

UNIVERZA V LJUBLJANI

Fakulteta za elektrotehniko

Stanislav Kovačič

Komunikacije v avtomatiki

(študijsko gradivo)

Ljubljana, 25. oktober 2001

Kazalo

1	Uvod	1
1.1	Osnovni gradniki komunikacijskih omrežij	1
1.2	Osnovne oblike omrežij	3
1.3	Smernost prenosa	4
1.4	Sinhronost prenosa	5
1.5	Podatek in informacija	8
1.6	Bit in binarni simbol	8
1.7	Množina informacije in entropija signala	9
1.8	Informacijski pretok in kapaciteta kanala	11
1.9	Baud in bit na sekundo	12
2	Arhitektura omrežij	14
2.1	Slojnost omrežij	14
2.2	ISO OSI referenčni model	16
2.3	OSI, storitve, protokoli in načelo ovojnice	20
2.4	Primeri mrežnih arhitektur	24
3	Elementi fizičnega sloja	27
3.1	Oblike digitalnih signalov	27
3.2	Modemi in modulacije	34
3.3	Nekateri standardi fizičnega sloja	50
4	Elementi podatkovnega sloja	59
4.1	Okvirjenje	60
4.2	Nadzor nad napakami in nad pretokom podatkov	63
4.3	Vrednotenje podatkovnih protokolov	70
4.4	Odkrivanje in popravljanje napak	83

4.5	Ciklično preverjanje	95
4.6	Primeri protokolov podatkovnega sloja	107
5	Dostop do prenosnega sredstva in lokalna omrežja	113
5.1	Protokoli tipa ALOHA	116
5.2	Izkoristek ALOHA	119
5.3	Protokoli tipa CSMA	120
5.4	Protokol CSMA/CD	121
5.5	Izkoristek protokola CSMA/CD	123
5.6	Protokoli brez nevarnosti trčenja	125
5.7	Lokalna omrežja in standardi IEEE 802	127
5.8	FDDI	145
5.9	DQDB in IEEE 802.6	148
6	Industrijska omrežja	150
6.1	Profibus	153
6.2	P-Net	162
6.3	CAN	164
6.4	Interbus	170
7	Elementi mrežnega sloja	175
7.1	Omrežje in storitve	176
7.2	ARP in RARP	177
7.3	Notranja zgradba mrežnega sloja	178
7.4	Usmerjanje	179
7.5	Algoritmi usmerjanja	182
8	Elementi višjih slojev	193
8.1	Zgoščevanje podatkov	193

8.2 Prikrivanje podatkov	205
A Verjetnostni račun	219
B Naključne spremenljivke	223
C Entropija	230

1 Uvod

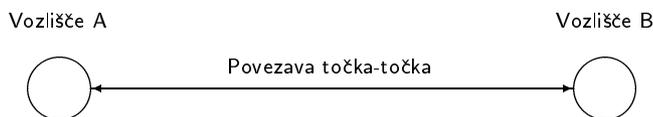
Osnovni problem na področju vodenja krajevno porazdeljenih sistemov je prenos informacij med krajevno ločenimi procesi ali pa med posameznimi deli istega procesa. Ni dolgo tega, kar se je ta problem reševal z nalašč za določen primer uporabe razvitimi komunikacijskimi sistemi. Takšni sistemi, ki so bili načrtovana za čisto specifične potrebe, so se s časom izkazali za toge in neprilagodljive. Razvoj tako imenovanih “inteligentnih” naprav, to je senzorjev, aktuatorjev, krmilnikov in podobnih naprav, ki so sposobne samostojno zajemati in postavljati signale, prenos velikih količin raznovrstnih podatkov, prožnost proizvodnje, združevanje proizvodnih in poslovnih informacijskih sistemov ter sprejemljivi stroški izvedbe in vzdrževanja so zahteve, ki jih takšni komunikacijski sistemi težko zagotovijo. V zadnjih letih je napredoval razvoj komunikacijskih tehnologij vseh vrst. Njihova morda najbolj pomembna lastnost je, da so splošnega značaja. V sodobnih komunikacijskih sistemih je postal računalnik nepogrešljiv sestavni del, prinesel pa je tudi nov način razmišljanja.

Kljub temu, da se komunikacijski sistemi bistveno razlikujejo v tem, kakšno informacijo prenašajo in čemu so namenjeni, pa temeljijo na podobnih osnovnih načelih. Ta načela bomo skušali spoznati v tem delu.

1.1 Osnovni gradniki komunikacijskih omrežij

Osnovna elementa omrežja sta *vozlišče* (ang. Node) in *povezava* (ang. Link). Vozlišče je naprava, ki v omrežju opravlja komunikacijske naloge. Večkrat je za vozlišče bolj primeren izraz *postaja*. Postaja je vozlišče, ki opravlja poleg komunikacijskih še druge naloge, ki niso neposredno vezane na komunikacijo, na primer daljinska postaja, delovna postaja, grafična postaja, ipd. V tem primeru je vozlišče sestavi del postaje, ki je priključena na omrežje.

Komunikacijsko omrežje je sistem med seboj povezanih vozlišč. Osnovni način povezovanja vozlišč je povezovanje *točka-točka*¹ (ang. Point-To-Point). Povezava točka-točka neposredno povezuje dve sosednji vozlišči med seboj, kot je narisano na sliki 1.



Slika 1: Povezava točka-točka.

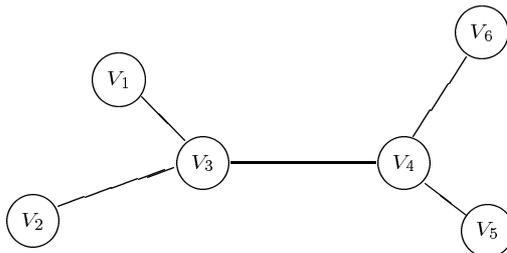
Komunikacijsko omrežje torej združuje množico vozlišč. V primeru, da je

¹Tudi točka s točko ali točka v točko.

vsako vozlišče neposredno povezano s svojo povezavo z vsakim drugim vozliščem, nastane popolnoma povezano omrežje. V primeru, da je v popolnoma povezanem omrežju N vozlišč, gre iz vsakega vozlišča $(N-1)$ povezav, skupno število povezav pa je:

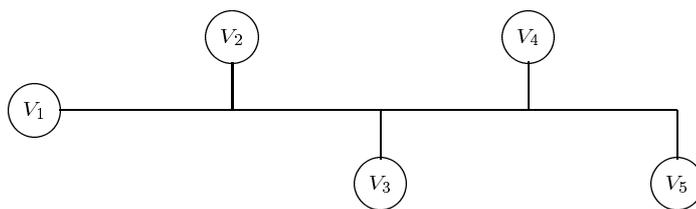
$$(N-1) + (N-2) + (N-3) + \dots + 2 + 1 = \frac{(N-1) \times N}{2}.$$

Popolnoma povezano omrežje je smiselno za majhno število vozlišč. Z naraščanjem števila vozlišč število povezav strmo narašča. V delno povezanem omrežju si dve ali več vozlišč deli skupno povezavo, kot je narisano na sliki 2.



Slika 2: Končna (V_1, V_2, V_5 in V_6) in vmesna (V_3 in V_4) vozlišča v delno povezanem omrežju.

V narisanim primeru vozlišče V_1 komunicira z vozliščem V_6 posredno preko vozlišč V_3 in V_4 . Komunikacijsko omrežje v splošnem sestavljata dve vrsti vozlišč: *končna* in *vmesna*² vozlišča. V končnem vozlišču informacija nastaja ali pa se koristi. V vmesnih vozliščih informacija niti ne nastaja niti se ne koristi. Vmesno vozlišče deluje kot posrednik informacije: informacijo sprejme, jo po potrebi začasno shrani, preoblikuje in pošlje naprej v zahtevani smeri. Na primer, končna vozlišča so telefoni, senzorji, aktuatorji, daljinske postaje, središče vodenja in tako dalje. Vmesna vozlišča so ponavljalniki (ang. repeaters), mostovi (ang. bridges), usmerjevalniki (ang. routers), koncentratorji, prehodi (gateways), ipd. Zaporedje povezav od enega končnega vozlišča preko vmesnih vozlišč do drugega končnega vozlišča sestavlja prenosno pot.



Slika 3: Večtočkovno povezovanje vozlišč.

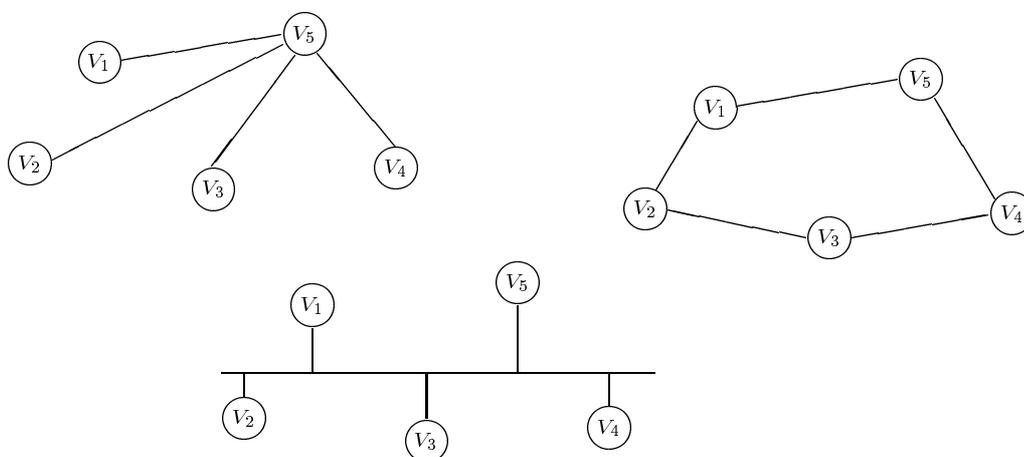
Druga možnost za povezovanje vozlišč je večtočkovno povezovanje (ang. Multi-point ali Multi-drop). Slika 3 prikazuje takšen način povezovanja. Obstajata torej dve osnovni vrsti povezovanja: povezovanje točka-točka in večtočkovno

²Tudi komunikacijska vozlišča.

povezovanje. Večtočkovno povezovanje se je zaradi številnih dobrih lastnosti močno uveljavilo v industrijskih omrežjih.

1.2 Osnovne oblike omrežij

S povezovanjem vozlišč nastane omrežje določene oblike. Osnovne oblike (topologije) delno povezanih komunikacijskih omrežij so: zvezda, obroč (tudi znaka) in vodilo. Na sliki 4 so skicirane osnovne oblike omrežij. Tipično komunikacijsko omrežje je *splošne ali mešane oblike* in je kombinacija omrežij osnovnih oblik (Slika 5).

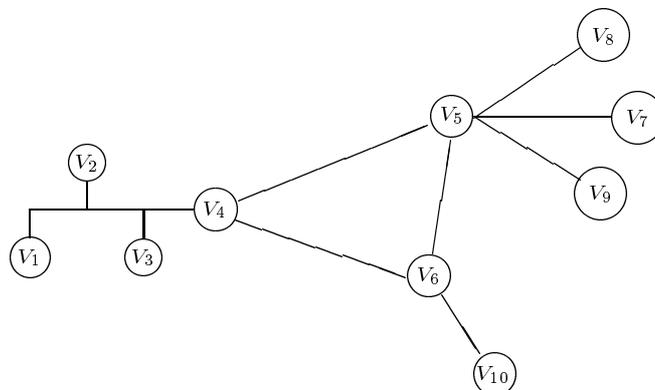


Slika 4: Osnovne oblike omrežij: zvezda, obroč (zanka) in vodilo.

V zvezdastem in zankastem omrežju so povezave med vozlišči tipa točka-točka. V omrežju zvezdaste oblike (ang. Star) komunicirajo zunanja (končna) vozlišča preko osrednjega (vmesnega) vozlišča. Pretok informacije v omrežju ureja osrednje vozlišče. V omrežju, ki je po obliki obroč oz. zanka (ang. Ring, Loop) informacija kroži v predvideni smeri od vozlišča do vozlišča. Nadzor nad pretokom podatkov v obroču je lahko centraliziran ali decentraliziran. V decentraliziranem obroču so si s tega stališča vsa vozlišča enakovredna, v centraliziranem pa ureja pretok podatkov eno od vozlišč.

V omrežju tipa vodilo (ang. Bus) se koristi večtočkovno povezovanje, zato ga imenujemo tudi večtočkovno omrežje. Večtočkovno omrežje je tipičen primer omrežij z množičnim dostopom (ang. Multiple-Access), ki se med seboj razlikujejo ravno po tem, kako se vozliščem dodeljuje ali kako vozlišča dostopajo do skupnega prenosnega sredstva (medija). Večtočkovna omrežja se zaradi številnih dobrih

lastnosti veliko uporabljajo v industrijskih okoljih za povezovanje enostavnejših naprav (senzorjev, aktuatorjev, programljivih krmilnikov, i.t.d.).

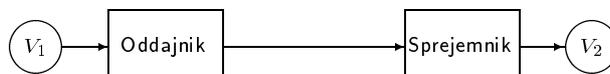


Slika 5: Splošna oblika omrežja. Vozlišča V_4, V_5 in V_6 so vmesna vozlišča in tvorijo komunikacijski podsistem, na katerega so vezana končna vozlišča $V_1, V_2, V_3, V_7, V_8, V_9$ in V_{10} .

Večja omrežja se le redko kdaj javljajo v eni od osnovnih oblik. Omrežje, ki je sestavljeno iz omrežij zvezdaste in večtočkovne oblike ter obroča, je mešano omrežje.

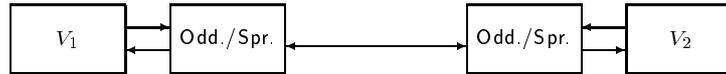
1.3 Smernost prenosa

Glede na možnost sočasnosti prenosa podatkov v obe smeri med dvema oddaljenima napravama govorimo o treh vrstah prenosa. Imenujemo jih enosmerni prenos ali simpleks, pol-dvosmerni ali poldupleks (ang. half-duplex ali HDX) in dvosmerni prenos ali dupleks (ang. full-duplex ali FDX). Prenos vrste *simpleks* omogoča prenos podatkov le v eni smeri, kot to ponazarja slika 6. Tak način prenosa pride v poštev pri zbiranju podatkov.



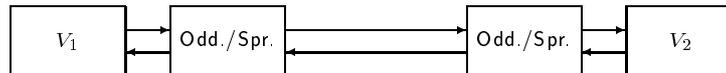
Slika 6: Prenos podatkov vrste simpleks.

Prenos podatkov vrste *poldupleks* omogoča prenos podatkov v obe smeri, vendar pa ne istočasno. Naprava, ki je sporočilo oddala, se mora 'preklopiti' (spremeniti način delovanja) z oddaje na sprejem in počakati na odgovor. Ko sprejme odgovor, se ponovno preklopi na oddajo in odda naslednje sporočilo, glej sliko 7. Informacija se torej prenaša enkrat v eni smeri, drugič v drugi smeri.



Slika 7: Prenos podatkov vrste poldupleks.

Prenos podatkov vrste dupleks (ali polni dupleks), omogoča prenos podatkov v obeh smereh istočasno, glej sliko 8. Realiziramo ga s prenosom po dveh vodih (štirižično) ali po enem samem vodu (dvožično) z multipleksiranjem.

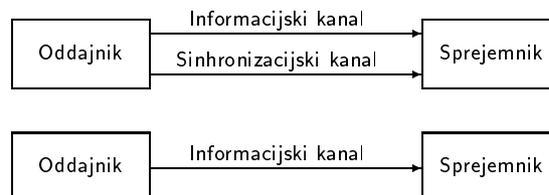


Slika 8: Prenos podatkov vrste dupleks.

1.4 Sinhronost prenosa

Pri prenosu digitalnih signalov se mora sprejemna naprava sinhronizirati z oddajno napravo. To pomeni, da mora natančno ugotoviti, kdaj se začne in kdaj konča posamezen bit, kdaj se začne in konča posamezen znak (skupina bitov), pa tudi kdaj se sporočilo začne in kdaj se konča. Glede na to, kako se sprejemnik sinhronizira z oddajnikom, govorimo o dveh vrstah prenosa, sinhronem in asinhronem prenosu (imenujejo ju tudi sinhronski in asinhronski prenos). Pri sinhronem prenosu tečeta generatorja takta sprejemnika in oddajnika *sinhrono*, torej se morata ujemati tako po frekvenci kot po fazi. Zato se od oddajnika do sprejemnika poleg informacijskega signala prenaša tudi sinhronizacijski signal, ali pa je informacijski signal tak, da se da iz njega pridobiti informacijo o taktu oddajnika (slika 9).

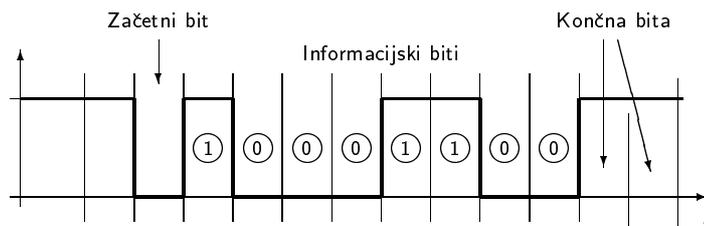
Pri asinhronem načinu prenosa tečeta sprejemni in oddajni generator asinhrono, informacija o frekvenci in fazi oddajnika pa se ne prenaša. Zato pa morata biti frekvenci sprejemnega in oddajnega generatorja enaki.



Slika 9: Razlika med sinhronim (zgoraj) in asinhronim (spodaj) načinom prenosa.

1.4.1 Asinhroni prenos

Pri asinhronem (ali asinhronskem) prenosu se sinhronizacija sprejemnika z oddajnikom opravlja na nivoju krajšega zaporedja bitov (na nivoju osmih bitov ali znaka). Vsak znak v sporočilu ima svoj 'okvir' ali takoimenovane 'sinhronizacijske bite'. To so začetni bit (START bit) in eden, eden in pol ali dva končna bita (STOP bita), glej sliko 10.

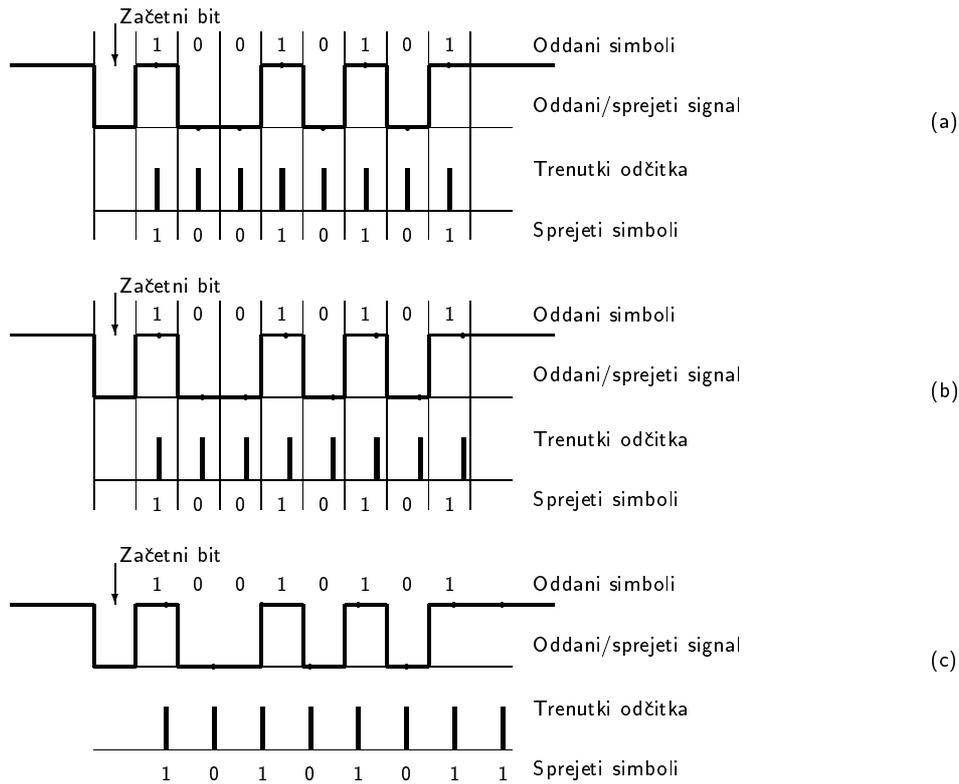


Slika 10: Primer signala za asinhroni prenos z začetnim, z osmimi informacijskim biti in dvema končnima bitoma.

Hitrost oddajanja se mora ujemati s hitrostjo sprejemanja, ujemati pa se mora tudi število bitov na znak in število končnih bitov oddajnika in sprejemnika. Ker ima vsak znak označen svoj začetek in svoj konec so lahko časovni presledki (premori) med zaporednimi znaki v sporočilu poljubno dolgi. Asinhroni prenos se uporablja pri nižjih hitrostih prenosa. Primeren je tam, kjer prihajajo podatki v nepredvidljivih časovnih presledkih. Njegova slabost je nizek izkoristek prenosne poti zaradi časovnih presledkov med zaporednimi znaki in zaradi relativno velikega števila sinhronizacijskih (odvečnih) bitov. Sprejemnik se sinhronizira z oddajnikom tedaj, ko zazna prisotnost začetnega bita (slika 11.a), sicer pa teče njegov generator takta, ki določa trenutke odčitkov, popolnoma asinhrono z oddajnikom. Če se hitrost sprejemnika preveč razlikuje od hitrosti oddajnika, se sprejeti signal odčita napačno, (glej sliki 11.b in 11.c). Končni biti so potrebni za izravnavo razlike med hitrostjo oddajnika in hitrostjo sprejemnika. Pravzaprav določajo najkrajši časovni presledek med zaporednima znakoma, sicer pa je presledek poljubno daljši.

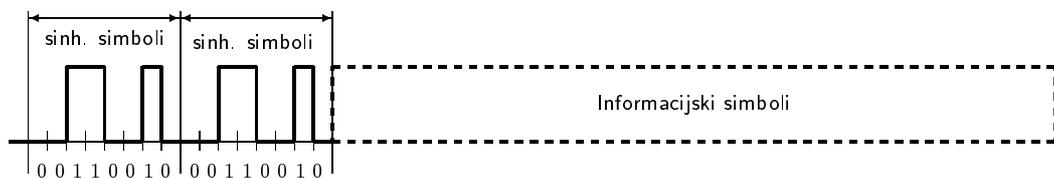
1.4.2 Sinhroni prenos

Pri sinhronem (ali sinhronskem) prenosu podatkov prenašamo znak za znakom brez vmesnih presledkov, sinhronizacija sprejemnika z oddajnikom pa se vzpostavi pred prenosom daljšega zaporedja znakov (okvirja) in se nato vzdržuje cel čas prenašanja. Za sinhronizacijo služi predpisano zaporedje uvodnih znakov, ki jim pravimo sinhronizacijski znaki. Oddajna naprava pred začetkom oddajanja vsebine okvirja najprej oddaja zaporedje enega ali več sinhronizacijskih znakov,



Slika 11: Asinhroni prenos podatkov. a) Oddajna hitrost je enaka sprejemni hitrosti: sprejemnik pravilno odčita informacijski signal. b) Oddajna hitrost je višja od sprejemne: odčitki sprejemnika so še pravilni. c) Oddajna hitrost je previsoka: sprejemnik napačno odčita sprejeti signal.

sinhronizacijske znake pa po potrebi odda tudi sredi oddajanja samega okvirja. Sinhronizacijski znaki so domenjani, tako da jih sprejemna naprava 'spozna' in loči od koristne informacije. Ko sprejemna naprava sprejme sinhronizacijski znak, ga izloči iz koristne informacije in skuša sinhronizirati svoj generator urnih impulzov z generatorjem oddajnika oziroma nastaviti hitrost (in fazo) urnih impulzov svojega generatorja.



Slika 12: Sinhroni prenos podatkov: najprej gredo sinhronizacijski simboli in za njimi daljše zaporedje informacijskih simbolov.

Pred začetkom oddajanja okvirja (ang. frame) se znaki zbirajo v vmesnem pomnilniku oddajne naprave in naprava med tem ne potrebuje prenosne poti (kanala). Šele ko je okvir v celoti pripravljen, ga oddajna naprava odda.

Dolžina okvirja je lahko od nekaj znakov do nekaj tisoč znakov. To je odvisno od velikosti medpomnilnikov, od lastnosti prenosne poti, od hitrosti prenosa in izvedbe komunikacijskih naprav. Možna je stalna ali spremenljiva dolžina okvirjev. Spremenljiva dolžina omogoča boljšo izrabo prenosne poti. V tem primeru mora biti na koncu okvirja še znak za konec okvirja ali pa mora biti na začetku okvirja podatek o številu znakov. Običajno sledi končnemu znaku še znak za preverjanje pravilnosti prenosa.

1.5 Podatek in informacija

V vsakodnevem pogovoru največkrat ne razlikujemo med podatkom in informacijo, vendar med njima obstaja razlika, ki je včasih zelo pomembna. Informacijo v tehniki opredelimo z zmanjšanjem nedoločenosti oziroma negotovosti o določeni stvari. Z zmanjšanjem nedoločenosti torej nastane informacija. Podatek ni nujno povezan z zmanjšanjem nedoločenosti. Na primer, izjava ob lepem vremenu *danes je lepo vreme*, ni informacija. Je pa podatek, le da je njegova informacijska vsebina majhna, praktično nič. Rečemo lahko, da je podatek nosilec informacije oziroma da vsebuje informacijo, ki pa je lahko v skrajnem primeru tudi nič. Podatek je simboličen zapis informacije.³

1.6 Bit in binarni simbol

Izraz bit ima dva pomena. Po eni strani je *bit* enota za množino informacije. Informacijo enega bita dobimo z odgovorom na vprašanje, na katerega sta možna dva enakovredna odgovora. Ali, informacijo enega bita nosi signal z dvema enakovrednima stanjema, t.j. dvovrednosti ali binarni signal.

Bit ima še drugi pomen. Z njim na kratko imenujemo binarni simbol (ang. BIT = BInary digiT⁴). Binarni simbol je eden od dveh možnih simbolov binarne abecede. Z njim je možno zapisati 1 bit informacije.⁵ Tudi mi bomo izraz bit uporabljali v obeh oblikah, kot enoto za množino informacije in kot ime za binarni simbol, le v primeru dvoumnosti se bomo zatekli k daljšemu izrazu binarni simbol.

³Slovar slovenskega knjižnega jezika: (1) Podatek je dejstvo, ki nam o določeni stvari kaj pove ali se nanjo nanaša, (2) Podatek je informacija v obliki primerni za računalnik.

⁴Digit v angleškem jeziku pomeni prst ali cifro pri štetju do deset. Binary Digit je dvojiška cifra oziroma dvojiški simbol.

⁵Ko rečemo, da je nek podatek sestavljen iz osmih bitov, s tem ne mislimo, da vsebuje informacijo osmih bitov, ampak da ga sestavlja osem binarnih simbolov. Res pa je, da bi lahko vseboval tudi toliko informacije.

1.7 Množina informacije in entropija signala

Hartley je leta 1928⁶ predlagal, da bi signalu S , ki zmore zavzeti eno od N diskretnih vrednosti ali stanj s_i , ($i = 1, 2, \dots, N$) pripisali informacijsko vsebino

$$I(S) = \log_2 N \text{ [bitov]}. \quad (1)$$

Po definiciji (1) je informacijska vsebina tem večja, čim več stanj zmore zavzeti signal,

$$N_1 < N_2 \implies (I_1 = \log_2 N_1) < (I_2 = \log_2 N_2). \quad (2)$$

Vzemimo, da signal z N stanji opazujemo v G trenutkih, tako da poznamo G med seboj neodvisnih realizacij $S^{(t)}$ signala,

$$S_G = S^{(1)}, S^{(2)}, \dots, S^{(t)}, \dots, S^{(G)}. \quad (3)$$

Možnih je N^G različnih in enako verjetnih načinov, kako se to zgodi. Zato je

$$I(S_G) = \log_2 N^G = G \times \log_2 N = G \times I(S). \quad (4)$$

Izraz (4) pravi, da iz zaporedja G med seboj neodvisnih opazovanj signala S dobimo G -krat toliko informacije, kot jo dobimo iz enega samega opazovanja. To smo tudi pričakovali.

Informacijsko vsebino signala se da razložiti tudi takole. Vzemimo, da zmore signal S zavzeti eno od N stanj oziroma vrednosti, vendar zaenkrat ne vemo, v katerem od stanj se v resnici nahaja. O tem obstaja dvom ali nedoločenost. Čim več je takih stanj, tem večji je dvom (težje uganemo, v katerem stanju je signal), zato pa se nedoločenost tem bolj zmanjša, ko zvemo, v katerem stanju je. Z drugimi besedami, ko zvemo, v katerem od stanj se signal dejansko nahaja, dobimo o signalu informacijo. Sprejeta informacija je tem večja, čimbolj se zmanjša nedoločenost, oziroma čim večja je bila nedoločenost pred tem.

Hartleyevi izsledki veljajo za signale z enakoverjetnimi stanji. E.C. Shannon je Hartleyeve izsledke posplošil in leta 1948 [1] postavil temelje sodobne teorije informacij.

Shannon je izhajal iz predpostavke, da je informacijska vsebina signala (in z njo povezana nedoločenost) odvisna od verjetnostne porazdelitve stanj. Stanja z veliko verjetnostjo so manj nedoločena in zato nosijo manj informacije od tistih, ki imajo majhno verjetnost. Nedoločenost *posameznega stanja* signala je definiral z logaritmom njegove verjetnosti:

$$h(s_i) = \log_2 \frac{1}{p_i}, \quad (i = 1, 2, \dots, N). \quad (5)$$

⁶R.V.L. Hartley, *Transmission of Information*, *Bell System Technical Journal*, vol.7, pp.535–563, 1928.

Stanja z manjšo verjetnostjo so po tej definiciji bolj nedoločena, jih je težje uganiti ali napovedati in zato nosijo tudi več informacije. Shannon je definiral še povprečno nedoločenost $H(S)$ na stanje signala S (naključne spremenljivke – signal),

$$H(S) = \sum_{i=1}^N p_i \log_2 h(s_i) = \sum_{i=1}^N p_i \log_2 \frac{1}{p_i}. \quad (6)$$

Ker je $\log_2 1/p_i = -\log_2 p_i$, dobimo:

$$H(S) = - \sum_{i=1}^N p_i \log_2 p_i \quad (7)$$

Kadar nas dejanska vrednost izraza (7) ne zanima, hočemo pa vseeno poudariti, da je njegova vrednost odvisna od porazdelitvenega zakona, zapišemo zraven porazdelitveni zakon:

$$H(S) = H(p_1, p_2, \dots, p_i, \dots, p_N). \quad (8)$$

Povprečno nedoločenost na stanje imenujemo *entropija* signala. Entropija je *nenegativna* funkcija verjetnostne porazdelitve signala, in narašča z N . Velja tudi pravilo seštevanja entropij. Kaj lahko se pričamo, da je entropija $H(S_G)$ zaporedja G (*neodvisnih*) opazovanj signala:

$$S_G = S^{(1)}, S^{(2)}, \dots, S^{(t)}, \dots, S^{(G)} \quad (9)$$

G -krat večja od entropije enega samega opazovanja:

$$H(S_G) = G \times H(S). \quad (10)$$

Entropija je merilo za 'nered' signala. Entropija je pri danem N največja, ko je signal najbolj 'neurejen'. To je tedaj, ko so vsa stanja enako pričakovana - enako verjetna. V tem primeru velja:

$$H(S) = - \sum_{i=1}^N p_i \log_2 p_i = - \sum_{i=1}^N p_i \log_2 \frac{1}{N} = \log N, \quad (11)$$

kar sovpada s Hartleyevo definicijo.

Informacijska vsebina signala ali *množina informacije* je povezana z entropijo. Čim večja je entropija signala, tem več informacije nosi signal. Naj bo njegova entropija v danem trenutku enaka $H(S)$. Ko spoznamo signal, se njegova nedoločenost zmanjša na nič. S tem dobimo (v povprečju)

$$I(S) = H(S), \quad (12)$$

informacije. To pomeni, da bi pri opazovanju signala skozi daljši čas dobili z vsako realizacijo signala v povprečju $H(S)$ informacije.

1.8 Informacijski pretok in kapaciteta kanala

Leta 1924 je H. Nyquist dokazal, da se da poljuben frekvenčno omejen signal s frekvenčno vsebino 0 do F Hz popolnoma rekonstruirati, če signal vzorčimo v enakomernih časovnih presledkih $t_s = 1/(2 \times F)$. Edini pogoj za popolno rekonstrukcijo je, da vzorci signala niso moteni s šumom. Vzorčenje z višjo frekvenco ne prispeva h kvaliteti rekonstrukcije. Povedano drugače, $2 \times F$ vzorcev na sekundo vsebuje vso informacijsko vsebino signala. Edini pogoj za popolno rekonstrukcijo je, da vzorci signala niso moteni s šumom. Nyquistovo spoznanje je temeljnega pomena za prenos informacije po brezšumnem informacijskem kanalu⁷ (prenosni poti) z omejeno frekvenčno širino. Da je signal s frekvenčno vsebino 0 – F še možno neokrnjeno prenašati po kanalu, mora biti frekvenčna širina kanala enaka F . Tak signal popolnoma opišemo z $2 \times F$ signalnimi elementi na sekundo. Če je za vsak element signala možnih V diskretnih vrednosti (stanj), lahko vsak nosi največ $\log_2 V$ bitov informacije. Najvišja možna hitrost prenašanja informacije potem je:

$$C = 2 \times F \times \log_2 V \quad \frac{\text{bitov}}{\text{sekundo}}. \quad (13)$$

Najvišjo možno hitrost prevajanja informacije imenujemo kapaciteta informacijskega kanala in jo običajno označimo s C . Na primer, po brezšumni prenosni poti s frekvenčno širino 3000 Hz (frekvenčna širina telefonskega voda) teoretično ne moremo prenašati dvovrednostnega (binarnega) signala z višjo hitrostjo od

$$C = 2 \times 3000 \times \log_2 2 = 6000 \frac{b}{s},$$

v praksi pa je ta hitrost zaradi prisotnosti šuma in nepopolnosti komunikacijskih naprav še nižja.

Kapaciteto kanala C moramo razumeti kot lastnost, ki je dana z izvedbo prenosne poti in ni odvisna od dejanske hitrosti prenosa. Dejansko hitrost prenosa narekuje oddajna naprava in je lahko poljubno nižja od kapacitete kanala. Oddajna naprava ne sme nikoli oddajati hitreje kot je kapaciteta kanala. V nasprotnem primeru je izgubljanje informacije neizbežno.

Enačba (13) pravi, da se da doseči višjo hitrost prenosa pri isti frekvenčni širini kanala, če za prenos koristimo večvrednostni signal. Denimo, da je $V = 256$. V tem primeru lahko vsak signalni element nosi osem bitov informacije. Kapaciteta kanala pri enaki frekvenčni širini kot prej potem je

$$C = 2 \times 3000 \times \log_2 256 = 6000 \times 8 = 48000 \frac{b}{s}.$$

⁷Informacijski kanal je sredstvo, medij ali predpis za prenos informacije skozi prostor in/ali čas.

Je pa v tem primeru na sprejemni strani težje razpoznati odposlano vrednost signala - signal je težje ločiti od šuma. Enačba (13) ne pove ničesar o tem, kako moč šuma vpliva na kapaciteto kanala. Dejstvo je, da z naraščanjem moči šuma kapaciteta upada.

C. E. Shannon je ugotovil (1948), da je kapaciteta kanala s pasovno širino F v prisotnosti aditivnega gaussovega šuma (oziroma ob predpostavki, da sta tako signal kot šum porazdeljena normalno) in pri razmerju *moč signala/moč šuma* = S/N podana z enačbo:

$$C = F \times \log_2 \left(1 + \frac{S}{N} \right). \quad (14)$$

Iz enačbe vidimo, da je pri manjši frekvenčni širina kanala manjša tudi njegova kapaciteta. Podobno, če moč šuma raste, potem kapaciteta pada. Kapaciteto kanala se da povečati pri istem razmerju $\frac{S}{N}$ z večanjem frekvenčne širine ali pa pustimo frekvenčno širino nespremenjeno in povečamo moč signala ali pa zmanjšamo šum. Pri tem ni važno, koliko nivojev (stanj) zmore zavzeti informacijski signal. Shannonova formula kapacitete kanala neposredno ne povezuje s številom možnih stanj (nivojev) signala, vendar se jo da tolmačiti tudi na ta način. Zapišimo Shannonovo enačbo še enkrat v nekoliko drugačni obliki:

$$C = F \times \log_2 \left(1 + \frac{S}{N} \right) = 2 \times F \log_2 \sqrt{\frac{S+N}{N}}$$

Enačni (13) in (14) postaneta identični za $V = \frac{\sqrt{S+N}}{\sqrt{N}}$ iz česar sledi, da pri dani moči signala moč šuma pravzaprav določa število razpoznavnih stanj signala.

Za številčni primer vzemimo, da je frekvenčni pas kanala 3000 Hz in da je razmerje signal/šum = 1000 (nekako tako, kot pri telefonskem vodu). Teoretično največja možna hitrost prenosa podatkov po kanalu znaša:

$$C = 3000 \times \log_2(1 + 1000) \approx 3000 \times 10 \approx 30000 \frac{b}{s}.$$

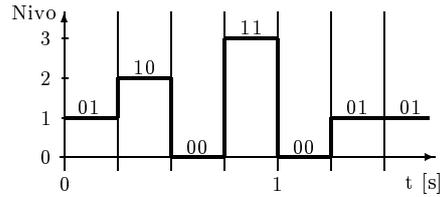
Gornjo formulo je treba razumeti kot teoretično zgornjo mejo hitrosti prenosa, kar pa ne pomeni, da je takšno hitrost tudi enostavno doseči. S sodobnimi postopki kodiranja in modulacije se danes dosega 33600 [b/s]⁸, dejanska hitrost prenosa pa seveda zavisi od kvalitete telefonskih linij.

1.9 Baud in bit na sekundo

Število (možnih) sprememb informacijskega signala na časovno enoto podajamo v **baudih** (izgovorimo bod).⁹ V baudih izražamo hitrost spreminjanja signala

⁸Priporočila ITU-T V.90, ITU-T V.34

⁹Baud (**Bd**) je dobil ime po francoskem strokovnjaku E. Baudot-u kot priznanje za njegove dosežke na področju telekomunikacij v preteklem stoletju.



Slika 13: Štirivrednostni signal, ki zmore manjati stanje štirikrat na sekundo, signalna hitrost je 4 Baude. Ker vsako od štirih možnih stanj nosi dva bita informacije, je hitrost prenosa v bitih na sekundo enaka $4 \times 2 = 8$.

oziroma število signalnih elementov na sekundo in ji zato večkrat pravimo tudi signalna ali modulatorska hitrost. Ta nam ne pove veliko o dejanski hitrosti prenosa informacije. Na hitrost prenosa informacije v bitih na sekundo namreč ne vpliva samo hitrost spreminjanja (frekvenca) signala, temveč tudi to, koliko informacije nosi posamezen signalni element, glej enačbo (13). Zato je hitrost prenosa izražena v baudih le izjemoma številčno enaka hitrosti prenosa, ki jo podamo v bitih na sekundo.

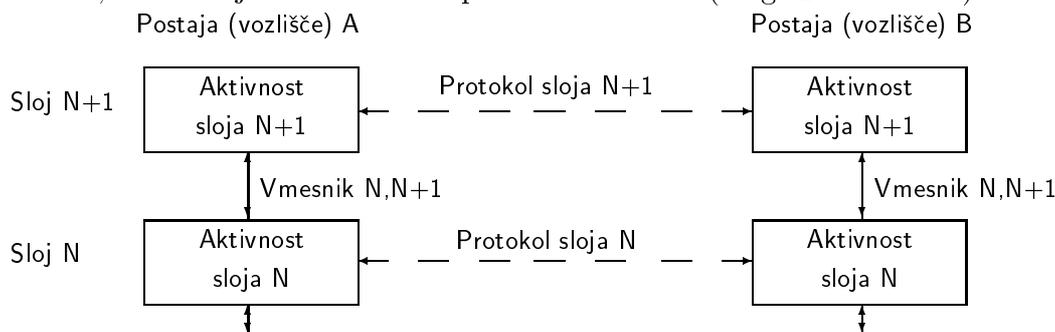
Odnos med baudi in biti na sekundo smo ponazorili na sliki 13. Vsako sekundo oddamo štiri signalne elemente. Signal zmore spremeniti stanje štirikrat na sekundo in hitrost spreminjanja signala je 4 Bd. Vsak signalni element lahko zavzame eno od štirih diskretnih vrednosti (stanj). Z vsakim stanjem zapišemo (zakodiramo) dva binarna simbola (dva bita informacije). Vsako sekundo lahko s takim signalom prenesemo $4 \times 2 = 8$ bitov informacije in hitrost prenosa informacije je zato osem bitov na sekundo.

Torej, ko rečemo, da prenašamo podatke s hitrostjo 2400 baudov, mislimo s tem povedati, da zmore signal spremeniti stanje 2400-krat na sekundo oziroma, da prenašamo 2400 signalnih elementov na sekundo. V primeru, da je signal večvrednosten (denimo $V = 16$), nosi vsak signalni element več bitov informacije (za $V = 16$ štiri bite) in najvišja možna hitrost prenosa v bitih na sekundo postane (štirikrat) višja od tiste, s katero se spreminja signal. V tem primeru bi morali torej povedati, da prenašamo 9600 bitov na sekundo z 2400 baudi. Hitrost prenosa izražena v baudih se številčno ujema s hitrostjo izraženo v bitih samo tedaj, ko je signal dvovrednosten (binaren).

2 Arhitektura omrežij

2.1 Slojnost omrežij

Komunikacijska omrežja sodijo med kompleksne sisteme. Da bi zmanjšali zahtevnost in stroške snovanja, izgradnje, obratovanja, vzdrževanja in uporabe takih sistemov, je večina sodobnih omrežij zasnovana modularno kot sestav *slojev*, pri čemer je vsak višji sloj nadgrajen nad neposredno nižji sloj. Skupna lastnost sodobnih omrežij je slojna urejenost postopkov in naprav. Osrednji element v tej hierarhični večslojni zgradbi omrežja je *sloj* (Ang. Layer), imenovan tudi plast ali nivo (Ang. Level). Število slojev, imena slojev, kot tudi vsebina in funkcija posameznih slojev, se od omrežja do omrežja razlikujejo. Kar je značilno za sloj v vsakem omrežju je predvsem to, da nudi storitve neposredno višjemu sloju, pri tem pa skriva podrobnosti o izvedbi storitev pred uporabnikom storitev. Sloj, ki nudi storitev, se imenuje dajalec storitve (Ang. Service provider). Sloj, ki uporablja storitev, se imenuje koristnik ali uporabnik storitve (Ang. Service user).



Slika 14: Načelo slojnosti omrežja.

Med sosednjima slojema je vmesnik. Vmesnik slojev N in $N + 1$ natančno določa storitve, ki jih sloj N nudi sloju $N + 1$, pa tudi osnovne operacije za izvedbo teh storitev.

Proces sloja N na eni postaji komunicira s procesom sloja N na drugi postaji. Pravimo, da komunikacija v omrežju poteka med istorodnimi procesi (ang. peer-to-peer). Pravila in dogovori, ki jih upoštevata istorodna procesa na obeh straneh na sloju N , imenujemo s skupnim imenom *komunikacijski protokol sloja N*. Prenos podatkov med postajama na istem sloju (v vodoravni smeri) pa je zgolj navidezen. Resnični tok podatkov gre v navpični smeri. Višji sloj preda podatke (skupaj z nadzorno informacijo) naslednjemu nižjemu sloju, ta pa zopet nižjemu. Končno pride na najnižjem sloju po prenosne poti do dejanskega prenosa informacije z ene na drugo stran, kjer gre informacija od sloja do sloja navzgor - v navpični smeri. Dejanski vodoraven prenos informacije je prisoten samo na najnižjem nivoju - na nivoju prenosne poti.

Množica slojev in protokolov ter vmesnikov, oziroma storitev, ki so dane na vmesnikih, določa *arhitekturo omrežja*. Določila mrežne arhitekture morajo zadoščati za realizacijo vseh postopkov in naprav. Sam način izvedbe protokolov in vmesnikov ne sodi k arhitekturi. Ta je z arhitekturnega stališča popolnoma svoboden. Arhitektura omrežja je torej povezana z vprašanjem *kaj*, ne pa *kako*.

2.1.1 Primer večslojnosti komuniciranja

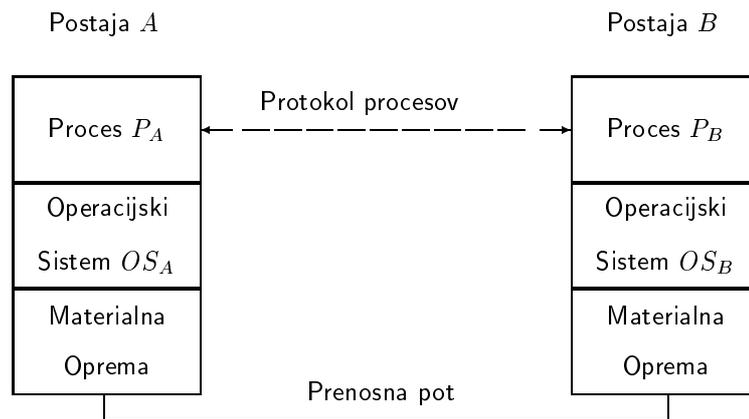
Da bi se znebili občutka nestvarnosti večslojne komunikacije in da bi razjasnili razliko med vodoravno in navpično komunikacijo, si zamislimo naslednji netehnični primer. Denimo, da želi slovenski diplomat D_S komunicirati s finskim diplomatom D_F . Med diplomati obstajajo dogovorjena pravila (protokol diplomatov), ki se jih je morata oba držati. Če želi vsak od diplomatov komunicirati v svojem jeziku, sta potrebna slovenski P_S in finski P_F prevajalec. Prevajalca s prevajanjem nudita storitve (prevajanje) diplomatoma. Kako in v kakšnem jeziku (recimo francoskem) se v resnici sporazumevata prevajalca (protokol prevajalcev), diplomatov ne zanima. Za oba diplomata je važno le, da jima prevajalca omogočata komunikacijo. Prevajalca v tem primeru realizirata storitve, ki jih dajeta diplomatoma (višjemu nivoju) s komunikacijo na svojem nivoju - s protokolom prevajalcev.

Diplomata dejansko ne govorita neposredno drug z drugim, ampak vsak s svojim prevajalcem. V resnici teče informacija v navpični smeri (vmesnik diplomat-prevajalec). Kljub temu se diplomata vedeta kot da govorita neposredno drug z drugim. Dejstvo, da govorita vsak s svojim prevajalcem, je za vsebino njunega pogovora nepomembno.

Tudi za prevajalca obstaja več možnosti komunikacije. Lahko komunicirata eden z drugim po telefonu, po običajni pošti, po elektronski pošti, i.p.d. Na tem, v našem primeru najnižjem nivoju, se zgodi dejanski prenos informacije med diplomatoma.

Poglejmo bolj tehnični primer. Razmislimo, kako razmišlja programer, ki se ukvarja s problemom komuniciranja procesa P_A s procesom P_B na krajevno ločenih postajah A in B , ki sta na nek način povezani, glej sliko 15.

Postaji delujeta pod nadzorom operacijskega sistema (OS). Problematiko komuniciranja bi programer na primer razčlenil na vzpostavljanje zveze med procesoma, na prenos podatkov v obe smeri in na sproščanje zveze, ko ta ni več potrebna. Njegov program bi med drugim verjetno obsegal podprograme z imeni kot so: *VzpostaviZvezo(Proces)*, *PošljiPodatke(Proces)*, *SprejmiPodatke(Proces)* in *SprostiZvezo(Proces)*. Programer pri tem očitno razmišlja tako, kot da procesa P_A in P_B dejansko komunicirata neposredno drug z drugim v vodoravni smeri. Dejstvo, da procesa v resnici komunicirata 'navzdol' z operacijskim sistemom je na tem nivoju abstraktnosti drugotnega pomena. Je pa zagotovo



Slika 15: Večslojnost komuniciranja, vodoraven in navpičen prenos podatkov.

pomembno pri realizaciji podprogramov. Kar vzemimo primer realizacije podprograma *PošljiPodatke(Proces)*. Naloga podprograma je, da poskrbi za prenos določenih podatkov od enega procesa do drugega procesa. Programer za realizacijo podprograma (realizacijo prenosa podatkov) uporablja tisto, kar mu nudi operacijski sistem. Na primer, podatke, skupaj z nadzorno informacijo preda operacijskemu sistemu, ki s krmiljenjem ustrezne materialne opreme poskrbi za dejanski prenos podatkov. Podprogram torej 'pošlje podatke navzdol' operacijskemu sistemu in ne procesu na drugi strani. Kako operacijski sistem poskrbi za dejanski prenos podatkov na drugo stran, je programerju (podprogramu) popolnoma prikrito in je zanj na tem nivoju tudi nepomembno. Po drugi strani pa je s stališča operacijskega sistema popolnoma nepomembno, kaj pomenijo podatki, ki jih prenaša v korist obeh procesov.

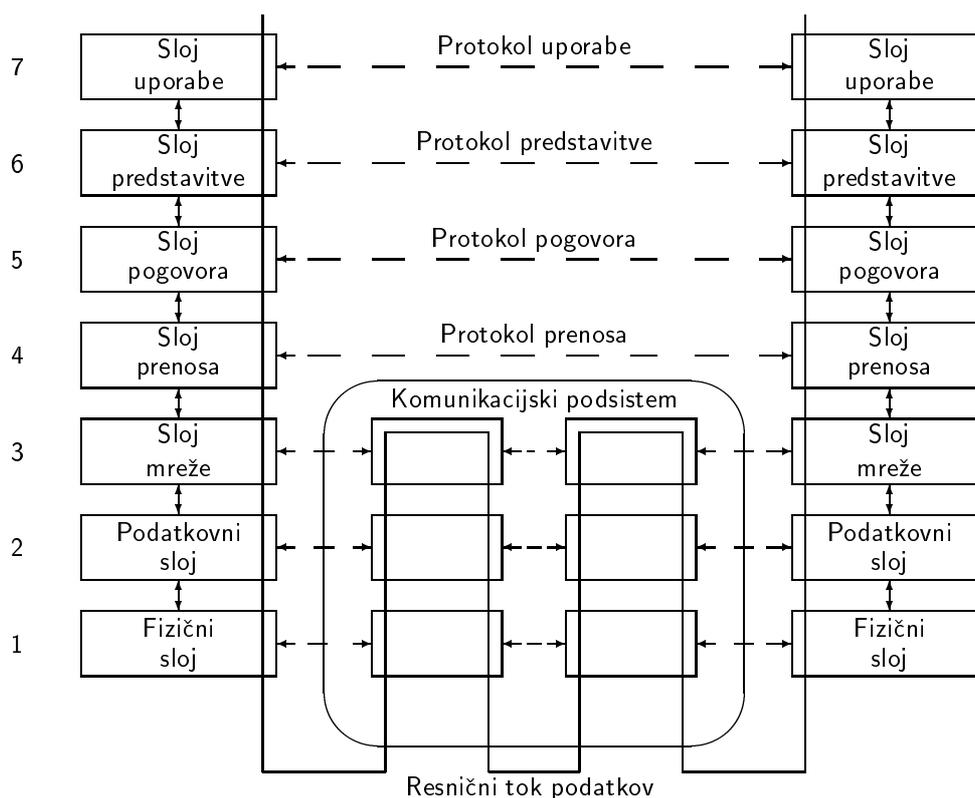
2.2 ISO OSI referenčni model

Da bi zagotovili združljivost sistemov različnih proizvajalcev ter omejili število različnih izvedb komunikacijskih postopkov in opreme, je mednarodna organizacija za standardizacijo ISO v začetku osemdesetih let predlagala model, po katerem naj bi gradili računalniška omrežja.¹⁰ Referenčni model OSI (ang. Open Systems Interconnection), kot pove tudi ime, se nanaša na odprte sisteme, to je sisteme, ki so odprti za komunikacijo z drugimi odprtimi sistemi. Model OSI obsega sedem slojev. Do sedmih slojev so prišli ob upoštevanju naslednjih načel[2]: sloj mora opravljati natančno določeno nalogo, nov sloj je potreben tam, kjer je potreben nov nivo abstrakcije, upoštevati je treba možnosti standardizacije, sloji morajo biti izbrani tako, da se čimbolj zmanjša pretok informacije med sloji (na

¹⁰ISO 7498-1984: Information Processing Systems - Open systems Interconnection - Basic Reference Model.

vmesnikih), število slojev naj se zadrži v razumnih mejah, da ne bi s prevelikim številom slojev po nepotrebnem večali obsežnosti, vendar naj bo slojev dovolj tako, da ne bi bili prisiljeni združiti dveh načelno različnih aktivnosti v isti sloj.

Referenčni model ISO OSI je narisan na sliki 16. Slika prikazuje odnos med dvema končnima vozliščema, ki ju povezuje komunikacijski podsistem. Model OSI sam po sebi ni arhitektura mreže, saj ne določa niti storitev niti protokolov posameznih slojev. Določa le funkcije posameznih slojev. Res je ISO izdelal ali privzel tudi standarde za posamezne sloje, vendar ti standardi niso sestavni del modela. Za posamezne sloje se lahko izbere poljubno kombinacijo protokolov. Iz tega sledi, da dve postaji, ki sicer komunicirata po referenčnem modelu OSI, a uporabljata različen protokol vsaj na enem sloju, med seboj ne moreta¹¹ komunicirati. Postaje lahko med seboj komunicirajo samo, če uporabljajo enak sklad protokolov ali enak protokolovni 'profil' (ang. Protocol Stack ali Protocol Suite ali Protocol Profil), enake protokole na vseh nivojih. V praksi se nekatere bolj uveljavljene protokolovne sklade imenuje na kratko kar protokol.¹² Čeprav to ni popolnoma dosledno, bi ne smelo biti razloga za dvoumnost.



Slika 16: Referenčni model ISO OSI.

¹¹Lahko s pomočjo protokolovnih prevajalnikov.

¹²Na primer TCP/IP protokol ali MAP protokol

Oglejmo si glavne naloge posameznih slojev, zaporedoma od najnižjega do najvišjega sloja.

Fizični sloj (ang. Physical Layer) skrbi za prenos informacijskih signalov po komunikacijskem kanalu. Osnovna informacijska enota tega sloja je bit. Sloj določa mehanske, električne in postopkovne lastnosti naprav in tokokrogov. Tipična vprašanja v zvezi s tem slojem so: napetostni nivoji signalov, hitrost prenosa, oblike signalov, vrste modulacij, konektorji in število priključkov na konektorjih, uporabljeni prenosni medij. Najbolj znani standardi in priporočila tega sloja so RS232, RS422, RS423, RS449, RS485, V.24, V.28, V.10, V.11, V.12, X.21, X.26, X.27, V.22, V.32, V.42, i.t.d.

Podatkovni sloj ali linijski sloj (ang. Data Link Layer) uporablja storitve fizičnega sloja (prenos signalov) in zagotavlja naslednjemu višjemu sloju (mrežnemu sloju) visoko zanesljivost prenosa podatkov v obe smeri med sosednjimi vozlišči. V ta namen deli daljša zaporedja bitov na manjše okvirje (Ang. Frames) ter skrbi za pravilen prenos posameznih okvirjev od vozlišča do vozlišča s preverjanjem pravilnosti prenešenega okvirja in s potrjevanjem pravilnosti sprejema. Tipična vprašanja v zvezi s tem slojem so: kako označiti začetek in konec okvirja, kako ugotavljati prisotnost napak na okvirjih, kako zahtevati ponoven prenos pokvarjenega okvirja in kako rešiti problem podvojenih okvirjev (dveh enakih okvirjev) v primeru, da se izgubi potrdilo že pravilno sprejetega okvirja. Tipični primeri protokolov (standardov), ki opravljajo funkcije tega sloja so: IBM-ova protokola BSC ali z drugim imenom BISYNC (Binary Synchronous Communication) in SDLC (Synchronous Data Link Control), ISO HDLC (High-Level Data Link Control), ki je povzet po SDLC. Tudi podatkovni protokol LAPB (Link Access Procedure - Balanced) v javnih podatkovnih omrežjih po priporočilu X.25 izhaja iz protokola HDLC. Za prenos podatkov med manjšimi računalniki sta zelo priljubljena Kermit in XMODEM ter njegove izpeljanke (YMODEM, ZMODEM, i.t.d.). Naslednja važna naloga tega sloja v lokalnih omrežjih je nadzor nad dostopom do skupnega kanala (Ang. Media Access Sublayer - MAC). Morda najbolj poznani standardi s tem v zvezi so Ethernet II, IEEE 802.3, IEEE 802.4, IEEE 802.5 ter podatkovni protokol LLC (Logical Link Control) po standardu IEEE 802.2.

Mrežni sloj (ang. Network Layer) skrbi za delovanje komunikacijskega podsistema. Na sliki 16 vidimo, da vmesna vozlišča, ki tvorijo komunikacijski podsistem, opravljajo naloge samo prvih treh (spodnjih) slojev, do vpljučno mrežnega sloja. Osnovna podatkovna enota tega sloja je paket (Ang. Packet). Ključna naloga tega sloja je zagotavljanje *poti prenosa* od oddajnega do sprejemnega vozlišča. Pot od oddajnika do sprejemnika je lahko določena čisto statično (vnaprej z izvedbo omrežja in se nikoli ne spremeni), lahko pa je drugačna za vsako novo zvezo, za vsako novo sporočilo ali celo za vsak posamezen paket (dinamično). Mrežni sloj torej skrbi za usmerjanje paketov od vozlišča do vo-

zlišča (ang. Routing), naslavljanje vozlišč, pa tudi za to, da ne pride do preobremenjenosti dela omrežja (ang. Congestion Control), opravlja razne (statistične) izračune, ki omogoča optimizacijo omrežja in/ali obračunavanje storitev. Med najbolj znanimi protokoli mrežnega sloja sta IP (Internet Protocol) v omrežju omrežij Internet, in PLP (Packet Layer Protocol) po priporočilu X.25 organizacije CCITT. V lokalnih omrežjih tega sloja ni ali pa je zelo 'tanek'. V omrežjih z Novellovim mrežnim operacijskim sistemom NetWare opravlja (približno) funkcije mrežnega sloja protokol IPX (Internetwork Packet Exchange) ki se vzgleduje po Xerox XDN ISD protokolu.

Prenosni sloj (Ang. Transport Layer) streže sosednjemu višjemu sloju (sloju pogovora). Od njega prevzema podatke, jih po potrebi deli na manjše enote - podatkovne enote prenosnega sloja (Ang. Transport Protokol Data Unit - T-PDU),¹³ zatem pa predaja naslednjemu nižjemu (mrežnemu) sloju ter skrbi, da podatki pogovornega nivoja pridejo pravilno na drugo stran. Torej skrbi za *prenos* informacije na drugo stran, kot pove tudi njegovo ime. Prenosni sloj je prvi 'pravi' sloj 'od-konca-do-konca' (ang. End-to-End). Z drugimi besedami, proces na eni končni postaji komunicira neposredno s sorodnim procesom na drugi končni postaji, brez posrednikov. Za primer protokola tega sloja navedimo protokol v omrežjih Internet TCP (Transmission Control Protocol), ki se pojavlja skupaj s protokolom IP, in ISO 8073, ki se uporablja v omrežjih MAP (Manufacturing Automation Protocol) ter TOP (Technical and Office Protocol).

Pogovorni sloj (Ang. Session Layer) omogoča 'pogovor' med uporabniškimi procesi na različnih postajah in s tem skrbi za organizacijo in strukturiranje dialoga. Podobno kot prenosni sloj omogoča prenos podatkov, poleg tega pa nudi tudi nekatere kvalitetnejše storitve specifične za določene aplikacije. Če bi za prenosni sloj rekli, da skrbi za prenos informacije med enim in drugim končnim vozliščem, skrbi pogovorni sloj za pogovor med istorodnima procesoma enega in drugega vozlišča. Večina omrežij tega sloja sploh nima. Na tem sloju je običajno (a ni nujno) realiziran sistem daljinskih klicev (Ang. Remote Procedure Call), kot eden izmed možnih načinov komunikacije med krajevno porazdeljenimi procesi.

Predstavitveni sloj (Ang. Presentation Layer) opravlja pogoste storitve, ki zahtevajo splošno rešitev za večino uporabnikov teh storitev. Tipične storitve tega sloja so: kodiranje in prekodiranje (npr. ASCII V EBCDIC), šifriranje, zgoščevanje podatkov in podobno.

Sloj uporabe (ang. Application Layer) predstavlja vmesnik med končnim uporabnikom in komunikacijskim sistemom. Vsebuje morda največ storitev in protokolov. Najbolj znane med njimi so: prenos datotek (File Transfer), navidezni terminal (Virtual Terminal), elektronska pošta (E-mail) X.400, mrežne direktorijske storitve X.500, i.t.d.

¹³Od tega sloja navzgor za PDU nimamo posebnih imen kot paket in okvir.

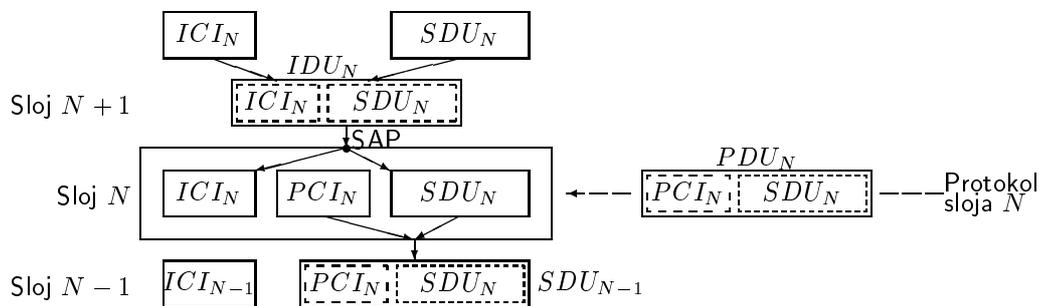
2.3 OSI, storitve, protokoli in načelo ovojnice

Temeljna naloga vsakega sloja v modelu OSI je, da daje storitve sosednjemu višjemu sloju. Aktivne elemente za izvedbo storitev imenujemo *aktivnosti* (ang. Entities). Na primer, aktivnosti prenosnega sloja imenujemo aktivnosti prenosnega sloja. Aktivnosti istega sloja na različnih postajah imenujemo istorodne aktivnosti (ang. Peer Entities). Istorodne aktivnosti sloja N skrbijo za izvedbo storitev, ki jih sloj N daje sloju $N + 1$. Pri tem se poslužujejo protokola sloja N . Za realizacijo svojih storitev se sme sloj N poslužiti storitev, ki jih njemu nudi nižji sloj $N - 1$.

Storitve sloja so dostopne na dostopni (ali pristopni) točki storitev (ang. Service Access Point), glej sliko 17. Dostopna točka storitve sloja N je mesto, kjer so njegove storitve dane sloju $N + 1$. Pri izmenjavi informacije na vmesniku morata sosednja sloja upoštevati pravila vmesnika. Ta pravila v bistvu definirajo vmesnik. Na primer, na tipičnem vmesniku preda aktivnost sloja $N + 1$ podatkovno enoto vmesnika IDU (ang. Interface Data Unit) aktivnosti sloja N . Podatkovno enoto vmesnika sestavljata dva dela: podatkovna enota storitve SDU (ang. Service Data Unit) in nadzorna informacija vmesnika ICI (ang. Interface Control Information). Nadzorno informacijo vmesnika ICI potrebuje aktivnost sloja N za sebe in ni del koristnih podatkov. Iz nje razbere, kaj mora narediti za koristnika storitve. Podatkovna enota storitve SDU pa je tista informacija, ki se mora prenesti po omrežju na drugo stran k istorodni aktivnosti sloja $N + 1$. Pri prenosu podatkovne enote storitve SDU na drugo stran, se istorodni aktivnosti sloja N obeh strani sporazumevata po protokolu sloja N . Podatkovne enote, ki si jih izmenjujeta preko omrežja, imenujemo protokolne podatkovne enote PDU (ang. Protocol Data Unit). Na primer, protokolna podatkovna enota sloja 2 je okvir, protokolna podatkovna enota tretjega sloja je paket in podobno. Protokolno podatkovno enoto sestavlja koristni del (SDU) in protokolna nadzorna informacija PCI ali 'glava' (ang. Protocol Control Information ali tudi Header), ki jo istorodni aktivnosti rabita za medsebojno sporazumevanje po protokolu sloja N .

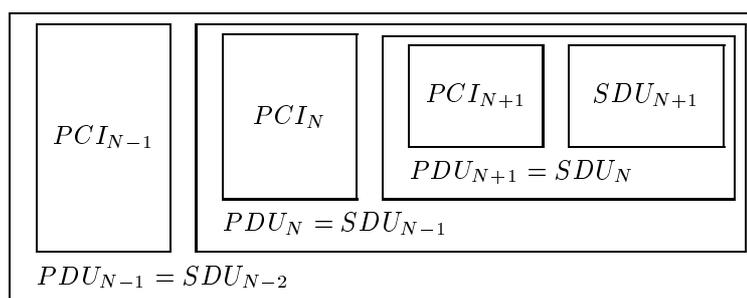
Za realizacijo storitve, ki jo sloj N daje sloju $N + 1$ se sloj N poslužuje storitev sloja $N - 1$ pod njim. Torej, opisane razmere, ki vladajo na vmesniku slojev $N + 1$ in N se v podobni obliki ponovijo na vmesniku slojev N in $N - 1$. Protokolni podatkovni enoti sloja N se doda nadzorna informacija vmesnika (ICI) in tako tvori podatkovno enoto vmesnika (IDU). Torej je protokolna podatkovna enota sloja N s stališča dajalca storitve, ki je v tem primeru sloj $N - 1$, podatkovna enota storitve (SDU). V splošnem velja, da je $PDU_N = SDU_{N-1}$. Tako smo spoznali *načelo ovojnice*, glej sliko 18.

Istorodne aktivnosti sloja $N + 1$ si izmenjujejo protokolne podatkovne enote PDU_{N+1} . Pri tem koristijo storitve sloja N . Aktivnosti na sloju N pa obrav-

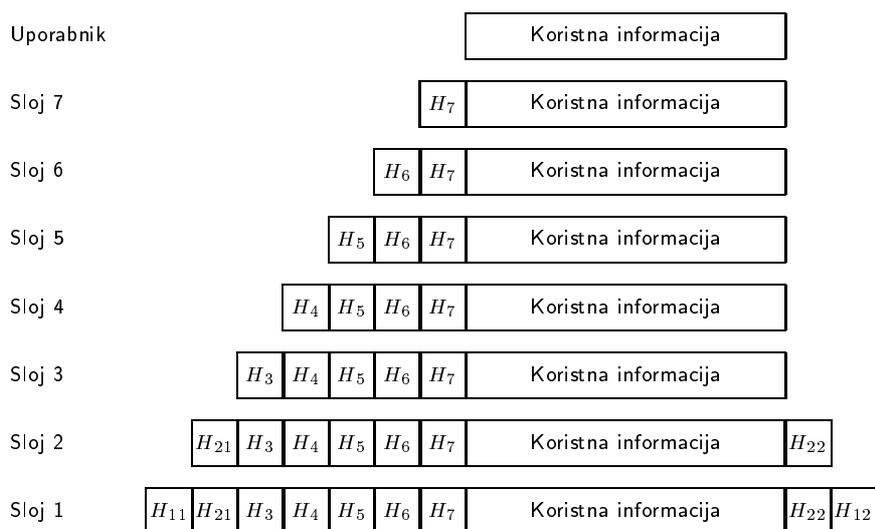


- | | | |
|-------------------------------------|---|---------------------------------|
| SAP (Service Access Point) | = | Dostopna točka storitve |
| IDU (Interface Data Unit) | = | Podatkovna enota vmesnika |
| ICI (Interface Control Information) | = | Nadzorna informacija vmesnika |
| SDU (Service Data Unit) | = | Podatkovna enota storitve |
| PCI (Protocol Control Information) | = | Protokolna nadzorna informacija |
| PDU (Protocol Data Unit) | = | Protokolna podatkovna enota |

Slika 17: Podatkovne strukture in razmere na vmesnikih med sloji $N + 1$, N in $N - 1$.



Slika 18: Načelo ovojnice.



Slika 19: Načelo ovojnice. Koristni informaciji uporabnika se na vsakem sloju doda kontrolna informacija protokola (glava H).

navajo PDU_N zgolj kot podatke SDU_N , ki jih morajo prenesti na drugo stran, njihova vsebina pa jih ne zanima. Podatkom PDU_{N+1} pred pošiljanjem dodajo spremno informacijo (podatke 'zapre v ovojnico'), kot določa protokol na sloju N . Tako nastane protokolna podatkovna enota PDU_N , ki jo sloj N preda nižjemu sloju $N - 1$, ta pa spet nižjemu, in tako dalje.

Slika 19 shematično prikazuje načelo ovojnice v sedemslojnem modelu OSI. Na vsakem sloju se koristni informaciji doda kontrolna informacija protokola tega sloja (glava) ter na koncu vse skupaj odda v kanal. Na sprejemni strani teče postopek v obratni smeri.

Pojma storitve ne smemo zamenjati s pojmom protokol. *Storitev* je določena z množico osnovnih operacij, ki jih sloj nudi višjemu sloju. *Storitev* določa, kaj (kakšne operacije) je nek sloj pripravljen opravljati za uporabnika storitev. Pri tem ni važno, kako so realizirane te operacije. *Storitev* se nanaša na vmesnik med sosednjima slojema (na navpično komunikacijo). Nasprotno temu pa se protokol nanaša na vodoravno komunikacijo med istorodnima procesoma. *Protokol* je množica pravil, ki jih morata upoštevati procesa (aktivnosti) znotraj nekega sloja, pa tudi določila o obliki in pomenu okvirjev, paketov ali sporočil. Aktivnosti znotraj sloja uporabljajo protokol za realizacijo storitev. S stališča uporabnika storitev je povsem nepomembno kakšen protokol je izbran za realizacijo storitev. Uporabniku storitev mora biti to popolnoma prikrito. Protokol se lahko spremeni (zamenja), ne da bi se spremenile storitve. Na primer *prenos datotek* je storitev aplikacijskega nivoja. Za realizacijo prenosa datotek je potreben *protokol za prenos datotek*. Prenos datotek je lahko realiziran na ta ali oni način,

s takim ali drugačnim protokolom. Za uporabnika storitev (prenosa datotek) je važno samo to, da je datoteka verno prenešena na drugo stran in prav nič, *kako* so realizirane storitve (kakšen je protokol za prenos datotek).

2.3.1 Osnovne operacije s storitvami

Storitev je definirana z množico osnovnih operacij (Ang. Primitives), ki so na voljo uporabniku storitve (ali kakšni drugi aktivnosti). S temi osnovnimi operacijami uporabnik storitve določi, kaj naj dajalec storitve naredi zanj. V ISO OSI referenčnem modelu obstajajo štiri vrste osnovnih operacij: zahteva (Request), naznanilo (Indication), odziv (Response) in potrdilo (Confirm):

Zahteva	aktivnost zahteva od dajalca storitve neko dejanje
Naznanilo	aktivnost je obveščena o nekem dejanju
Odziv	aktivnost se odzove na neko dejanje
Potrdilo	aktivnost je obveščena o realizaciji svoje zahteve

Zahteva je operacija, s katero neka aktivnost zahteva (kot pove ime) od dajalca storitve neko dejanje, na primer vzpostavitev zveze. Ko je zveza vzpostavljena, je istorodna aktivnost o tem obveščena z *naznanilom*. Aktivnost, ki dobi naznanilo se odzove, uporabi operacijo *odziv* ter tako pove, da v našem primeru pristaja na vzpostavitev zveze. Končno, aktivnost, ki je opravila zahtevo dobi obvestilo o vzpostavitvi zveze z operacijo *potrdilo*. Vse štiri operacije so prikazane na sliki 20.

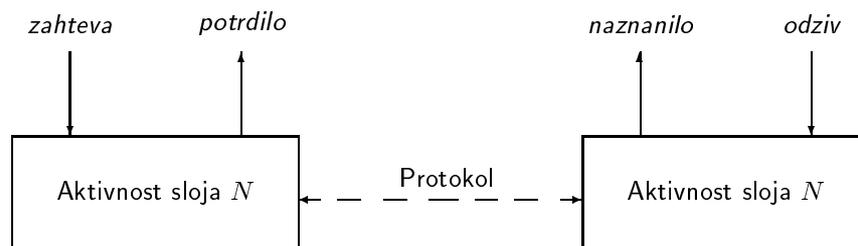
Storitve so lahko *potrjene* ali *nepotrjene*. Potrjena storitev omogoča vse naštetih štiri osnovne operacije. Za nepotrjeno storitev obstajata samo *zahteva* in *naznanilo*. Sedaj pa pogledjmo primer, ki nam je najbližji. Denimo, da želi oseba A po telefonu govoriti z osebo B. Možno zaporedje operacij je:

Storitev	Vzpostavitev zveze
Zahteva	Oseba A pozove osebo B (Zahteva vzpostavitev zveze),
Naznanilo	Oseba B sliši znak poziva (Naznanilo zahteve za vzpostavitev zveze),
Odziv	Oseba B dvigne slušalko (Pristanek na zahtevo za vzpostavitev zveze),
Potrdilo	Oseba A zazna prestanek pozivanja (Zveza je vzpostavljena),

Sledi pogovor

Storitev	Sprostitev zveze
Zahteva	Oseba A odloži slušalko (Zahteva sprostitve zveze)
Naznanilo	Oseba B sliši zahtevo za prekinitev zveze in odloži slušalko.

V tem primeru je vzpostavljanje zveze potrjena storitev, sprostitve zveze pa nepotrjena storitev.



Slika 20: Osnovne operacije za storitev *vzpostavi zvezo*.

Toliko o storitvah in operacijah s storitvami.

2.4 Primeri mrežnih arhitektur

Čeprav je arhitekturni model OSI sedem slojni pa to ne pomeni, da arhitekture z drugačnim številom slojev niso možne ali ne bi smele obstajati. Drugačne mrežne arhitekture so obstajale dosti prej, predno je nastal model OSI, na primer IBM-ova mrežna arhitektura SNA (System Network Architecture), Digital-ova arhitektura DNA, in druge. Najbolj prepričljiv dokaz za smisel ne-OSI arhitektur je globalno omrežje Internet, ki se dnevno potrjuje z vse večjim številom vozlišč. Ne glede na to je model OSI gotovo prispeval k bolj sistematičnemu in bolj enotnemu obravnavanju zgradbe in delovanja omrežij. Denimo, marsikatero omrežje, ki je nastalo pred samim modelom in se obravnavalo kot omrežje “v enem kosu”, se danes obravnava razslojeno z vidika modela OSI.

2.4.1 Arhitektura omrežja Internet

Z imenom Internet (z veliko začetnico) označujemo globalno svetovno omrežje, ki povezuje omrežja tipa TCP/IP. Ko govorimo o omrežjih TCP/IP, mislimo na omrežja, ki uporabljajo protokolovni sklad TCP/IP (Transmission Control Protocol/Internet Protocol) in pa tudi druge protokole, ki se bistveni za delovanje teh omrežij (npr. SNMP, UDP, i.t.d.). Omrežje Internet temelji na štirislojnim arhitekturnem modelu, ki ima svoje korenine v omrežju ARPANET iz konca šestdesetih let. Internet ni grajen po modelu OSI. To nikakor ne pomeni, da bi Internet ne bilo “odprto” omrežje ali da ne bi imelo možnosti realizacije vseh potrebnih omrežnih funkcij. Internet je omrežje, ki “deluje”. To potrjuje dejstvo, da je TCP/IP omrežij danes daleč več kot vseh drugih omrežij skupaj. Slika 21 prikazuje arhitekturni model in primer arhitekture. Spodnji trije sloji so mrežnega značaja, zgornji sloj je uporabniško usmerjen. V arhitekturi na levi strani slike so zajeti le nakateri najbolj tipični protokoli izmed sicer možnih protokolov. V

lokalnih omrežjih Internet se v spodnjem sloju (ang. Network Interface Layer) največ uporablja Ethernet protokol. Za povezovanje oddaljenih vozlišč točka-točka po podemskih linijah se uporabljata protokola SLIP (Serial Link IP) in PPP (Point-to-Point-Protocol). Ker spodnji sloj realizira funkcije spodnjih dveh slojev modela OSI, pravimo tudi, da je arhitektura petslojna. Če primerjamo Internet model z modelom OSI, bi medomrežni sloj (ang. internet layer) najbolj ustrezal mrežnemu sloju, prenosni sloj prenosnemu, aplikacijski sloj pa zgornjim trem slojem modela OSI.



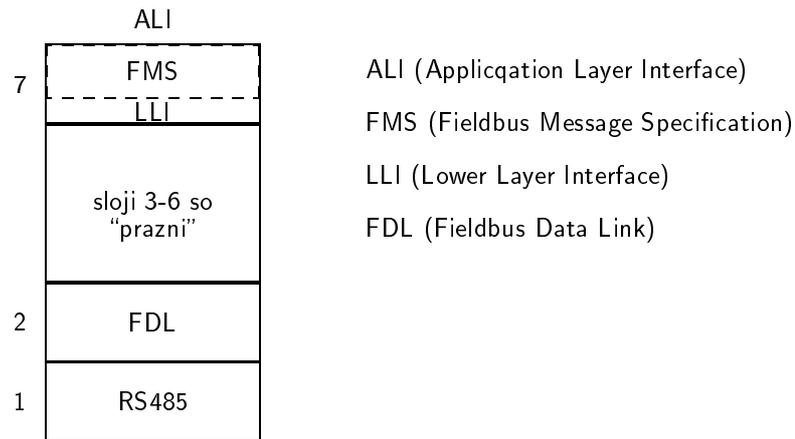
Slika 21: Model in arhitektura omrežja Internet.

2.4.2 Arhitektura Profibus

Procesno področno vodilo Profibus (ang. Process Field Bus) so razvili nemški proizvajalci industrijske komunikacijske opreme v sodelovanju s še petimi nemškimi inštituti. Namenjen je za povezovanje naprav v industrijskih okoljih (senzorjev in aktuatorjev, programljivih krmilnikov - PLC, numerično krmiljenih strojev - NC, CNC, industrijskih računalnikov - PC, i.t.d.). V arhitekturni zasnovi so razvijalci omrežja Profibus upoštevali model OSI, očitno pa so se zgledovali po arhitekturi MiniMAP. Da bi zadostili ostrim zahtevam po kratkih odzivnih časih, a hkrati ohranili možnost interoperabilnosti (zmožnosti skupnega delovanja naprav različnih proizvajalcev), so definirali samo tri sloje: fizični in podatkovni sloj ter sloj aplikacije. Funkcionalnost ostalih slojev so vgradili v podsloj aplikacije LLI (Lower Layer Interface). Obstajajo tri različice omrežja Profibus: Profibus-FMS, Profibus-PA in Profibus-DP. Slika 22 prikazuje arhitekturo omrežja Profibus-FMS. O procesnih področnih vodilih bomo še govorili.

Literatura

- [1] J. Day, H. Zimmermann, "The OSI Reference Model", *Proc. of IEEE*, 71, pp.1334-1340, Dec 1983.



Slika 22: Arhitektura omrežja Profibus-FMS.

- [2] V. C. Jones, *MAP/TOP Networking*, McGraw-Hill 1988.
- [3] J. Morris, *Introduction to Communication, Command, and Control Systems*, Pergamon Press, 1978.
- [4] D. M. Piscatello, A. L. Chapin, *Open Systems Networking, TCP/IP and OSI*, Addison-Wesley, 1993.
- [5] E. Parr, *Programable Controllers*, B.H. 1993.
- [6] T. Ramtheke, *Networks*, Prentice-Hall, 1994.
- [7] W. R. Stevens, *TCP/IP Illustrated, Vol. 1*, Addison-Wesley, 1994.
- [8] A. Tanenbaum, *Computer Networks*, 3rd Edition, Prentice-Hall 1996.
- [9] J. Virant, *Teleinformatika in ISDN*, Univerza v Ljubljani, Fakulteta za elektrotehniko in računalništvo, Ljubljana 1989.
- [10] Michael Volz, *PROFIBUS*, PROFIBUS Nutzerorganisation, Technische Druckschrift, 1994.
- [11] *Digital Industrial Networks Guidebook*, Digital, 1988.
- [12] ISO/IEC 7498-1:1994 Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, 1994. (Ekvivalenten priporočilu ITU-T X.200)
- [13] *Mikroračunalniški telekomandni sistem TI30/11*, ISKRA Avtomatika, Ljubljana 1980.
- [14] *Open Systems Interconnection, Technology Overview*, Novell 1991.