammu uz nejaušiem portiem. Serveris ir spiests atbildēt ar ICMP ziņām, ka porti nav pieejami.

UDP lieto, piemēram, DNS (komunikācija starp diviem DNS serveriem izmanto TCP) un RIP (tā kā tas ir uz apraidi bāzēts).

# Populārākie lietojumu slāņa protokoli un pielietojumi

TCP/IP modeļa lietojuma slānis ir lietotājam vistuvāk esošais slānis. Tas izveido saskarni starp lietojumprogrammām, ko izmanto komunikācijām, un zemākajiem tīkla slāņiem. Lietojumu līmeņa protokols tiek izmantots, lai apmainītu datus starp programmām uz atsevišķām tīkla iekārtām.

|  |  |  |
| --- | --- | --- |
| **Protokols** | **Ports** | **Apraksts** |
| DNS (Domēnu vārdu sistēma) | 53 | Glabā domēnu vārdiem atbilstošās IP adreses. Serviss ļauj gala lietotājiem izmantot vieglāk iegaumējamos domēnu vārdus. Serviss ļauj domēnu īpašniekiem mainīt savu domēnu IP adreses, nemainot pašu domēnu. Tas nozīmē, ka gala lietotājiem domēna adreses maiņa ir neredzama un neizraisa apgrūtinājumus. DNS satndarti nosaka pieprasījumu un atbilžu formātu. DNS komunikāciju sauc par ziņu. DNS ieraksts ietver domēna nosaukumu, adresi un tipu (A – galaiekārta ar IPv4 adresi, AAAA – gala iekārta ar IPv6 adresi, NS – autoritatīvs vārdu serveris, MX – pasta serveris). |
| Telnet | 23 | Divvirzienu tekstorientēts protokols, kas nodrošina virtuālu piekļuvi ierīcei. Telnet nav drošs, jo saziņas ietvaros nenotiek šifrēšana. |
| SSH (Secure Shell) | 22 | Nodrošina virtuālu piekļuvi ierīcei. Atšķirībā no Telnet, SSH ir drošs, jo saziņas ietvaros notiek drošas piekļuves nodrošināšana un šifrēšana. Autentifikācija notiek ar publisko un privāto atslēgu pāri. |
| FTP (Datņu pārsūtīšanas protokols) | 20, 21 | Izmanto failu pārsūtīšanai starp datoriem. Tas izmanto klienta-servera modeli, kur klients var datus augšupielādēt serverī un lejupielādēt no tā. FTP pieprasa divus savienojumus (plūsmas kontrolei un datu sūtīšanai). |
| DHCP (Dinamiskais saimniekdatora konfigurācijas protokols) | 67, 68 | Automātiski iedala IP adreses un tīkla uzstādījumus. Kad tīkla iekārta, kas ir konfigurēta izmantot DHCP, pieslēdzas tīklam , šis klients apraida DHCP “discover” ziņu, lai identificētu pieejamos DHCP serverus. DHCP serveris atbild ar IPv4 adresi, apakštīkla masku, IPv4 DNS servera adresi un IPv4 noklusējuma vārtejas adresi. |
| HTTP (Hiperteksta transporta protokols) | 80 | Paredzēts datu apmaiņai starp tīmekļa serveriem un pārlūkprogrammām. HTTP definē ziņas veidu jeb metodi (GET – klients pieprasa datus, POST – klients sūta datus un PUT – augšupielādē resursu uz serveri) |
| SMTP (Vienkāršais pasta pārsūtīšanas protokols) | 25, 465 | Paredzēts saziņai starp e-pastu serveriem. Definē pieprasījumu galvenes un datu formātu. Kad klients sūta e-pastu, klienta SMTP process savienojas ar SMTP serveri un cenšas nosūtīt ziņu. Serveris nosaka, vai ziņa ir lokāla, atbilstoši izvēloties, vai pārsūtīt to tālāk uz citu serveri vai nē. |

# Enkapsulācijas un dekapsulācijas procesi

Datu enkapsulācija jeb iekapsulēšana ir augstāka līmeņa protokola metadatu (piemēram, galvenes) un datu ievietošana zemāka līmeņa protokola datu vienībās.

1. **Lietojumu līmenī** (OSI – L7) tiek definēti nosūtāmie dati. Piemēram, HTTPS pieprasījums tīkla serverim.
2. **Prezentācijas līmenī** (OSI – L6) pieprasījums tiek izveidots HTTPS atbilstošā formātā un tiek pielietota šifrēšana.
3. **Transporta slānī** (OSI – L4) augstāko slāņu dati tiek segmentēti.

Segmentēšana ietver datu sadalīšanu mazākās vienībās, kuras ir vienkāršāk pārsūtīt pa tīklu. Segmentēšana ir nepieciešama jo TCP/IP saimes protokoli sūta datus individuālās paketēs.

Tas ļauj katrai paketei individuāli noteikt ātrāko ceļu uz galamērķi, kas palielina pārraides ātrumu, kā arī ļauj vairākām datu straumēs noritēt vienlaikus.

Segmentēšana nodrošina lielāku datu pārraides efektivitāti, jo segmenta sūtīšanas kļūdas gadījumā ir jāpārsūta tikai konkrētais segments.

HTTPS izmanto TCP, lai nodrošinātu uzticamu piegādi. Lai nodrošinātu datu secības saglabāšanu, sūtīšanas procesā tiek izmantota sekvencēšana. Katram segmentam tiek piešķirts tā secībai atbilstošs skaitlis.

Katram segmentam tiek pievienota galvene. TCP gadījumā sūtītāja ports, galamērķa ports, sekvences skaitlis kontrolsumma u.c.

1. **Tīkla līmenī** (OSI – L3) segments tiek iekapsulēts, tam pievienojot tīkla galveni.

Pieņemot, ka datu pārraide notiek IPv4 tīklā, paketes sākumā tiktu pievienoti galvenes lauki: sūtītāja IPv4 adrese, Galamērķa IPv4 adrese, DSCP, protokols u.c.

IP adrese ļauj paketei nonākt pareizajā galamērķī, DSC ļauj noteikt paketes prioritāti un protokols ļauj noteikt iekapsulētā segmenta protokolu. Piemēram, ja reizē ar šo HTTP pieprasījumu notiktu VoIP zvans, tā paketes tiktu nosūtītas ar augstāku prioritāti.

1. **Kanāla līmenī** (OSI – L2) IP paketes tiek iekapsulētas kadros.

Paketes sākumā tiek pievienota kadra galvene un beigās – kadra noslēgums (frame trailer). Pieņemot, ka kadrs tiek sūtīts Ethernet medijā, tam pievienoti lauki: sūtītāja MAC adrese, galamērķa MAC adrese, tips un kadra kontrolsumma.

1. **Fiziskajā līmenī** (OSI – L1) tīkla karte kadrus pārraida fiziskos signālos, piemēram, pa vītā pāra kabeli elektrisko impulsu veidā.

Kad saņēmējiekārta (tīkla serveris) saņem šos signālus, tie tiek dekapsulēti – pārveidoti no zemāka līmeņa protokoliem – augstākos.

1. Ethernet kadram tiek pārbaudīta kontrolsumma, izmantojot CRC pārbaudi. Ja tā ir nederīga, kadrs tiek izmests. Ja tā ir derīga, tiek noteikts kadra tips un kadrs tiek dekapsulēts (Ethernet līmeņa lauku noņemšana) par IP paketi.
2. No IP paketes tiek noņemti nevajadzīgie lauki (IP adreses). IP paketei tiek pārbaudīta galvenes kontrolsumma.
3. IP paketēs esošie dati tiek verificēti izmantojot transporta līmeņa kontrolsummu. Dati tiek salikti kopā, izmantojot sekvences skaitļus. Tiek identificēts galamērķa ports un dati tiek nosūtīti pareizajai programmai.

# DNS sistēma

Tā kā atcerēties konkrētas iekārtu adreses, lai piekļūtu tīkla resursiem ir nepraktiski, tiek izmantoti domēnu nosaukumi. DNS protokols definē automatizētu servisu, kas ļauj atrast domēnu vārdiem atbilstošās IP adreses.

DNS apmaiņas struktūra:

1. Saimniekdators vēlas veikt pieprasījumu uz domēnu, kuram tas nezina IP adresi, piemēram, <https://google.com>
2. DNS pieprasījums tiek nosūtīts uz klienta konfigurēto DNS serveri
3. DNS serveris atrod FQDN (Fully Qualified Domain Name) atbilstošo ierakstu
4. DNS serveris atbild klientam ar pieprasītā FQDN IP adresi
5. Klients veic pieprasījumu uz FQDN adresi

DNS serveri uzglabā dažādu tipu ierakstus. Ieraksti satur vārdu, adresi un ieraksta tipu:

* A – galaiekārta ar IPv4 adresi,
* AAAA – gala iekārta ar IPv6 adresi,
* NS – autoritatīvs vārdu serveris,
* MX – pasta serveris

Kad klients veic pieprasījumu, DNS serveris vispirms apskata savus ierakstus. Ja pieprasītais domēns nav pieejams tā ierakstos, tas sazinās ar citiem DNS serveriem. Pēc tam, kad ir atrasts meklētais ieraksts, DNS serveris to īslaicīgi saglabā, lai gadījumā, ja tas tiek prasīts vēlreiz, DNS serverim tas būtu pieejams.

DNS ziņa satur laukus: jautājums, atbilde, autoritāte un citi. DNS parasti izmanto UDP to mazās aizkaves dēļ, taču ir daži izņēmumi:

* DNS pieprasījums ir pārāk liels
* Saziņai starp DNS serveriem
* “DNS-over-HTTPS”

DNS protokolos izmantota hierarhiska sistēma, kuras hierarhiju nosaka domēna vārds.

Domēnu vārdu struktūra ir sadalīta mazākās zonās. Katrs DNS serveris uztur konkrētu datubāzes failu un ir atbildīgs tikai par konkrētiem vārds-par-IP pieprasījumiem. Kad tas saņem pieprasījumu ārpus tā zonas, to pārsūta citam DNS serverim.

Domēniem ir vairākas daļas, piemēram augšējā līmeņa domēni (TLD), kurus iedala vairākās kategorijās:

* ccTLD (country-code-Top-Level-Domain) – vienmēr 2 simbolus garš, kas identificē valsts kodu
* gTDL (generic-Top-Level-Domain) – vienmēr garāks par 2 simboliem, lieto vispārīgiem nolūkiem

Ir divi DNS serveru tipi:

* Autoritatīvie DNS – autoritāte pār kādu reģionu, tie glabā savas zonas DNS ierakstu tabulu un atbild uz rekursīvo DNS serveru pieprasījumiem
* Rekursīvie DNS – neglabā ierakstus (tikai īslaicīgi), taču zina, kur atrast atbildes:

1. Veic pieprasījumu root vārdu serverim
2. root serveris atgriež adresi augšējā līmeņa domēnu DNS serverim, piemēram, .net serverim
3. Rekursīvais DNS serveris veic pieprasījumu TLD DNS serverim, kurš atgriež konkrētā domēna adresi
4. Rekursīvais DNS serveris veic pieprasījumu uz DNS vārdu serveri, lai iegūtu domēna adresi.

Katru domēna daļu, kas atdalīta ar punktiem, sauc par iezīmi (label). Katras iezīmes maksimālais garums ir 63 simboli, bet visa domēna maksimālais garums 253 simboli. Lielie un mazie burti tiek uzskatīti par ekvivalentiem. Domēni var izmantot alfanumeriskos simbolus un defises, taču tie nevar sākties vai beigties ar defisi.

# Kādas datortīkla plūsmas notiek, atverot tīmekļa vietni

Lai informācija sāktu savu ceļu no lietojumu līmeņa, pirmais solis ir DNS uzmeklēšana. Tam izmanto DNS serverus, kas norādīti ierīces tīkla iestatījumos (vai kurus piešķīris DHCP). Ar UDP palīdzību tiek iegūts atbilstošais FQDN ieraksts, kas satur domēna vārdam atbilstošo IPv4 vai IPv6 adresi. Parasti tas tiek panākts ar ISP rekursīvo DNS serveri, kas vispirms vaicās augstākā līmeņa DNS serverim

Kad pārlūkprogrammai ir IP adrese, tā uzsāk saziņu ar attālo tīmekļa serveri. Tas tiek darīts, izmantojot norādīto protokolu. Pieņemsim, ka vietne izmanto HTTPS, kam nepieciešams TCP, tādēļ tiek izveidots savienojums ar serveri pirms pieprasījuma veikšanas. To panāk ar trīs virzienu rokasspiedienu.

Kad savienojums ir izveidots un zināmi sinhronizācijas, kā arī apstiprinājuma skaitļi, tiek veikts 'GET' pieprasījums lapai '/kontakti'. Pieprasījums no lietojuma slāņa iziet visus segmentācijas un enkapsulācijas posmus, kas raksturīgi datortīklu saziņai. Serveris saņemto kadru dekapsulē un iegūst sākotnējo pieprasījumu, uz kuru atbild ar HTML dokumenta saturu.

Iespējams, serveris dara to pašu, ko lietotāja parlūkprogramma – izmanto DNS, veido savienojumus un pieprasa vajadzīgos ārējos resursus, kuri ir nepieciešami vietnes funkcionalitātei, piemēram, datubāzes, stilus, aprēķinus, bildes u.c. Daži serveri var turpināt asinhronizācijas pieprasījumus tikai pēc atbildes.

Kad iegūti visi segmenti no servera, tie tiek sakārtoti sākotnējā secībā pēc sekvences skaitļiem. Dati tiek nogādāti konkrētajai pārlūkprogrammai pēc porta numura. HTTPS gadījumā dati ir šifrēti un kompresēti, par to attēlošanu pārlūkprogrammai saprotamā formātā atbild prezentācijas slānis. Visbeidzot pārlūkprogramma parsē dokumentu un atrod visus vietrāžus URL, kas tai ir jāievieto lapā (piemēram, attēlus vai saistītās stila lapas), un veic GET pieprasījumus par katru no tiem. Vietnes renderēšana sākas tūlītēji, tādēļ, iespējams, ka tā sākotnēji redzama tikai daļēji.

Pārlādējot lapu, pārlūkprogramma pārbauda, ​​vai šī lapa jau ir saglabāta jūsu sistēmas kešatmiņā, un, ja tā, tā veic HTTP pieprasījumu tikai dokumenta galvenei un pārbauda tās modificēšanas datumu. Ja šis datums ir vēlāks par kešatmiņā saglabāto kopiju, tas vēlreiz pieprasa pilnu dokumenta saturu un atsvaidzina lapu. Pretējā gadījumā tas izmanto tikai vietējo kopiju.

# Cisco IOS lietošana un iekārtas konfigurēšana

**Aparatūra** (*Hardware*) ir datora fiziskā daļa, **kodols** (*Kernel*) komunicē starp programmatūru un aparatūru un **čaula** (*Shell*) ir lietotāja saskarne.

Cisco IOS ir operētājsistēma, ko izmanto Cisco tīkla iekārtas – virži un maiņi. Lai pieslēgtos, jāizmanto termināļa emulators PuTTy.

Cisco IOS ir divi dažādi komandu izpildes režīmi – lietotāja režīms jeb User EXEC un privileģētais režīms jeb Privileged EXEC.

Lietotāja režīms ļauj veikt apsekošanas darbības, bet neļauj veikt izmaiņas konfigurācijā.

Privileģētais režīms ļauj veikt sistēmas izmaiņas. Privileģētajam režīmam ir vairāki apakšrežīmi. Lai konfigurētu iekārtu jāieiet globālās konfigurācijas režīmā, no kura iespējams veikt vispārīgas izmaiņas iekārtai. Globālajam konfigurācijas režīmam nepieciešams piekļūt pirms lietotājs var ieiet kādā no specifiskajiem konfigurācijas režīmiem, piemēram:

1. **Līnijas konfigurācijas režīms (*Line Configuration Mode*)** – izmanto, lai konfigurētu konsoles, SSH un Telnet piekļuvi.
2. **Interfeisa konfigurācijas režīms (*Interface Configuration Mode*)** – izmanto, lai konfigurētu maiņa portus un virža interfeisus.

Lai pārietu no viena režīma uz otru, izmanto šādas komandas:

* enable – pāriet no lietotāja režīma uz privileģēto
* disable – pāriet no privileģētā režīma uz lietotāja
* configure terminal – pāriet no privileģētā režīma uz globālās konfigurācijas režīmu
* exit - pāriet no globālās konfigurācijas režīma uz privileģēto režīmu. Arī pāriet no jebkura apakškonfigurācijas režīma vienu soli atpakaļ.
* line console 0 – pāriešana no globālās konfigurācijas režīma uz konfigurācijas režīmu

Iekārtu konfigurācijas iespējas:

* hostname (globālajā konfigurācijas režīmā) – iekārtas pārsauc, lai tās vieglāk būtu identificēt. Pēc noklusējuma Cisco maiņiem ir vārds “Switch”, taču tos vēlams pārsaukt tā, lai to vārdiem būtu cilvēkam saprotama nozīme, piemēram, “mainis-stavs2” varētu apzīmēt otrajā stāvā esošu maini. Iekārtu vārdiem jāsākas ar burtu, tie nedrīkst saturēt atstarpes un drīkst lietot tikai burtus, ciparus un domuzīmes.
* Password (līnijas konfigurācijas režīmā) – paroles iestata iekārtām, lai nodrošinātu labāku tīkla drošību. Paroles iestata atsevišķi konsoles piekļuvei, virtuālā termināļa jeb VTY piekļuvei, kā arī privileģētā režīma piekļuvei.
* Password-encryption (globālajā konfigurācijas režīmā) – tā kā pēc noklusējuma iespējams apskatīt konfigurācijas failos nešifrētas paroles, liek mainim šifrēt visas paroles. Tādējādi tiek likvidēti draudi tīkla drošībai, neļaujot neautorizētiem lietotājiem apskatīt paroles.

Cisco iekārtām ir divu veidu konfigurācijas:

1. startup-config – konfigurācijas fails, kas saglabāts NVRAM (*Non-Volotile RAM*). Saglabāts arī pēc iekārtas izslēgšanas.
2. running-config – pašreizējās konfigurācijas informācija, kas saglabāta RAM. Šī faila izmainīšana rezultējās tūlītējās izmaiņās iekārtas konfigurācijā.

Ja ir nepieciešams pašreizējo konfigurāciju saglabāt arī pēc iekārtas izslēgšanas, nepieciešams to pārkopēt startup-config vietā, izmantojot copy komandu. ? palīdzība un Tab pabeigšana.

# Datoru tīklu drošība

Pastāv četri galvenie uzbrukumu mērķi:

* **Informācijas zādzība (*Information Theft*)** – ielaušanās datorsistēmā, lai iegūtu konfidenciālu informāciju, kas var tikt pārdota vai izmantota citiem nolūkiem.
* **Datu iznīcināšana un manipulēšana (*Data Loss and Manipulation*)** – ielaušanās datorsistēmā, lai iznīcinātu vai mainītu ierakstus.
* **Identitātes zādzība (*Identity Theft*)** – personīgās informācijas zādzība ar nolūku pārņemt cilvēka identitāti. Izmantojot zagtu personīgo informāciju, ir iespējams, piemēram, pieteikties aizdevumiem, veikt neautorizētus pirkumus utt.
* **Servisa traucēšana (*Disruption of Service*)** – traucējumu izraisīšana, kas liedz sistēmas lietotājiem tai piekļūt pilnvērtīgi.

**Vājību** (*vulnerabilities*) veidi:

1. **Tehnoloģiskie uzbrukumi**

**Nedroši TCP/IP protokoli** (protokoli kā HTTP un FTP datus pārraida uzbrucējiem potenciāli redzamā formā, jo nelieto šifrēšanu)

**Uzbrukumi operētājsistēmām** (mēdz tikt atklātas drošības problēmas, kas var ļaut piekļuvi uzbrucējiem)

**Uzbrukumi tīkla iekārtām** (maiņiem, ugunsmūriem un viršiem mēdz būt drošības caurumi, piemēram, nepilnīga autentifikācijas sistēma.

1. **Nedroša konfigurācija**

**Nedroši lietotāju konti** – lietotāju piekļuves informācija var nedroši tikt pārsūtīta pa tīklu.

**Konti ar viegli uzminamām parolēm**

**Nepareiza konfigurācija interneta servisiem** – piemēram, JavaScript izmantošana nedrošās lapās vai nepareiza termināļa konfigurācija.

**Nedroši noklusētie iestatījumi**

**Nepareizi konfigurētas tīkla iekārtas** – piemēram, nepareizi konfigurēta piekļuve tīkla iekārtām vai nepareizi konfigurēti maršrutēšanas protokoli.

1. **Nedroša politika**

**Negadījumu plāna trūkums** – ļauj rasties haosam, kad dabīga katastrofa vai cilvēka radīts drauds uzbrūk uzņēmumam.

**Nedroša autentifikācijas politika** – nedroša paroļu politika vai noklusētās paroles

**Saglabāta piekļuve izbijušiem darbiniekiem**

* **Nepiemērota vide** – pārāk augsta vai zema temperatūra vai gaisa mitrums
* **Elektriskie bojājumi** – sprieguma pīķis, nepietiekams spriegums, elektrības padeves zudums
* **Nepietiekama uzturēšana** – rezerves detaļu trūkums, nepareizi instalēti kabeļi

Pastāv trīs ļaunatūras tipi:

1. **Vīrusi** – vīrusi izplatās, pievienojot savu kodu citām programmām, saistīti ar izpildāmu failu, kas nozīmē, ka tie nevar strādāt līdz lietotājs nav palaidis saistīto failu.
2. **Tārpi** – cenšas izplatīties uz vairākām iekārtām, taču tārpi neprasa lietotāja iesaisti, lai izplatītos.
3. **Trojas zirgi** – neinficē citas programmas, bet gan replicē savu kodu.

**Datu un konfigurācijas failu kopiju veidošana** (veidot bieži, pārbaudīt pareizību un šifrēt)

**Programmatūras atjauninājumu politika** (anti-ļaunatūras programmatūrai jābūt atjauninātai)

**Autentifikācija, autorizācija un ierakstu veidošana** jeb AAA (kurš drīkst piekļūt, ko piekļuvējs drīkst darīt un ko viņš ir darījis)

**Ugunsmūri** (novērš nevēlamas datu plūsmas, var ļaut piekļuvi tikai daļai tīkla (DMZ)) filtrē:

* **Paketes** (pēc IP un MAC adresēm)
* **Lietojumprogrammas** (pēc portiem)
* **URL** (pēc konkrētiem atslēgas vārdiem)
* **“*Stateful packet inspection*”** (tikai atbildes uz iekšējo iekārtu prasījumiem)

**Satura rādītājs**

[1. OSI, TCP/IP modeļi 1](#_Toc185619130)

[2. Vītā pāra kabeļi 2](#_Toc185619131)

[3. Optisko kabeļu datu pārraides vide 3](#_Toc185619132)

[4. Bezvadu lokālie tīkli 4](#_Toc185619133)

[5. Decimālā, binārā un heksadecimālā skaitīšanas sistēmas 5](#_Toc185619134)

[6. Ethernet protokols 6](#_Toc185619135)

[7. Ethernet komutēšana (*switching*) 7](#_Toc185619136)

[8. IPv4 protokols 8](#_Toc185619137)

[9. IPv6 protokols 9](#_Toc185619138)

[10. IP virzīšana (routing) 10](#_Toc185619139)

[11. TCP protokols 11](#_Toc185619140)

[12. UDP protokols 12](#_Toc185619141)

[13. Populārākie lietojumu slāņa protokoli un pielietojumi 13](#_Toc185619142)

[14. Enkapsulācijas un dekapsulācijas procesi 14](#_Toc185619143)

[15. DNS sistēma 15](#_Toc185619144)

[16. Kādas datortīkla plūsmas notiek, atverot Tīmekļa vietni 16](#_Toc185619145)

[17. Cisco IOS lietošana un iekārtas konfigurēšana 17](#_Toc185619146)

[18. Datoru tīklu drošība 18](#_Toc185619147)