

## OPERACIJSKI SISTEMI

**DEFINICIJA:** Osnovni sistemski program, ki nadzira vsa računalnikova sredstva (leži med strojno opremo in uporabniškimi programi)

**NALOGE** operacijskega sistema:

- delo s pomnilnikom – kar se izvaja, se izvaja v pom. prostoru, kjer CPE opravlja naloge.  
Dodeljevanje pomnilnika
- nadzor V/I enot
- upravljanje s sredstvi računalnika in možnost upravljanja programske opreme
- skrb za uporabniški vmesnik
  - Tekstovni
  - Grafični (Graphic User Interface)
- podpira uporabniške programe
- delo sistema z datotekami
  - Datotečni sistem

## UPORABNIŠKI VMESNIK-UI

C:\del\*.\*-tekstovni vmesnik

Grafični vmesnik

## NAVIDEZNI RAČUNALNIK

**UPORABNIŠKI NIVO** – praktična uporaba računalnika (več različnih programov, uporabnik pozna le program s katerim dela)

**PROBLEMSKI NIVO** – izvajamo s programskimi jeziki, imamo dostop do strojne opreme (SOFTWARE)

**NIVO ZBIRNEGA JEZIKA** – simbolni ukazi, imena podatkov, posplošitev, večji koraki, glede na strojni jezik (MOTOROLA)

**NIVO OPERACIJSKEGA SISTEMA** – kodirani klici in ukazi operacijskega sistema

**NIVO STROJNEGA JEZIKA** – delno tolmači OS – majhni koraki

**NIVO MIKROPROGRAMA** – izvajanje OS s pomočjo mikroprograma (skupek kode, ki izvaja določeno funkcijo, razume ga mikroprocesor)

**NIVO DIGITALNE LOGIKE** – tolmačenje operacijskega sistema in strojne opreme (010010101010100100100)

## OPERACIJSKI SISTEMI

## 1.1 UVOD

Rutine operacijskega sistema predstavljajo vmesnik med uporabniškim nivojem in računalniškimi viri (computer resources), kot so pomnilnik posamezne strojne enote računalniškega sistema, potem podatkovne zbirke na zunanjih podatkovnih nosilcih, sistemski programi splošnega namena in uporabnosti.

## 1.2 DELITEV OS

- ENOUPORABNIŠKI – DOS, Win
- VEČUPORABNIŠKI – Win 2000/Xp server, Unix
- ENOOPRAVILNI – preemptivni
- VEČOPRAVILNI – multitasking – kooperativni

Čas omejen, OS določa uporabo določenega števila ciklov za en program.

Nekateri procesi vezani na ukaze (print, copy) se izvajajo enakovredno – paralelno

Po povezljivosti – omrežna povezava – vsi OS



GONILNIKI-povezujejo jedro OS in krmilnike – direktno dostopajo do strojne opreme

LOGIČNI NIVO nam omogoča dostop do rutin logičnega sloja, ki povezujejo fizični sloj OS oz. opravila z rač. viri.

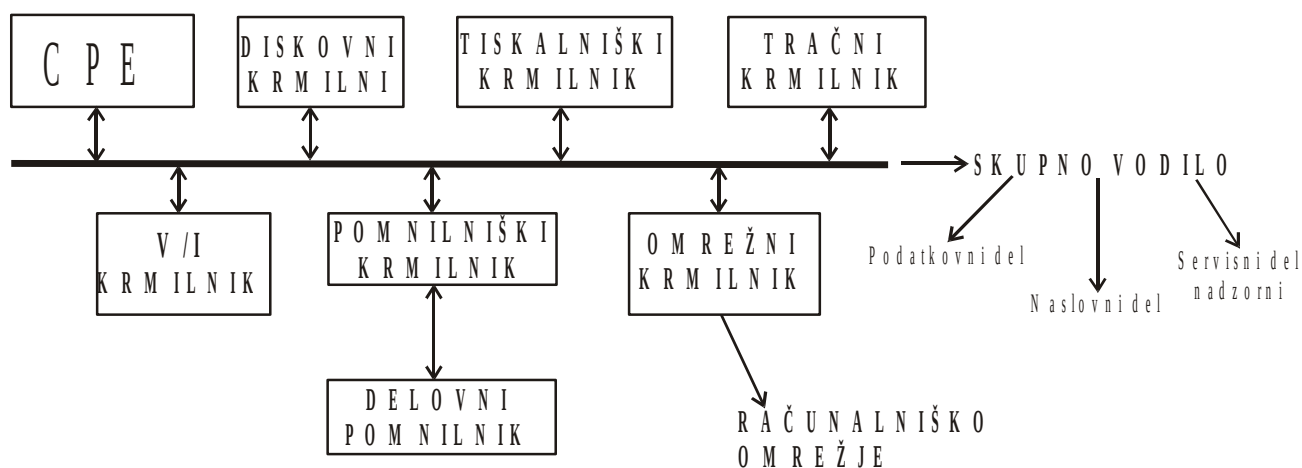
V FIZIČNEM SLOJU so podane lastnosti in značilnosti posameznih delov in opreme, ki jo neposredno uporabljamo.

LUPINA opravlja vlogo uporabniškega vmesnika, ponuja uslužnostne rutine in pomaga pri delu z OS.

VMESNIK MED UPORABNIKOM IN RAČUNALNIŠKIM SISTEMOM deluje s pomočjo ukaznega jezika (delo iz ukazne vrstice) ali prek okenskega izbira (delo v grafičnem načinu).

USLUŽNOSTNE RUTINE opravljajo vodenje realnega časa, upravljanje s pomnilnikom, izvajanje matematičnih rutin, dostop do podatkov.

UPORABNIŠKI PROGRAMI preko knjižnic, objektnih ali izvedljivih modulov izkoriščajo usluge OS.



Računalniški sistemi večinoma temeljijo na tehnologiji skupnega systemskega vodila – to razdeljeno na tri dele: nasl., pod., servisni. Vodila so sestavljena iz določenega števila linij (vodnikov), ki prenašajo dvojiške informacije (bite) v obliki el. impulzov. Računalniški sistem vsekakor ima CPE, ki sodeluje z delovnim pomnilnikom, v katerem so shranjeni podatki in programi, ki se trenutno izvajajo. Na skupnem vodilu so priključeni še zun. pom., diski, disketniki, tračne in kasetne enote, CD-ROM ter V/I enote: terminali, tiskalniki, itd.

Z dodajanjem mrežnega krmilnika na skupno vodilo je omogočena skupna povezava v omrežju. CPE razume svoj strojni jezik, ki ga sestavlja nabor ukazov, vgrajenih v procesorjevo logično strukturo. Ukazi se shranjujejo v delovnem pomnilniku, izvajajo pa v dostavno izvršnem ciklu. Procesor v dostavnem ciklu izda naslov naslednjega ukaza, ga prebere iz pomnilnika in razpozna. V izvršilnem delu cikla sledi izdajanje naslovov za operande, njihovo včitavanje in uporaba pri izvedeni operaciji.

Procesor lahko izda navzven  $2^n$  naslovov in s tem je določen naslovni prostor. Glede na širino podatkovnih vodil govorimo o 16, 32, 64-bitnih procesorjih. Nadzorno vodilo združuje signale, ki omogočajo usklajeno delovanje vseh računalniških enot. Pri delu CPE z periferijo je možno izvajanje občasno prekiniti z t.i. prekinitvami (interrupts).

CPE je celota, sestavljena iz logičnih vezij, ki obravnava posamezne ukaze. Ukaz - opravilo na logičnem nivoju.

## PREKINITVE

- STROJNE prekinitve (na vodilu med kontrolerji)
- PROGRAMSKE prekinitve (rutineki jih uporabljamo v napisanem programu za kontrolo periferije)

Strojne prekinitve so električni signali, ki prek posebnih vhodni linij vplivajo na procesor.

Programske prekinitve so posebni strojni ukazi, ki do procesorja zahtevajo preusmeritev toka izvajanja.

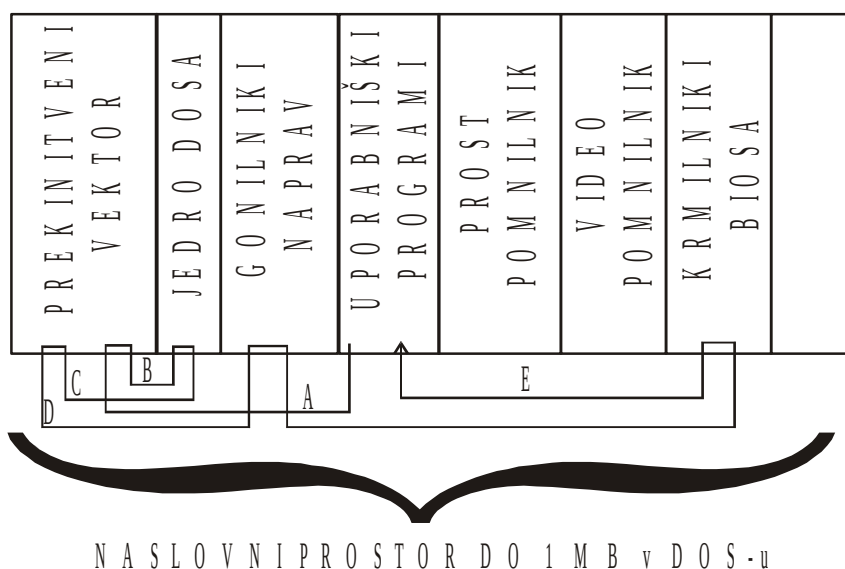
## DOGAJANJE OB KATERI KOLI PREKINITVI :

Procesor prekine izvajanje trenutnega opravila, v skladu shrani naslov kjer prekine izvajanje ter nadaljuje na naslovu, ki je vezan na sproženo prekinitvev. Za vsako prekinitvev je predviden kazalec v

katerega je vpisan nadaljevalni naslov za izvajanje, ki sledi prekinitvi. Ti kazalci so prekinitveni vektorji, rutine ki stečejo po prekinitvah so prekinitvene rutine.

Prekinitve so določene po prioritetah. Nivo izvajanja prekinitve je zabeležen v statusni besedi procesorja ali programa. Procesor lahko onemogoči oziroma omogoči izbrane prekinitve s pomočjo maskiranja (postavitev nadzornih bitov, ki to omogočajo). Prekinitve so asinhrono – neodvisne od programa, ki se izvaja. Nad aparaturno opremo srbi posebna enota, prekinitveno krmilnik. Tak krmilnik omogoča programsko nastavitvev prioritet za posamezne prekinitve (16 – strojnih prekinitvenih prekinitvev pri IBM), če se zaporedoma pojavi več prekinitvev se tiste postavijo v vrsto.

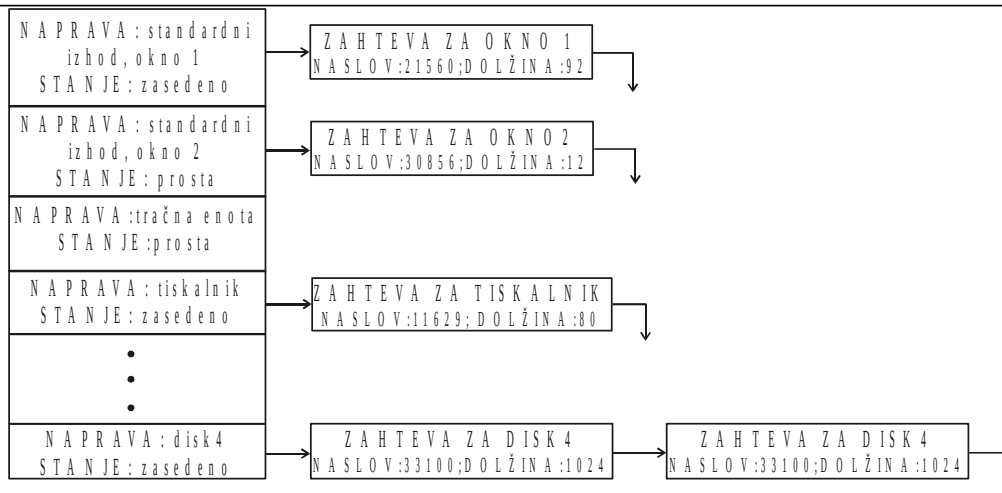
Programske prekinitve so posebej izvedeni programski klici iz posameznega programa, ki kličejo eno od splošni namenskih rutin. Tako nadzor poteka indirektno preko prekinitvenih vektorjev, kjer lahko kadarkoli vpišemo trenutno veljavne naslove prekinitvenih rutin.



Na sliki je prikazan tok prenosa podatkov med UP (uporabniški program) in OS. Začne se z ukazom za programsko int. znotraj UP (želja po branju z diska). V ustrezne registre CPE se vpišejo informacije za branje prek določenega kanala v pomnilniški vmesnik. Prek int. vektorja se prenese nadzor v jedro DOS-a (A). Tam se podatki iz registrov CPE dešifrirajo in ker je to branje z diska, je njegov naslov podan v int. vektorju za dostop do diska (C). Gonilniki, ki povezujejo OS in strojno opremo, pokličejo diskovni programski krmilnik in poskrbi da se izvede zahtevano branje z diska, oz. da se prebrani podatki prenesejo v pomnilniški vmesnik (D). Razen teh prekinitvev imamo še reševanje izjemnih stanj (razne napake). Pri izjemnih stanjih se najprej pojavi določena strojna prekinitvev, potem pa rutina za obravnavo napak znotraj jedra OS.

HAL (Hardware Abstraction Layer)-strojna oprema prikazana kot gonilniki, s katerimi potem komuniciramo (vgrajene zaščite).

**DELO Z V/I NAPRAVAMI**

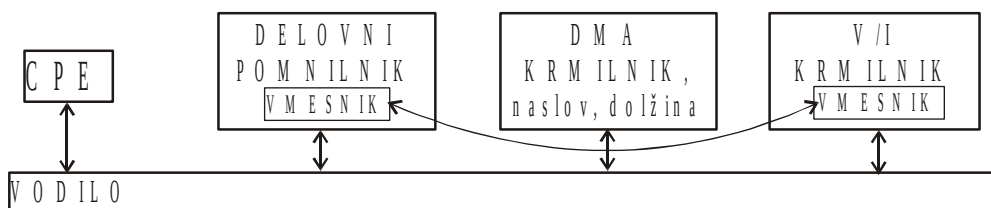


Posegi na perifernih napravah so običajno izvedeni, da trenutno izvajani program počaka (stoji) na konec V/I operacije (sinhroni V/I poseg), lahko pa se določeni program izvaja vzporedno z izvajanjem V/I operacije (asinhroni V/I poseg). Uporabniški progami kličejo OS, ki posreduje pri V/I operacijah. Primer takega systemskega klica je Wait (čakaj). Za V/I naprave si OS vodi statusno labelo, kjer je zabeležen trenutni status posamezne naprave (glej sliko).

Neposredni stik s periferijo ustvarjajo krmilniki naprav, katere opravlja OS prek gonilnikov. Pri prenosu podatkov v V/I operacijah je soudeležen CPE. Ker je to potratno, posamezne krmilne naprave med periferijo in pomnilnikom brez sodelovanja CPE po principu neposrednega dostopa do pomnilnika DMA (Direct Memory Access).

## DMA (Direct Memory Access)

Krmilniki DMA sodelujejo s krmilniki periferije in nadzirajo tok podatkov neposredno med V/I napravami in pomnilnikom. Pri tem uporabljamo sistemsko vodilo, ki pri posegih DMA ni na razpolago CPE. Zato se DMA in CPE dogovarjata, kdo sme uporabljati vodilo.

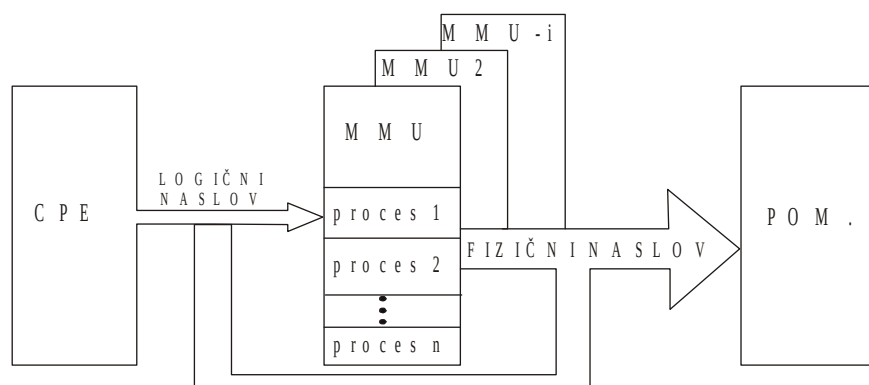


Krmilnik DMA vodi s pomočjo svojih notranjih reg. podatke o naslovih in št. Podatkov zlogov ali besed, katere se morajo iz periferije prenesti v pomnilnik in obratno. Krmilnik DMA lahko deluje v načinu prikrito (cycle stealing-kraja ciklov CPE), tako da odvzame CPE vodilo za prenos ene besede. Drugi način je v nizih (burst), kjer krmilnik DMA obdrži vodilo, dokler ne opravi celotnega prenosa med periferijo in pomnilnikom (CPE čaka na vodilo). Vzporedno s krmilnikom DMA CPU lahko dela, dokler ima vse potrebne ukaze in operande. V svoji notranji pretočni strukturi (pipeline-cevovod) ali v svojem predpomnilniku (cache).

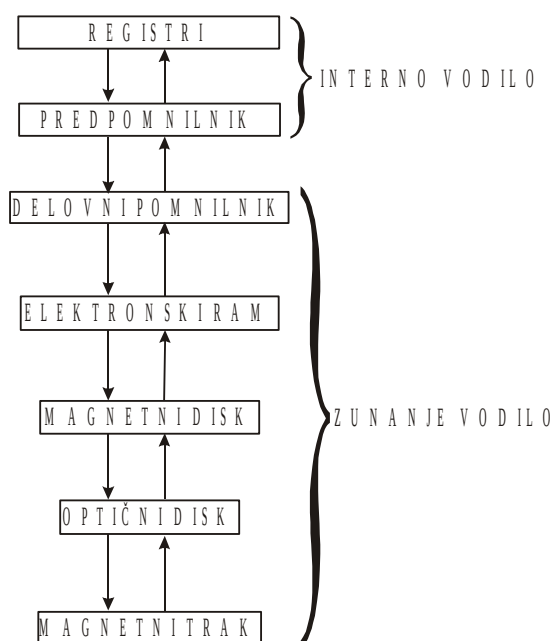
## ZASNOVA POMNILNIKA

Računalniški sistemi uporabljajo notranji ali delovni pomnilnik, kjer so shranjeni programski ukazi in nekateri podatki in je neposredno povezan s CPE (hiter pom.); zunanji ali sekundarni pomnilniki, ki služijo

za shranjevanje večjih količin podatkov (periferne enote, ki jih upravljajo strojni krmilniki – kontrolerji). Do pomnilnika dostopamo z naslavljanjem lokacij, kjer so shranjene besede (8-bitna beseda =>oktet=>zlog=>Byte). Velikost notranjega pomnilnika je danes med 64 Mb in 1GB. OS prek enot za upravljanje s pomnilnikom (Memory Managment Unit), ki so na strojnem nivoju vključeni med CPE in pomnilnik, zagotavlja zanesljivo delovanje več procesov, ki se izvajajo v zaščitenem (rezerviran) pomnilniškem prostoru.



## POMNILNIŠKA HIRARHIJA



Pomnilniška hierarhija v računalniškem sistemu začneja z najhitrejšimi (registri) in najbolj počasnimi (magnetni trak) pomnilniškimi enotami. Pomembno vlogo ima predpomnilnik, ki sodeluje v izmenjavi podatkov med počasnejšim delovnim pomnilnikom in procesorjem. Algoritmi iz operacijskega sistema skrbijo z optimalno izkoriščanje pomnilniške hierarhije, kot tudi za preprečevanje razdrobljenosti (fragmentacije) pomnilnika. V počasne pomnilniške enote trajno shranjujemo večje količine podatkov.

## ZAŠČITA PODATKOV in ZANESLJIVOST DELOVANJA SISTEMA

V računalniškem sistemu je pomembna zaščita podatkov pred uničenjem, izgubo ali zlorabo in zanesljivostjo delovanja sistema oziroma aplikacij. Ta zaščita je omogočena na nivoju strojne in programske opreme.

Razlikujemo več postopkov ki se uporabljajo:

- Delovanje procesorja v dveh načinih (uporabniški, nadzorniški)
- zaščita V/i operacij
- zaščita delovnega pomnilnika
- zaščita procesorja

Procesor deluje v nadzorniškem načinu kadar izvaja rutine iz operacijskega sistema ali drugih delov sistemske, programske opreme, pri tem uporablja posebne privilegirane ukaze s katerimi nadzoruje delovanje celotnega sistema in posameznih aplikacij. V uporabniškem načinu je predvideno izvajanje aplikacij in so tukaj večje omejitve kot v nadzorniškem stanju.

Zaščita V/i operacij je pod nadzorom operacijskega sistema. Zato so V/i posegi definirani kot privilegirane operacije, ki se smejo izvajati, le v nadzorniškem načinu.

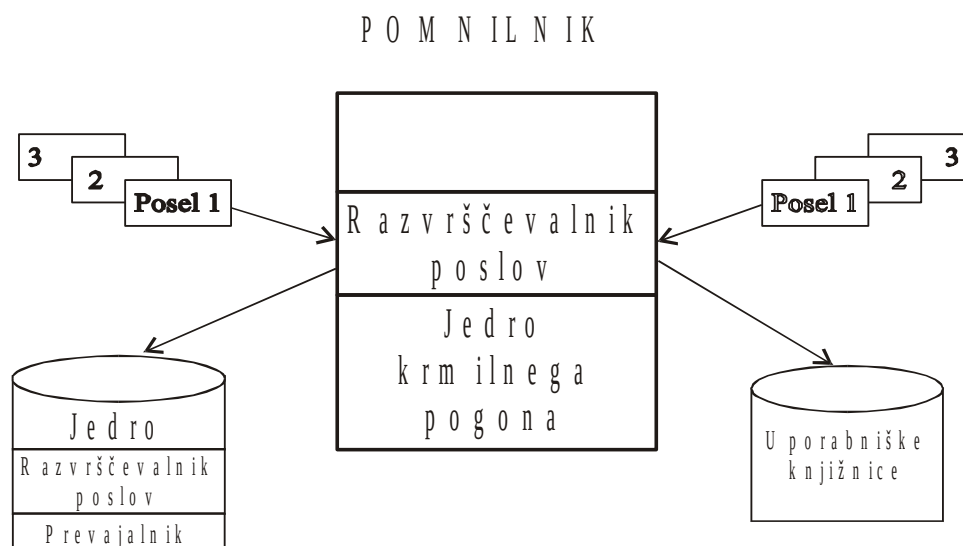
Zaščita delovnega pomnilnika je podana z dodajanjem parnostnih bitov (na nivoju pomnilnika). Sam operacijski sistema skrbi da vsaka aplikacija uporablja svoj pomnilniški prostor in ne posega čez meje tega dela pomnilnika.

Zaščita procesorja je narejena na strojnem in programskem nivoju kjer se v primeru nedovoljenih zastojev aktivirajo prekinjivne strojne oziroma programske rutine, ki poskušajo sistem spraviti v nadaljnje delovanje.

## ZAPOREDNI PAKETNI SISTEMI

Znajo delovati le enoprogramsko, vsi posli se razvrstijo v vhodno vrsto, od koder jih jemlje OS. Le-ta interno opravlja naslednje naloge.

- Nudi povezavo z V/I gonilniki
- Obvladuje prekinitve, razrešuje napake, omogoča uporabo logičnih imen in naslovov za podatke



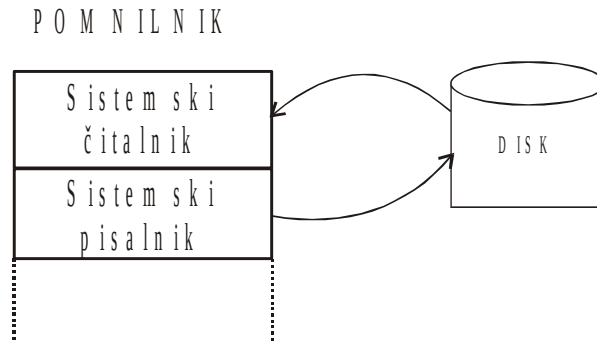
## VEČPROGRAMSKI OS

Sočasno delovanje periferije in procesorja (spooling).

- Vse V/I operacije se izvajajo praktično vzporedno
- Nadgradnja enoprogramskih OS, tako da so operacije z počasnimi V/I enotami (tiskalniki) simulirane s posegi na hitrejše enote (diski).

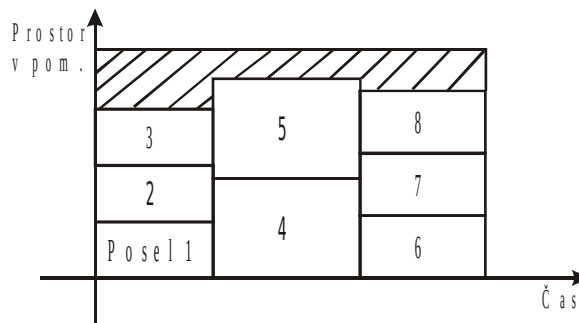
Sistemu sta dodani dve prekinitveni rutini, sistemski čitalnik in sistemski pisalnik, ki preusmerjata vse zahteve za V/I enoto na rezervno področje diska. Kadar računalniku z vhodne enote pošljemo nov paket poslov, sistemski čitalnik na mah prebere celoten vhodni paket v vhodno vrsto na disku.

Zaradi preusmeritve enot uporabniški programi berejo vhodne podatke iz vhodne vrste na disku in ne neposredno iz vhodne naprave. S tem je pohitreno izvajanje programov, da ne čakajo na tiskalnik. Pri takšni zasnovi hkrati tečejo tri opravila, če izvzamemo OS (uporabniški programi, sistemski čitalnik, sistemski pisalnik). Opisani način razvrščanja dovoljuje, da katerikoli sistemski poseg prekine izvajanje uporabniškega programa ali da sistemski čitalnik prekine sistemski pisalnik (prekinitve obratno niso mogoče).



## PRAVO PROGRAMIRANJE

Če hočemo doseči pravo večprogramsko delovanje moramo prilagoditi OS in razen tega imeti strojno opremo z zaščito delovnega pomnilnika, privilegirani ukazi v prekinitvenem načinu delovanja, ura realnega časa in časovnih intervalov. Omenjeni dodatki so prisotni v vseh današnjih računalnikih. OS pri izvajanju več programov hkrati omogoča, da si ti programi delijo računalniške vire v prostoru in času. Vsak od hkrati izvajajočih se programov mora biti avtonomen, nobeden od ostalih ne sme posegati v njegove podatke ali mu celo spreminjati toka izvajanja. OS ostaja nadrejen vsem in odloča o pomembnih posegih oziroma pri upravljanju z vsemi računalniškimi viri. Več opravil časovno deli računalniške vire le, če lahko asinhrono prekinejo delovanje procesorja (z prekinitvenimi mehanizmi). Dejansko je jedro OS skupek prekinitvenih rutin, ki omogočajo, da ima OS ves čas nadzor nad celotnim delovanjem računalniškega sistema.





Ena bistvenih funkcij večprogramskega OS je upravljanje s pomnilnikom. V tem primeru je pomnilnik lahko razdeljen na naslednje načine:

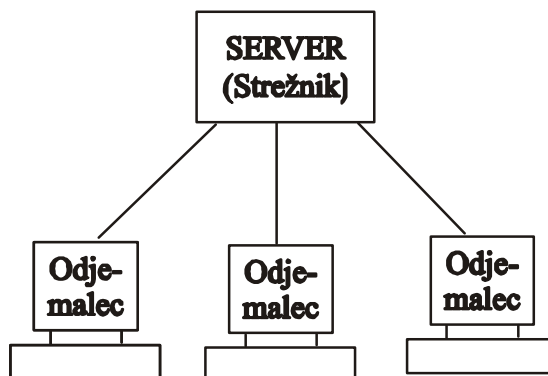
- Stalne patricije
- Več stalnih opravil
- Več spremenljivih opravil
- Dinamično dodeljevanja pomnilnika

### INTERAKTIVNI RAČUNALNIŠKI SISTEMI

V tem primeru imajo uporabniki neposreden takojšni stik za računalniškim sistemom, ker je omogočen dialog med uporabnikom in izvajanjem programov. Komunikacija se opravlja prek interaktivnih ukazov kjer vsak vnos poteka v obliki sporočila. Sporočila se nabirajo v pomnilniškem vmesniku, CPE ga dobi v obdelavo šele po znaku za konec sporočila. Če hočemo da bo več uporabnikov neovirano delalo moramo vpelčjati dodatni časovni proces. Temu pravimo DODELJEVANJE ČASOVNIH REZIN (time-sharing)

### MREŽNI ALI PORAZDELJENI OPERACIJSKI SISTEMI

Več računalnikov je lahko med seboj povezanih v računalniško omrežje, ker lahko med seboj izmenjujejo podatke in uporabljajo komunikacijske programe ki tečejo zunaj OS.



Eden od računalnikov služi kot datotečni strežnik do katerega lahko dostopajo vsi drugi računalniki v omrežju. Tak dostop omogoča mrežni OS. Del OS, ki teče na lokalnih računalnikih mora prepoznati klice, ki posegajo v podatke na strežniku in jih po omrežju preusmerjati tja. Del ki se izvaja na strežniku mora zagotoviti varnost in konsistentnost skupnih podatkov.

PORAZDELJENI OS omogočajo avtomatično razdelitev programskega bremena na več omrežnih računalnikov. To pomeni, da opravljanje neke naloge teče dobesedno v vzporednih opravilih na več računalnikih, delni izidi pa se nazadnje pretočijo prek omrežja in združijo v skupen rezultat.

### UPRAVLJANJE S POSLI IN PROCESI

## POSLE

- **Uporabnik:** zaposluje Os z določenimi nalogami (zahteve po izvajanju določenih programov)
- **Program:** skupek opravil, ki obdela izbrane podatke
- **Podatki:** izhodišča določenega izvajalnega okolja med katerimi vplivajo z ustreznimi parametri
- **Parametri:** so določeni viri s katerimi vplivamo na izvajanje procesov oziroma programa
- **Posel:** je naloga, ki jo opravi računalnik, pri interaktivnih sistemih se posel več uporablja kot sinonim za **proces**. Pri novejših distribuiranih sistemih pa s poslom poimenujemo več nezahtevnih opravil, ki si delijo pomnilniški prostor.
- Program, ki bo opravil zahtevano aktivnost moramo naložiti in pripraviti za izvajanje. Vsak zahteva za izvajanje, ki pride v računalnik iz okolja, sproži vrsto aktivnosti, ki so vezane na dodelitev posameznih računalniških virov (pomnilnik, procesorski čas, zunanji pomnilnik, V/I enote)

## IZVRŠEVANJE POSLOV

Posel se izvaja v naslednjih fazah:

- Vnos posla v računalniški sistem
- Čakanje posla v vhodni vrsti da pride do obdelave
- Obdelava posla
- Končanje posla

Na vhodu v sistem mora sistemski program ugotoviti za kakšen posel gre in pripraviti izvajalno okolje in zahtevane parametre. V paketnih sistemih je to NADZORNIK POSLOV, v interaktivnih sistemih pa UKAZNI TOLMAČ.

Nadzornik poslov uvrsti vsak nov posel v vhodno vrsto, kjer jih obravnava razvrščevalnik poslov. Osnovna naloga ukaznega tolmača je da razpozna ukaz in dodatne parametre, pokliče nalagalnik (LOADER), ki pripravi program za izvajanje. Čakanje poslov v vhodni vrsti je odvisno od aktualnega pravila, ki ga uporablja razvrščevalnik poslov:

- FCFS (first come first served – kdor prej pride prej melje)
- HSFS (highest static priority first served – zunanji statični promet)
- SXFS (shortests execution time firs served – najkrajši najprej)
- Dinamično spreminjajoče prioritete

Obdelava poslov se začne z izvajanjem programa, ki je bil specificiran v prvem koraku posla, imenujemo tudi OPRAVILO (task) oziroma proces.

## PROCESI

Program, ki je naložen v pomnilnik in pripravljen za izvajanje se imenuje proces (opravilo je tudi pravi pojem).

Pri večprogramskih sistemih je aktivno hkrati več procesov (več opravilno delovanje - MULTITASKING). Tudi v MS-DOSu delujejo posamezni procesi v ospredju (FOREGROUND) in v ozadju (BACKGROUND), kot je ukaza print in ukaz realnega časa time.

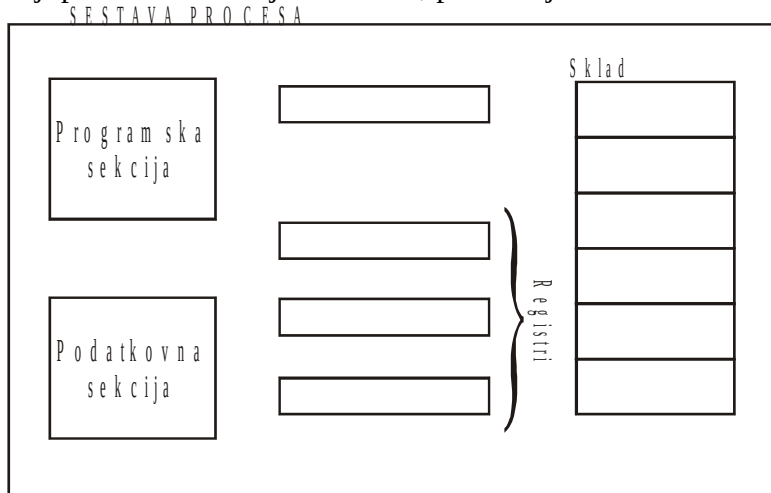
V enoprocesorskih računalniških sistemih procesor opravlja v času eno nalogo. Pri tem gre za delitev procesorja v času za izvajanje posameznih procesov.

V večprogramskem sistemu si programi prostorsko delijo pomnilnik, oziroma uporabniki interaktivnega sistema delijo celoten sistem v času. O tem skrbi OS, ki dodeljuje časovne rezine po pravilu razvrščanja po krožni prioriteti.

## ZGRADBA IN STANJE PROCESA

Po definiciji procesa, ki pravi da je program pripravljen na izvajanje ali se že izvaja v pomnilniku razlikujemo naslednje elemente:

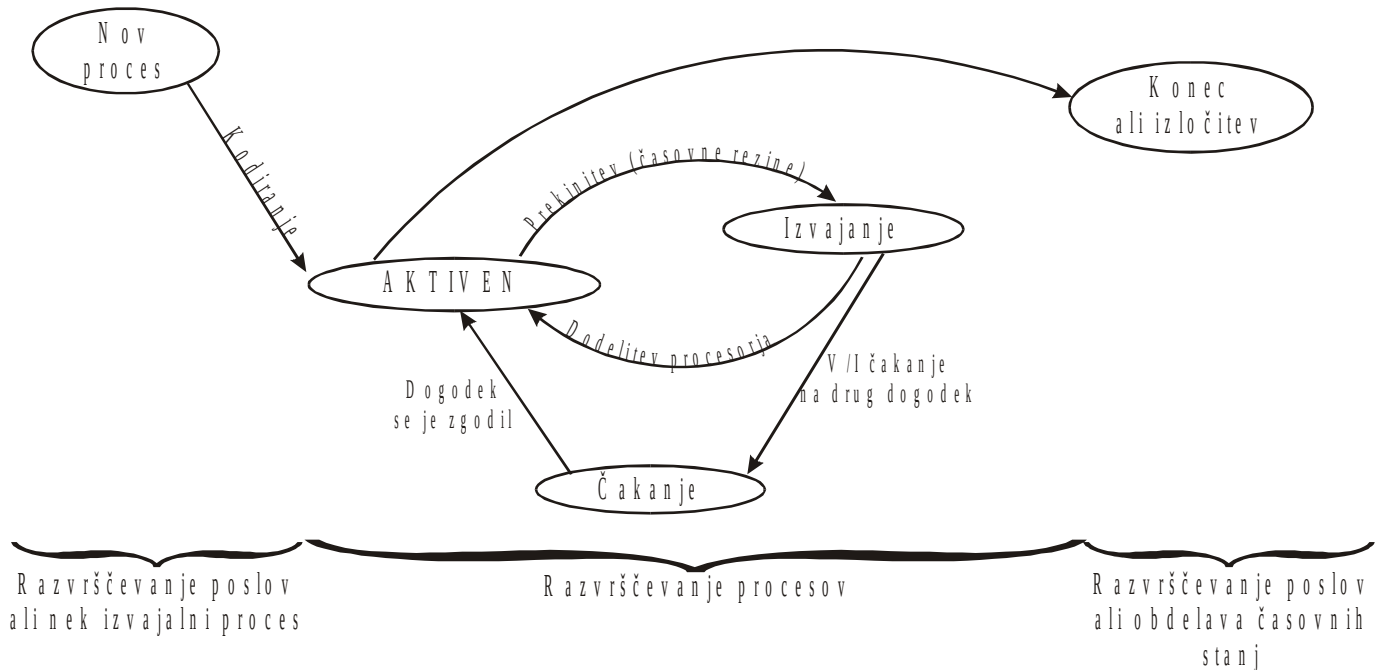
- IZVEDLJIVI MODUL sestavljen je iz podatkovne in programske sekcije
- DINAMIČNE KOMPONENTE
- PROGRAMSKI ŠTEVEC, ki kaže vsebino procesorjevih registrov, ki sodelujejo pri interpretaciji posameznih strojnih ukazov, procesorjev sklad



Za pravilno razvrščanje izvajanje več aktivnih procesov hkrati skrbi RAZVRŠČEVALNIK PROCESOV. Razvrščevalnik spremlja proces od začetka do konca in ga vodi skozi tri stanja:

- aktivno oz. stanje pripravljenosti
- izvajanje
- blokiranje oz. čakanje

Ko je proces ustvarjen (Program pripravljen v pomnilniku za izvajanje) pravimo da je prišel v aktivno stanje. Iz tega stanja ga razvrščevalnik procesov postavi v izvajanje, razvrščevalnik poslov ga lahko začasno deaktivira oz. izloči. Iz tega izvajanja sodijo tri poti: zaradi prekinitve (iztek časovne rezine) ali zaradi normalnega ali nenormalnega konca izvajanja



## PROCESOV NADZORNI BLOK

Da lahko OS nadzoruje izvajanje posameznega procesa mora razpolagati z informacijami zapisanimi v interni podatkovni strukturi, ki ji pravimo **PCB (proces control block)**. Bistveni delci informacij v PCB-ju so :

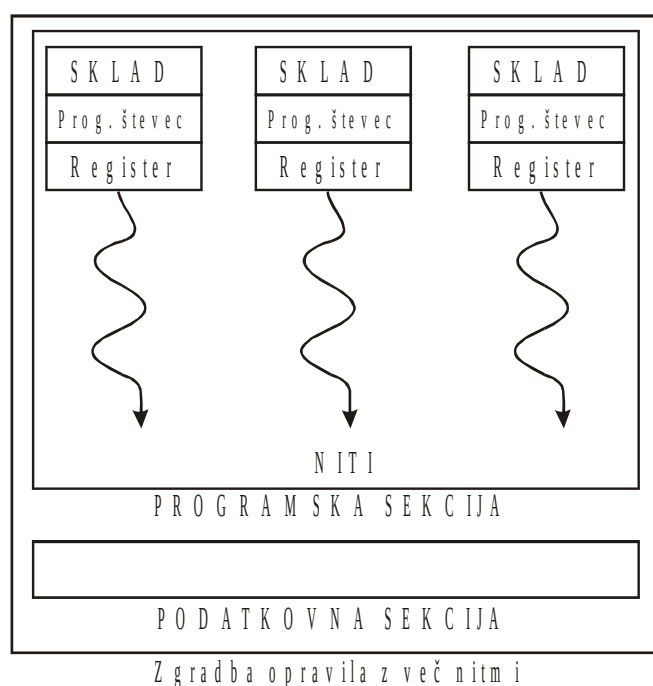
- **Procesno stanje:** aktiven, izvajan, čaka, izločen.
- **Programski števec:** kaže na naslednji ukaz, njegovo stanje shranimo pri vseh prekinitvah
- **Registri** procesorja sodelujejo pri izvajanju strojnih ukazov njihovo stanje se pri prekinitvi procesovega delovanja shrani.
- Informacije o razvrščevanju se nanašajo na procesovo prioriteto.
- Podatki za upravljanje pomnilnika vsebujejo meje dodeljenega pomnilnika
- Vhodno izhodni status opisuje V/I naprave pridružene procesu
- Obračunavanje obdelav se opravi na osnovi potrošenega procesorskega in dejanskega časa

Naloga razvrščevalnika procesov je da po dogovorjenem pravilu poskuša čimbolj zaposliti procesor. Ko se more izvajanje trenutnega procesa prekiniti mora razvrščevalnik med morebitnimi ostalimi aktivnimi procesi izbrati enega, ki mu bo dodelil procesor. Ob tem shrani vse vitalne informacije v njegov PCB potem pa iz PCB-ja naslednjega izbranega procesa prebere njegove podatke. Temu pravimo **PREKLOP (context swich – zamenjava procesa)**. Ta postopek traja precej časa, ki je izgubljen za delo procesov. Preklopi med procesi trajajo med 1 in 1000  $\mu$ s, kar občutno poslabša zmogljivost sistema. Delno se te težave zmanjša z vpeljevanjem NITI. Menjava stanja procesov je povezana v seznam: aktivna vrsta, čakalna vrsta. OS pozna začetne kazalce ne vse vrste (običajno je to enosmerni povezani seznam).

## NITI

Pri enoprogramskih sistemih je možno opravljati en proces v času. Npr. dokler prelagamo podatke z diska procesor čaka. V večprogramskem okolju posel sestavimo iz treh opravil: branje podatkov, obdelava, pisanje podatkov, ki se izvedejo kot samostojni procesi

Ti procesi morajo biti pravilno sinhronizirani, ker jih ne moremo istočasno izvajati. Če bi lahko ti procesi delovali paralelno ne bi bilo težav s preklapljanjem in ločenim pomnilniškim prostorom, če to funkcijo prenesemo v OS pomeni da se nizi programske kode izvajajo vzporedno z ločenimi skladi, programskimi števci in registri. S tem smo definirali niti ali lahki procesi.



Vsak običajen ali tako imenovan težak proces ima natanko eno nit. Niti imajo enaka stanja kot procesi: so aktivne, blokirane, se izvajajo ali pa so končane. Delo z nitmi je lahko neposredno podprto z rutinami v jedru OS ali pa na uporabniškem nivoju z dodajanjem posebnih knjižnic. Pri tem je preklapljanje

med nitmi hitreje na uporabniškem nivoju. Vse niti, ki so znotraj istega opravila delijo skupen pomnilniški prostor in nastopajo usklajeno.

## KOMUNIKACIJA MED PROCESI

V večopravilnem okolju je običajno da neodvisni procesi med seboj sodelujejo (cooperatin) taki sodelujoči procesi so hkrati sočasni (concurrent). V tem primeru komunikacija med procesi poteka skozi skupni pomnilniški vmesnik (kjer je sinhronizacija dosežena z opazovanjem stanja v vmesniku), ali na drug način s pošiljanjem sporočil v OS z možnostjo medprocesne komunikacije prek delovnega pomnilnika, komunikacija med procesi poteka neposredno ali posredno prek komunikacijske zveze. Posredna komunikacija poteka prek OS, ki omogoča tako imenovani nabiralnik – mailbox (lahko se imenuje tudi priključek ali vrata), v katerem lahko vsi puščajo svoja sporočila. Pri tem se nabiralniki ustvarjajo in odstranjujejo, komunikacija pa se usklajuje prek potrjevanja z odgovorom.

Takšna komunikacija se izvaja s klicem oddaljene procedure – RPC(remote procedure call), ko v računalniku z enim procesorjem program sproži klic podprograma mora počakati da se podprogram izvede in vrne pričakovane rezultate.Enako deluje če proces P1 z računalnika A požene s pomočjo klica RPC proces (rutino oziroma proceduro), P2 na računalniku B in počaka da v vrnjenem odgovoru dobi rezultate delovanja tega procesa. Pri tej komunikaciji lahko nastajajo nesporazumi, ki jih potem rešuje programska koda predvidena za to.

## RAZVRŠČANJE PROCESOV

Enoprogramski OS nimajo razvrščevanja na nivoju procesov oziroma računalniških enot. Večprogramski OS dosti boljše ekonomično izkoriščajo računalnikove zmogljivosti z razvrščanjem procesov. Vsi aktivni procesi so povezani v kazalčni seznam, ki tvori aktivno vrsto. Razvrščevalnik procesov izbere naslednjega kandidata za delo s procesorjem, na osnovi podatkov zapisanih v PCB. Narava večine uporabniških programov je takšna, da se zahteve procesorjev in V/I enot ciklično izmenjujejo.

Naloga razvrščevalnika je, da po sprejetem opravilu poišče med aktivnimi procesi naslednjega za izvajanje, dispečer pa temu izbranemu procesorju omogoči delo s procesorjem.

## INFORMACIJSK. KOM. SISTEM (IKS) - TERMINOLOGIJA IN DEFINICIJE

- **Informacijsko komunikacijski sistem** ali **IKS** je enovit sistem z nalogo povezovanja uporabnikov in tehnologije v produktivno celoto
- sestavljen je iz informacijskega dela z nalogo povezovanja ljudi in tehnologije (arhitektura IKSa), ter komunikacijskega dela z nalogo povezati med seboj tehnologijo (struktura IKSa)
- informacijske storitve podpirajo uporabniško komunikacijo (z podatki ki imajo sintakso in semantiko razumljivo človeku)
- komunikacijske storitve podatke predelajo v obliko, primerno za prenose, ki jih omogoča današnja tehnologija

## POVEZOVANJE UPORABNIKOV IN RAČUNALNIKOV

- **uporabnik** je človek, ki komunicira oziroma je povezan z računalniškim in informacijskim sistemom prek različnih vhodno izhodnih enot
- vmesnik med uporabnikom in informacijsko komunikacijsko tehnologijo imenujemo **uporabniški vmesnik** oziroma **uporabniška pristopna točka (UPT)**
- preko informacije na zaslonu uporabnik interaktivno posega znotraj aplikacije in komunicira neposredno s tehnologijo

## RAČUNALNIŠKI SISTEM IN POVEZOVANJE

- računalniški sistem je sestavljen iz sistemskih elementov in uporabniških elementov
- **sistemski elementi** računalniškega sistema so tisti, do katerih uporabnik nima neposrednega dostopa
  - strojna oprema
  - programska oprema
  - sistemski podatki
- **uporabniški elementi** ali aplikacije računalniškega sistema
  - uporabniška programska oprema
  - uporabniške podatkovne strukture

## POVEZOVANJE ELEMENTOV MED RAČUNALNIKI

- **vertikalne povezave – fizične povezave:** uporabnik z uporabo aplikacije in pripadajoče baze podatkov izkorišča vire lokalnega računalnika z podporo operacijskega sistema
- **horizontalne povezave – logične povezave:** uporabnik z uporabo aplikacije na svojem lokalnem računalniku, s pomočjo virov obeh računalnikov in komunikacijskega sistema ima dostop do virov oddaljenega računalnika
- **vertikalne in horizontalne povezave med aplikacijama**, operacijskima sistemoma in strojno opremo lokalnega in oddaljenega računalnika zahtevajo **združljivosti pri povezovanju med lokalnimi in oddaljenimi viri istega tipa:**
  - uporabnik <> uporabnik,
  - aplikacija <> aplikacija,
  - operacijski sistem <> operacijski sistem,
  - strojna oprema <> strojna oprema.

## TEHNOLOŠKA VSEBINA IKS-a

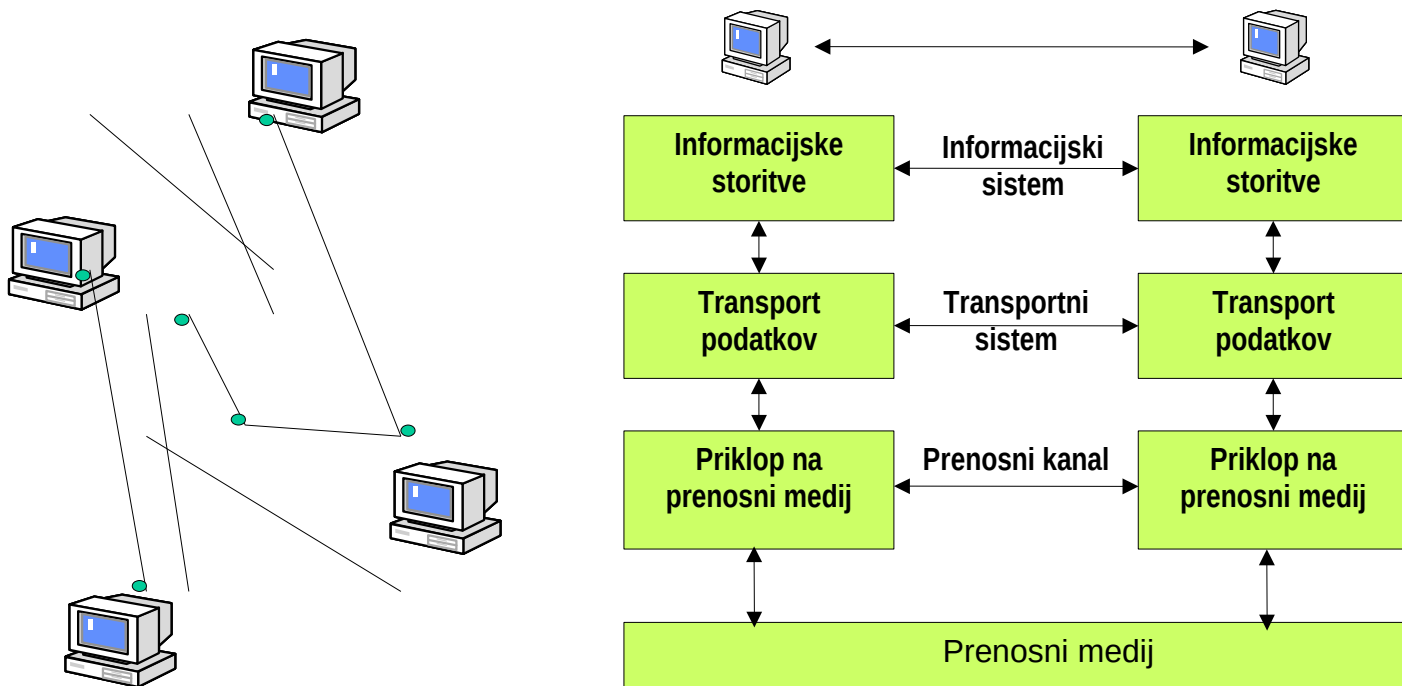
- **komunikacijski kanal**, skupaj tvorijo vmesniki in prenosni mediji, zagotavlja prenos binarnih podatkov
- **računalniško omrežje** (komunikacijski sistem) zagotavlja prenos podatkov iz okvira oddajnega operacijskega sistema/računalnika v okvir operacijskega sistema drugega računalnika
- **transportni sistem** znotraj komunikacijskega sistema zagotavlja pravilno usmerjanje podatkov od izvora do ponora po različnih fizičnih poteh (prenosnih kanalih), glede na to, kakšno **topologijo** (razpored vozlišč in prenosnih kanalov) ima
- **storitve IKS-a** so različne računalniške aplikacije, ki delujejo v okviru različnih operacijskih sistemov in na različnih računalnikih

## STRUKTURA in ARHITEKTURA

- **plasti so funkcionalne celote** ali zaključeni deli IKSa, med seboj hierarhično odvisni
- **arhitektura sistema** opredeljuje plasti, njihova mesta v hierarhiji, logične povezave in funkcionalno vsebino
- plasti so med seboj vertikalno povezane, znotraj plasti so možne horizontalne povezave
- **struktura sistema** opredeljuje njegovo topologijo in izvedbo vertikalnih povezav
- **decentraliziranost sistema** opisuje krajevno ali logično porazdeljenost sistemskih virov enega tipa, (strojne opreme, podatkov, kontrolnih funkcij,..)
- **porazdeljenost sistema** predstavlja kompleksno karakteristiko kot vsoto posameznih tipov decentraliziranosti virov
- **transparentnost sistema** zagotavlja uporabo storitev na enak način ne glede ali se ta izvajata na enem računalniku ali množici računalnikov povezanih v omrežje

### STRUKTURA IKS-a

### ARHITEKTURA IKS-a

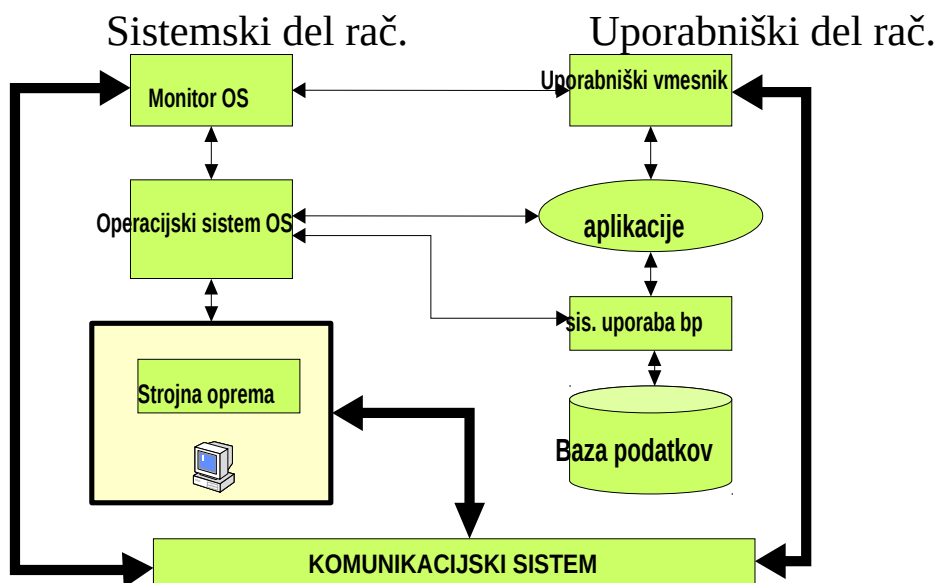


## STANDARDIZACIJA IN RAZVOJ

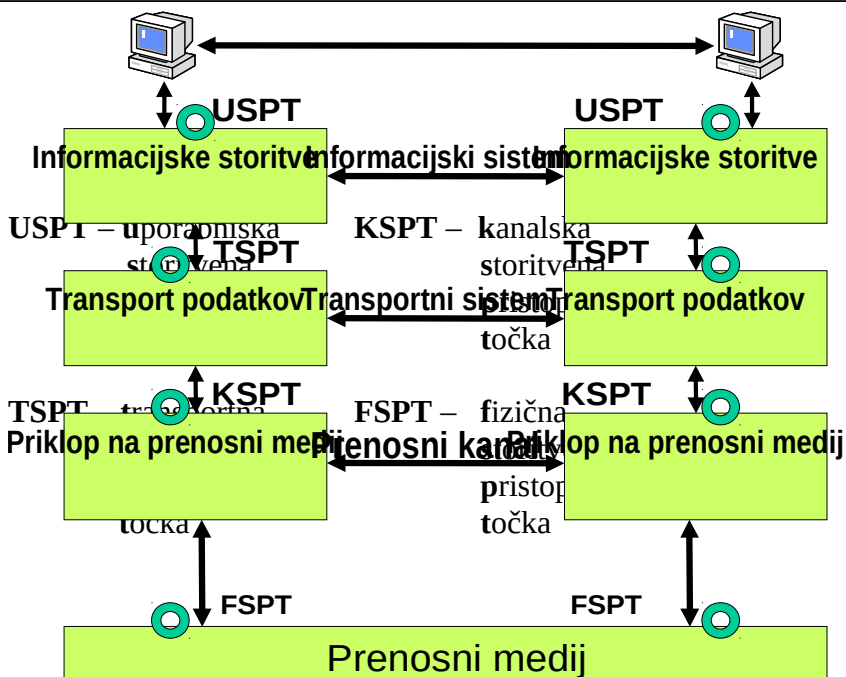
- **uporabniški pristop** - dodajanje komunikacijskih funkcij aplikacijam zunaj operacijskega sistema (~1975)

- **pristop proizvajalcev** - dodajanje komunikacijskih funkcij v operacijski sistem posameznih velikih proizvajalcev (IBM - SNA arhitektura, Digital - DECNET arhitektura) – zaprtost in omejenost na enega proizvajalca (~1980)
- **odprti sistem** je omogočil arhitekturo komunikacijskega sistema neodvisnega od operacijskega sistema in aplikacij kjer preko **standardnega vmesnika** lahko sistemski in aplikacijski moduli izkoriščajo komunikacijske storitve (~1985)
- **standardizirani večplastni modeli IKSa**: OSI, TCP/IP

## ODPRTI KOMUNIKACIJSKI SISTEM



## VEČPLASTNA ARHITEKTURA IKS-a



## VMESNIKI MED PLASTMI

- **USPT – uporabniška storitvena pristopna točka**, zagotavlja uporabnikom transparentnost dostopa in izvedbo določene storitve



- **TSPT – transportna storitvena pristopna točka**, informacijskemu sistemu omogoča izkoriščanje storitev transporta, obenem pa tudi transparentnost izvedbe transportnega sistema
- **KSPT – kanalska storitvena pristopna točka**, zagotavlja prenos podatkov od vozlišča do vozlišča v omrežju (vozlišče je tu vsaka omrežna točka kjer se zaključujejo prenosni kanali), obenem pa tudi transparentnost izvedbe prenosnega kanala
- **FSPT – fizična storitvena pristopna točka**, omogoča priključitev strojne opreme računalnika na prenosni medij, (bakrena žica, koaksialni kabel, optična vlakna, brezžični prenos), preko fizičnega vmesnika, (konektor)

## SPLOŠNO O PLASTI

- plast vzpostavimo, ko identificiramo grupo storitev, ki zahtevajo specifično obravnavo
- vsaka plast mora pokrivati dobro opredeljene storitve
- meje plasti naj bodo opredeljene tako, da se minimizira pretok informacij prek vmesnika med sosednjimi plastmi
- število plasti mora biti takšno da lahko smiselno grupiramo sorodne funkcije v določeno plast in da arhitektura IKSa ostane pregledna
- funkcionalnost posamezne plasti naj bi bila opredeljena tako da se lahko vključi v mednarodno standardizacijo
- do sedaj smo opredelili tri funkcionalne celote ki jih lahko obravnavamo kot plasti:
  - plast informacijskega sistema
  - plast transportnega sistema
  - plast prenosnega kanala

## ZAHTEVA A STORITEV

- storitev je funkcija ali aplikacija, ki jo lahko izvede določena plast
- plast na nivoju N nudi svoje storitve plast N+1, ki leži neposredno pod njo
- tipična storitev transportne plasti je vzpostavljanje zveze med izvirnim in ponornim računalnikom
- primer:
  - zahtevo za določeno storitev (service), poda uporabnik storitve (**service user**), na višji plasti N+1
  - storitev pa nudi izvajalec storitve (**service provider**) na nižji plast N
  - hjahteva se posreduje prek vmesnika, ki ga imenujemo storitvena pristopna točka (**service access point**)

## KOMUNIKACIJSKI PROTOKOL

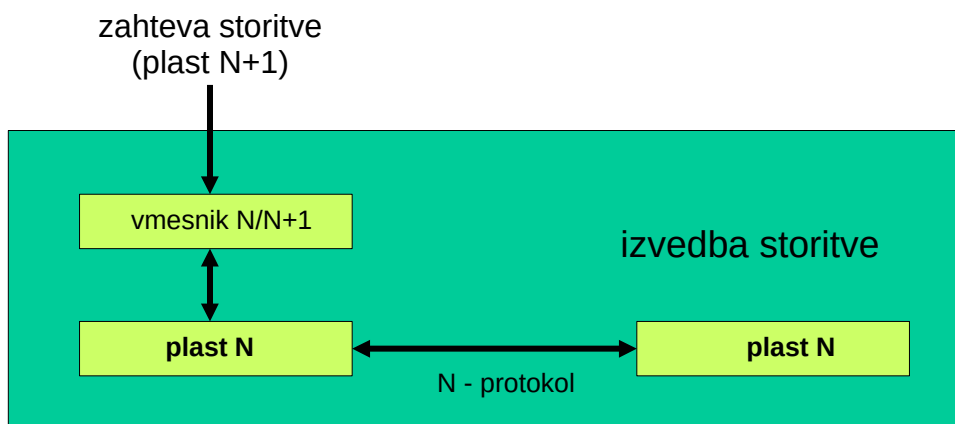
- Izvedba storitve IKS je pogojena z interakcijo med dvema procesoma, ki tečeta v dveh računalniških okoljih (na dveh točkah v omrežju).
- **Komunikacijski protokol** določa pravila po katerih se z interakcijo dveh procesov na plasti N izvaja storitev. Namesto imena plasti pogosto uporabljamo izraz N-protokol.
- **N-protokol** predpisuje način komuniciranja dveh procesov. N-protokol predstavlja izvedbo storitev plasti N (predpisani komunikacijski standard)

## ENTITETNI PAR

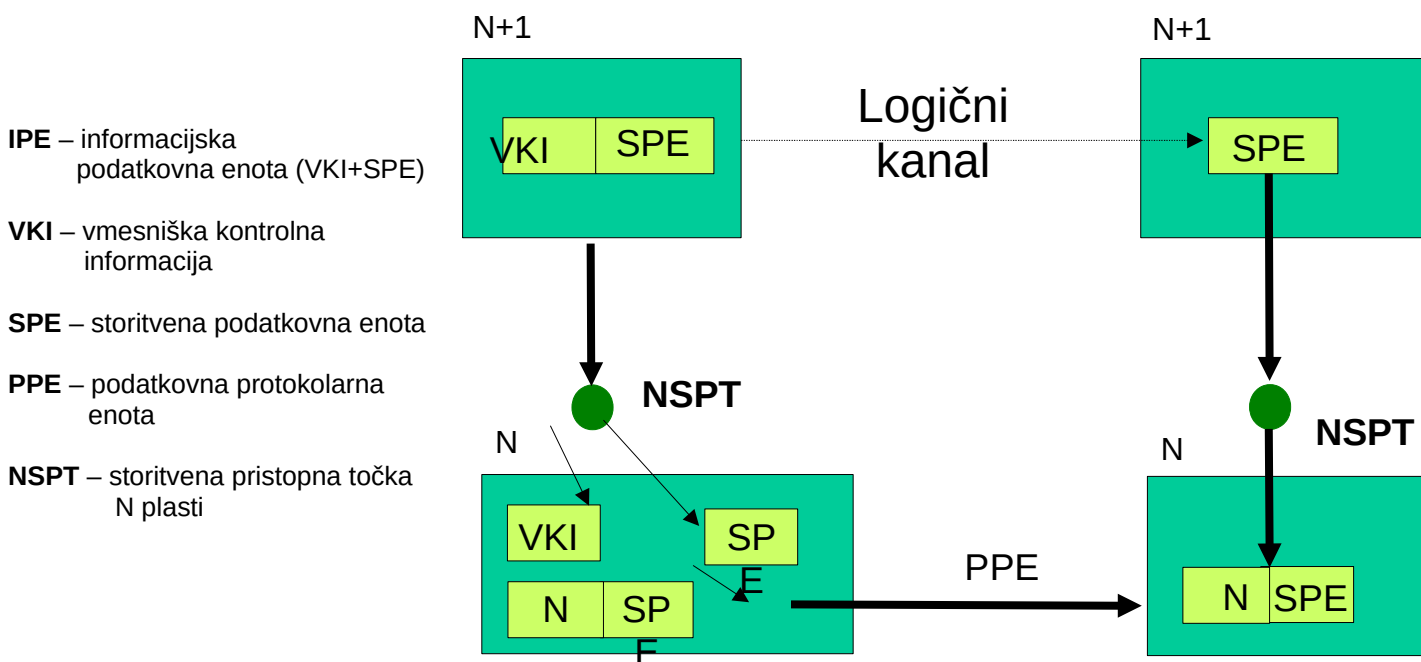
- N-protokol predstavlja logično komunikacijo, ki poteka le med procesi plasti N. Proces, ki komunicira s procesom na neki drugi točki imenujemo **entiteta**. Entiteta je lahko strojni ali programski element, lahko pa je kombinacija obeh.

- Par komunicirajočih procesov imenujemo tudi **istoležna procesa** (peer processes) ali **entitetni par**. Procesna entitetnega para za medsebojno komuniciranje uporabljata N-protokol.
- **Standardni dvosmerni vmesnik** posreduje zahteve in njihove rezultate med višjo in nižjo protokolarno plastjo (kar je bistveni pogoj združljivosti elementov)
- Storitve, ki jo nudi plast N, je za uporabnika formalno opredeljena z naborom osnovnih enot **storitvenih primitivov** (service primitive). Ti uporabniku omogočajo dostop do storitve prek vmesnika storitvene pristopne točke na N plasti, ki lahko vsebujejo sporočilo soležni entiteti, kako naj se ta vključi v izvedbo storitve.

### DVOSMERNI VMESNIK MED PROTOKOLARNIMI PLASTMI



### MEHANIZEM DELOVANJA PLASTI

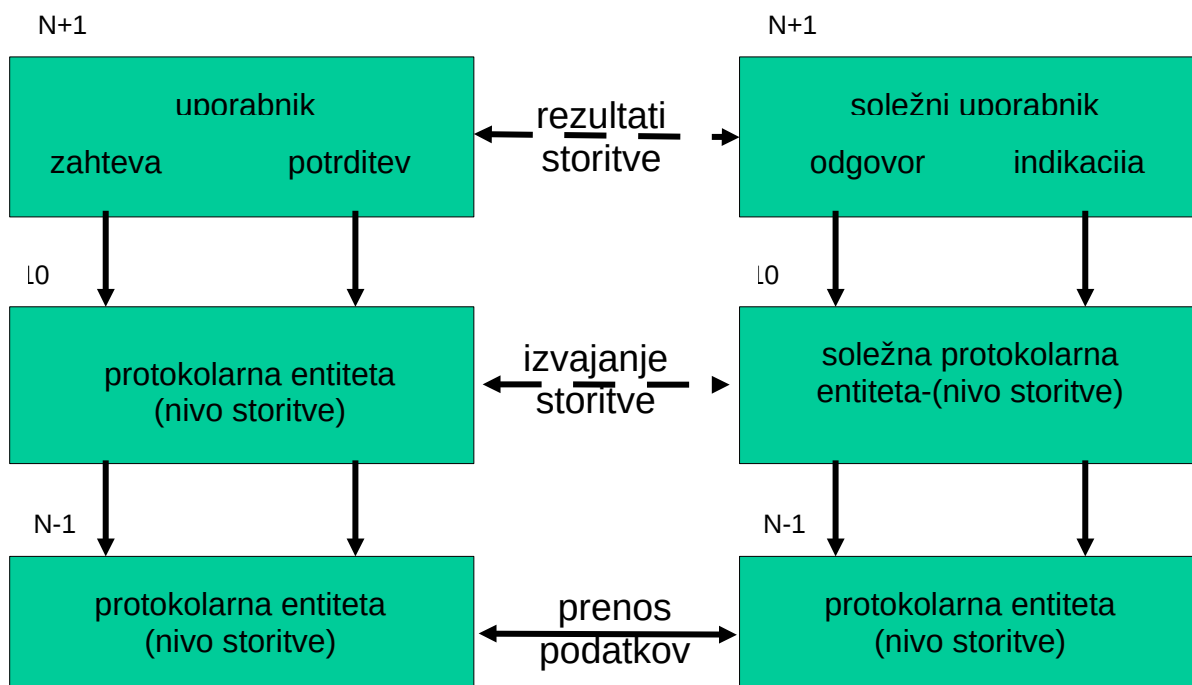


### PODATKOVNE ENOTE » N « PROTOKOLA

V okvir N-protokola se prenašajo standardne skupine podatkov:

- **IPE – informacijska podatkovna enota** (information data unit) ki jo entiteta plasti N+1 pošlje plasti N, je sestavljena iz podatkov zahteve za storitev – SPE (service data unit), in nekaterih vmesniških kontrolnih funkcij VKI (interface control information).
- VKI se uporabi na plasti N kot krmilno informacijo pri izvršitvi storitve in ni del podatkov namenjenih plasti N+1.
- Znotraj plasti N se SPE razdeli v več **paketov** ki jih plast N opremi z dodatnimi informacijami v **glavi** paketa (header). Tipičen podatek, ki sodi v glavo paketa, je zaporedna številka paketa, ki omogoči sprejemni strani sestaviti SPE iz koščkov v originalno obliko.
- SPE ali njen del, opremljen z glavo (paket), imenujemo **protokolarna podatkovna enota** (PPE). Za prenos PPE poskrbi N-protokol ki na sprejemni strani odstrani glavo paketa in preda SPE plasti N+1.
- Tak mehanizem prenosa predstavlja **logični kanal**, ki zagotavlja logično komunikacijo med entitetama na tej plasti.

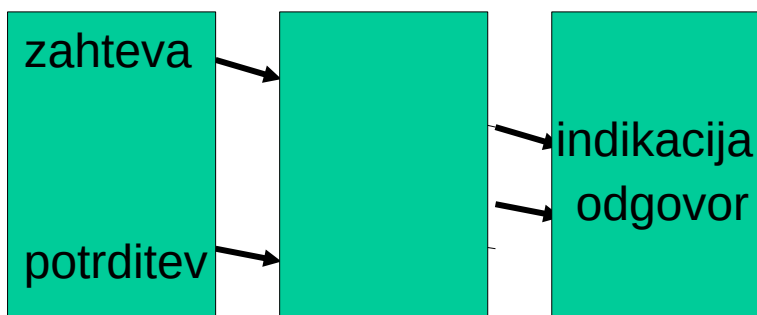
### IZVAJANJE STORITVE V VEČPLASTNEM IKS-u



**logični** prenos podatkov  
**fizični** prenos podatkov

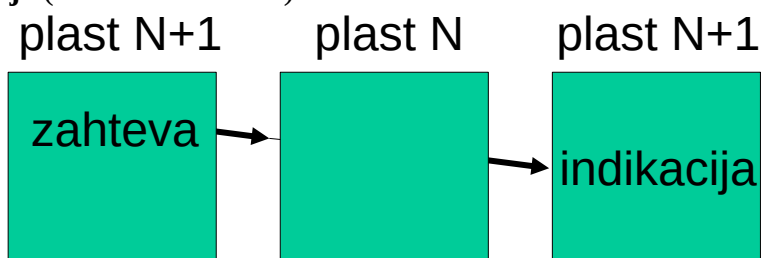
### LASTNOSTI LOGIČNEGA KANALA

**Storitev z potrditvijo** (confirmed service)



- entiteta – oddajnik pošlje **zahtevo** (request) za izvedbo storitve
- soležna entiteta – sprejemnik zazna zahtevo – **indikacija** (indication)
- sprejemnik zahtevo izvrši (uspešno ali neuspešno) in o tem obvesti oddajnik – **odgovor** (response)
- oddajnik odgovor interpretira kot pozitivno ali negativno **potrditev** (confirmation)

**Storitev brez potrditve** (confirmed service)

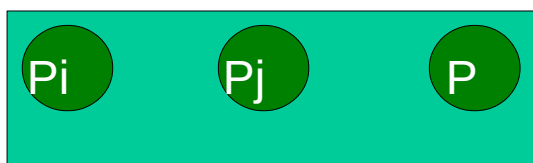


- Ne izvaja se potrjevanje storitve in to ni zanesljiva storitev, kar ni potrebno v vseh primerih (prenos govora).

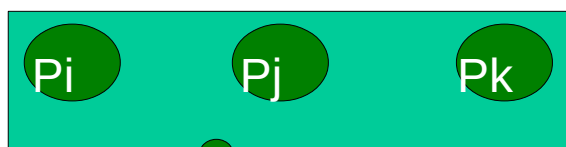
## VZPOSTAVLJANJE ZVEZE

- Entitete lahko komunicirajo s potrjevanjem ali brez. Oba tipa storitev, zanesljive ali nezanesljive, lahko izvedemo z virtualno ali datagramsko povezavo.
- Pri **virtuelni zvezi** se vnaprej vzpostavi med entitetama logični kanal. Med dvema točkama v sistemu je lahko istočasno vzpostavljenih več virtualnih povezav. Paketi v glavi nimajo naslova ponora niti informacije o izvoru (pretakajo se kot po cevi), zato zveza poteka v več fazah.
  - Faza vzpostavljanja virtualne zveze
  - Faza prenosa
  - Faza rušenja zveze
- Pri **datagramski zvezi**, vzpostavljanje zveze predvideva da se vsak paket pošilja neodvisno, lahko po različnih poteh, in se na sprejemni strani individualno identificira. Za identifikacijo ima vsak paket v glavi podatke o svojem izvoru in ponoru.

Povezane storitve – virtualna zveza



Nepovezane storitve – datagramska zveza



## PREGLED RAZVOJA IKS-a

- 1969 – ARPANET (1,5Mbps) – povezava univerz in razvojnih centrov v sklopu obrambnega programa USA
- 1987 – NSFNET (45Mbips) - povezava vladnih institucij, univerz in razvojnih centrov pod okriljem vlade USA

## KOMUNIKACIJSKI PROTOKOLI

- Komunikacijski protokol ali N-protokol je osnovni gradnik večplastne arhitekture IKS-a. Lastnosti komunikacijskih protokolov so razdeljene v dve skupini funkcij in mehanizmov:
  - **Skupina protokolov za odkrivanje in odpravljanje napak pri sprejemu**
  - **Skupina protokolov za kontrolo pretoka podatkov med entitetama**
- Implementacija komunikacijskega protokola (**komunikacijski proces**) je v praksi namreč množica med seboj povezanih procesov.
- Opis komunikacijskega protokola kot osnove komunikacijskega sistema obenem predstavlja tudi opis porazdeljenega sistema.
- **Izvedba protokola** je porazdeljen sistem, v katerem ima vsak proces določeno stopnjo lokalne avtonomije in nek način interakcije z okolico. V splošnem gre lahko za več procesov, ki pa se ponavadi grupirajo v dve večji skupini. Ti dve skupini predstavljata dve komunicirajoči točki (oddajnik, sprejemnik), povezani s prenosnim medijem ali v širšem smislu s prenosnim kanalom.
- V delovanju porazdeljenega sistema, ki ga povezuje komunikacijski podsistem, lahko pride pri komuniciranju do različnih nezaželenih situacij, kot so na primer: nedefiniran sprejem sporočila, smrtni objem, preseganje pomnilniških kapacitet prenosnega kanala, dvoumnost postopka posameznih komunikacijskih faz in podobno. Metode logičnega analiziranja in testiranja porazdeljenosti sistemov so namenjene odkrivanju in odpravljanju takih napak. Glavna naloga protokolskega inženiringa je premostiti vrzel med opisom protokola in njegovo izvedbo.
- Vsak **proces**, je opredeljen z množico vhodnih sporočil, množico izhodnih sporočil (V/I abecedo), s pravili preslikave vhodne abecede v izhodno in s povezavami med procesi.
- **Prenosni kanal** med procesi je največkrat dvosmeren. Za vsako smer prenosa ima čakalno vrsto omejene dolžine. Disciplina čakalne vrste predpisuje, v kakšnem vrstnem redu se iz nje jemlje sporočila. V nadaljevanje bomo predpostavljali najpogosteje uporabljano disciplino: FIFO – prvi noter, prvi ven.
- Oglejmo si do kakšnih napak lahko pride pri izvedbi protokola in katera so značilna stanja porazdeljenega sistema:
  - **Nedefiniran sprejem** je napaka, ki se pojavi, če eden ali več procesov pošlje svojemu paru protokolarno sporočilo, ki ga ta v trenutnem stanju ne zna sprejeti ali nima definirane operacije, ki naj bi sledila sprejemu tega sporočila.
  - **Smrtni objem** je napaka, ki jo spoznamo po tem, da v čakalnih vrstah kanalov ni nobenega protokolarnega sporočila, vsi procesi so v stanju čakanja na sprejem protokolarnega sporočila. Izhod iz takega stanja ni možen, saj noben od procesov ne more oddati sporočila.

- **Presežena dolžina čakalne vrste** ni logična napaka interakcije med procesi sistema, temveč posledica zasičenja enega ali več kanalov med njimi. Čakalna vrsta je polna, zato se protokolarno sporočilo izgubi.
- **Mrtva koda** pomeni, da je opis enega ali več procesov podan preobsežno, saj vsebuje dogodke, ki se v toku interakcije ne bodo nikoli zgodili.
- **Začetno in semi začetno stanje:** vse čakalne vrste v sistemu so prazne. Taka stanja nimajo neposrednega vpliva na logično pravilnost dialoga med procesi, zanimiva so predvsem v fazi testiranja, če želimo postopek strukturirati na več manjših testov.

## MEHANIZMI POTRJEVANJA

- **Mehanizmi potrjevanja** zagotavljajo avtomatsko odpravljanje napak pri povezanih storitvah, ki nastanejo pri prenosu protokolarnih sporočil ali v vsebini ki semantično ne ustreza.
- V dvosmerni komunikaciji sodelujeta entiteti **oddajnik** in **sprejemnik**. Oddajnik oddaja pakete – PPE in sprejema potrditve, sprejemnik pa sprejema PPE in oddaja potrditve.
- Vse različice imajo vgrajeno **časovno kontrolo in parameter ponovitve operacij**: časovna kontrola (interni mehanizem ki se sproži ob vsaki oddaji PPE, in zagotovi odziv po določenem času), oddajnika oziroma sprejemnika, največje število ponovitev (omeji dovoljeno število neuspešnih operacij istega tipa), oddajnika oziroma sprejemnika.
- **Sporočilo** je zaporedje protokolarnih podatkovnih enot.
- **Pozitivna potrditev (ACK – "acknowledge"**, ko sprejemnik spozna sprejeti paket za neoporečnega) in **negativna potrditev (NACK – "Not acknowledge"**, v nasprotnem primeru), sta posebna paketa (PPE), namenjena kontroli izvajanja storitve.
- Oddajnik zagotovi **avtomatsko ponovitev** pošiljanja v primeru, da so sprejeti podatki kakorkoli oporečni. Oddajnik deluje po pravilu dveh osnovnih načinov potrjevanja:
- **Sprotno potrjevanje:** pošiljatelj odda PPE šele takrat, ko sprejme potrditev predhodne PPE, zato ta način potrjevanja imenujemo tudi **pošlji in čakaj**. Dobra lastnost takega načina potrjevanje je, da pri pošiljanju ne more priti do poplavljanja sprejemnika s podatki, saj sprejemnik lahko podatek zavrne ali prekliče. Slabost sprotnega potrjevanja pa je počasnost protokola in nizka izkoriščenost prenosnega kanala.
- **Tekoče pošiljanje:** pošiljatelj oddaja PPE, ne da bi čakal na potrditev predhodno oddanih PPE. Dobra lastnost protokola s tekočim pošiljanjem je boljše izkoriščanje prenosnega kanala. Zato ima lahko v določenem trenutku oddajnik nepotrjenih več oddanih PPE, ki jih mora shraniti v **čakalno vrsto nepotrjenih PPE** za morebitno ponovno pošiljanje. Iz istega vzroka morajo potrditve vsebovati zaporedno številko paketa na katerega se nanašajo.

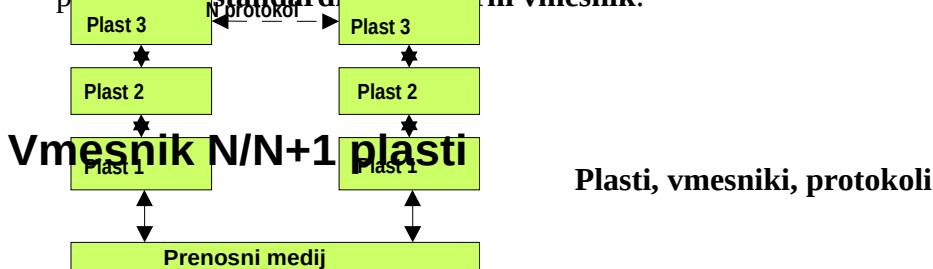
Pri obeh tipih potrjevanja razlikujemo dve različici:

- **posredno potrjevanje** - če sprejemnik potrjuje zgolj pravilno sprejete PPE
- **neposredno potrjevanje** - kadar sprejemnik negativno potrdi nepravilno sprejeto PPE, pozitivno pa pravilno sprejeto.

- Protokol s **sprotnim posrednim potrjevanjem** potrjuje samo pravilno sprejete protokolarne podatkovne enote PPE. Če v določenem času za poslano PPE oddajnik ne sprejme potrditve, posredno razume, da je prišlo do napake pri prenosu.
- Protokol s **sprotnim neposrednim potrjevanjem** v primeru napake sprejemnik odda neposredno obvestilo – negativno potrditveno sporočilo NACK. Tako oddajniku ni treba čakati do izteka časovne kontrole.
- Protokol z **tekočim pošiljanjem – posrednim potrjevanjem** ne razlikuje med izgubo in nepravilnim sprejemom PPE ali ACK. Oddajnik spozna, da paket PPE(N) ni potrjen, ko sprejeme ACK(N) namesto ACK(N+1). Zato ponovno odda PPE(N). Sprejemnik ves čas shranjuje pravilno sprejete pakete v začasni čakalni vrsti do trenutka ko sprejeme vse manjkajoče PPE (zaradi napačnega sprejema). Ko zazna podvojen paket, mora odvečne kopije zavreči. Šele potem jih v pravilnem vrstnem redu uvrsti v svoj delovni pomnilnik.
- Protokol z **tekočim pošiljanjem - neposrednim potrjevanjem osnovna različica**: - sprejemnik potrjuje pravilno in nepravilno sprejete pakete, izgubljenih ne potrjuje. Oddajnik se zave izgube paketa, ko ugotovi da manjka njegova potrditev (ob sprejemu naslednjega). Izgubljene in negativno potrjene pakete ponovno odpošilja kot tudi ob izgubi potrditve (ACK, NACK). Izboljšava te različice zmanjša število potrditvenih paketov.
- **Različica z potrjevanjem zaporedja**: po oddaji NACK(N) sprejemnik ne odda pozitivne potrditve dokler pravilno ne sprejeme ponovljene PPE. Z oddajo ACK(N) sprejemnik potrjuje sprejem vseh paketov iz zaporedja do vključno N-tega (slabost – dovoljuje napačen vrstni red sprejemanja paketov).
- **Različica z ponavljanjem zaporedja**: ko oddajnik sprejeme negativno potrditveno sporočilo NACK(I), ponovno odda vse PPE od vključno PPE(I). Sprejemnik zavrne vse pravilno sprejete PPE, ki so sledili nepravilno sprejeti ali izgubljeni PPE(I). Prednost te različice je ohranjanje vrstnega reda sprejetih PPE, slabost pa pogosto ponovno pošiljanje PPE, ki so že bile pravilno sprejete.

## INTERNETNI PROTOKOLI (IP)

- Za poenostavitev komunikacije med računalniki uporabljamo **večplastno strukturo**. Plasti in njihovo število se razlikuje od omrežja do omrežja. Plasti so med seboj vertikalno povezane, znotraj plasti so možne horizontalne povezave.
- Funkcija posamezne plasti je da servisira sosedno plast. **Storitev** je nabor operacij (storitvenih primitivov), ki jih opravi nižja plast za višjo plast nad sabo.
- Entiteta na obeh straneh, ki spadajo enaki plasti imenujemo **entitetni pari**. Med dvema sosednima plastma leži **standardni dvosmerni vmesnik**.



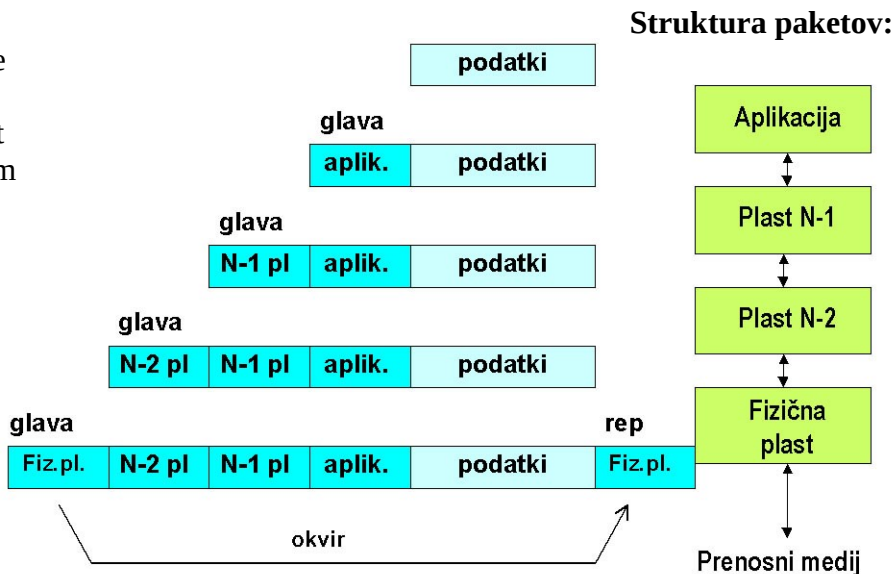
- **Protokol** splošno predpisuje način komuniciranja dveh procesov, oziroma skupek pravil ki definira format in pomen okvirjev, paketov, sporočil ki se izmenjujejo med entitetnimi pari istoležečih plasti.
- **Arhitektura sistema** opredeljuje plasti, njihova mesta v hierarhiji, logične povezave in funkcionalno vsebino.

- Listo protokolov ki jo uporablja določen sistem, enega za posamezno plast imenujemo **protokolni sklad**, (protocol stack).

## STRUKTURA PAKETOV PO ARHITEKTURNIH PLASTEH

- Ko aplikacija pripravlja sporočilo, doda blok ki ga imenujemo aplikacijska glava, (application header), in preda sporočilo N-1 plasti. Plast N-1 bo dodala glavo ki identificira sporočilo in mu daje funkcionalnost plasti N-1. To novo sporočilo preda plasti N-2 in tako naprej. Odvisno od plasti, sporočila drugače imenujemo kot:

- |             |            |
|-------------|------------|
| - sporočilo | - message  |
| - paket     | - packet   |
| - segment   | - segment  |
| - datagram  | - datagram |
| - okvir     | - frame    |
| - celica    | - cell     |



- Na koncu fizična plast (najnižja), okvir podatkov pošlje na prenosni medij k oddaljenemu računalniku.
- Tam se sprejeti okvir prenaša skozi plasti v obratnem vrstnem redu (proti višjim plastem).
- Ko sporočilo potuje skozi različne plasti, se izvajajo storitve posamezne plasti in se odstranjujejo posamezne glave pred prehodom v višjo plast do končne aplikacije ki uporabi sporočilo.

## STANDARDI IN PROTOKOLNI MODELI

- Da lahko pride do komuniciranja med zainteresiranimi udeleženci, morajo vsi uporabljati iste protokole. To dosežemo na dva načina:
  - S pomočjo organizacij, ki so zadožena za standardizacijo protokolov – to so PRIPOROČILA in STANDARDI
  - S pomočjo uporabe razširjenih protokolov – neformalnih standardov
- STANDARD** je več priznanih protokolov, ki jih razumejo vsi tisti, ki želijo komunicirati.
  - STANDARDI *de facto* nastajajo neodvisno od mednarodno priznanih organizacij za standardizacijo – standardi proizvajalcev (**TCP/IP** – Transmission Control Protocol / Internet Protocol)
  - STANDARDI *de iure* nastajajo pod zakonom pod pokroviteljstvom agencij za standardizacijo (**ISO OSI** – Open System Interconnection)

## ORGANIZACIJE ZA STANDARDIZACIJO

- CCITT** – Comitee Consultatif International Telegraphique et Telephonique
- ISO** – International Standard Organization
- ANSI** – American National Standard Institute
- BSI** – British Standard Institute



- **DIN** – Deutsche Industrie Norme
- **IEEE** – Institute of Electrical and Electronics Engineers
- **USM-RS** - Urad za Standardizacijo in Meroslovje **RS**

## MODEL ISO OSI

- Standard je ne lastniški in na področju rač. komunikacij najbolj aktualen. Obsega tipične storitve IKSa.
- Sestavljen je iz sedmih nivojev ali plasti. Vsaka plast omogoča uporabniku določene storitve. Predpisuje vmesnike med lokalno informacijsko infrastrukturo in transportnim sistemom. Plasti so:

1. Fizična plast;
2. Povezavna plast;
3. Omrežna plast;
4. Transportna plast;
5. Plast seje;
6. Predstavitvena plast;
7. Aplikacijska plast;

### 1. FIZIČNA PLAST (PHYSICAL LAYER) (339)

Skrbi za prenos bitov preko prenosnega medija in definira aparaturno opremo za prenos podatkov od rač. Do mreže. Standardi na tem nivoju določajo obliko in vrsto priključkov (DB25), vrsto in razpored signalov (RS 232C, RS 449, V.24...). Ta plast je realizirana z elektronskimi in mehanskimi komponentami.

### 2. POVEZOVALNA PLAST (DATA LINK LAYER) (319)

Prenaša podatkovne okvirje med dvema točkama. Odgovorna je za odkrivanje napak pri prenosu podatkov. Realizirana je s pomočjo programske opreme ali z elektronskimi vezji.

### 3. OMREŽNA PLAST (NETWORK LAYER) (297)

Skrbi za usmerjanje podatkov po omrežju. Določa karakteristike vozlišča in izvaja usmerjevalne algoritme.

### 4. TRANSPORTNA PLAST (TRANSPORT LAYER) (285)

Izvaja prenos podatkov med dvema računalnikoma in skrbi da pri prenosu ne pride do napake.

### 5. PLAST SEJE (SESSION LAYER) (187)

Podpira logično povezovanje oddaljenih procesov med seboj. Končni računalnik, ki želi izmenjavo podatkov z drugim končnim računalnikom, na tem nivoju vzpostavi zvezo in prevzame nadzor nad vzpostavljeno zvezo. Če se zveza ne vzpostavi ali pa prenos ne steče pravilno, mora ponovno poslati podatke oz. vzpostaviti povezavo.

### 6. PREDSTAVITVENA PLAST (PRESENTATION LAYER)

Skrbi za združljivost podatkov različnih računalniških okoljih in za zaščito podatkov. (ASCII, EBCDIC, NAPLP standardi). Na tem nivoju prihaja do kompresije in dekompresije podatkov, kodiranje...

### 7. APLIKACIJSKA PLAST ali NIVO (APPLICATION LAYER)

Uvaja vmesnik med uporabnikom in OSI modelom. Skrbi za hitrost prenosa podatkov, gesla, prijavo na omrežje, elektronska pošta.

## Model TCP/IP

- Ta protokol je bil razvit za privatne in ne komercialne namene, postal je model svetovnega omrežja Internet.
- Omogoča stalen priklop.

- Narejen je za prenos velikih paketov, spremenljive velikosti, predvsem za prenos podatkov, manj uporaben za prenos glasu in drugih storitev v realnem času. Plasti so:
  - **APLIKACIJSKA PLAST** vsebuje storitve in protokole, kot so Telnet, FTP (File Transfer Protocol) – protokol za prenos datotek, omogoča dostop do oddaljenega strežnika SMTP (Simple Mail Transfer Protocol) – prenos sporočil za elektronsko pošto, SNMP (Simple Network Management Protocol) – preprost protokol za upravljanje omrežja. **TRANSPORTNA PLAST** je podobno kot pri OSI modelu izvaja prenos podatkov med dvema računalnikoma in skrbi da pri prenosu ne pride do napake, uporablja TCP in UDP protokole.
  - **INTERNETNA PLAST** omogoča primopredajo IP paketov (datagramov) preko nepovezane storitve, kje se vsak paket usmerja ločeno in vsebuje podatke naslovnika. IP protokol definira format paketov in omenjeno storitev.
  - **RAČUNALNIK/OMREŽJE** je namesto povezovalne in fizične plasti.

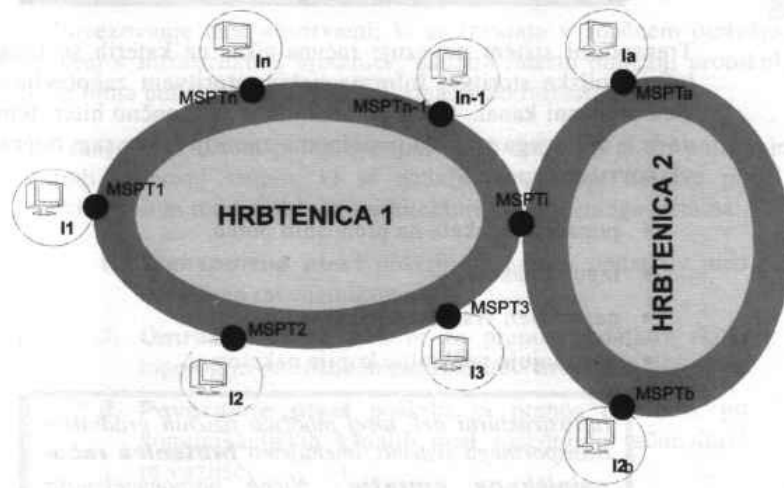
### TCP/IP in internet

WWW PREGLEDOVALNIK										
FTP	Telnet	HTTP	Gopher	WAIS	SMTP	NNTP	MBone	DNS	SNMP	
TCP							UDP			
IP (ICMP, ARP, RARP, ...)										
Native IP	SLIP	PPP	LABP (HDLC)	PPP	ATM	IEEE 802.2 Logical link control				
	PSTN		X.25	ISDN	B-ISDN	802.3 CSMA/CD	802.4 TB	802.5 TR	802.11 Brez-žična	802.6 DQDB
	WAN					LAN				MAN

**Opomba:** V nadaljevanju snov iz 4. letnika

### TOPOLOGIJA IKS-a

Slika prikazuje izhodiščno topološko strukturo IKS-a: dve hrbtenici povezujeta sedem informacijskih infrastrukturnih okolij.



Poleg informacijske infrastrukture je lahko izvor prometa po hrbtenici tudi kaka druga hrbtenica. Tak primer na sliki predstavlja MSPTi. Pristopna točka, ki povezuje dve hrbtenici, se imenuje **medomrežna pristopna točka**. Govorimo o medomrežnem povezovanju .

**HRBTENICA(backbone):**

*Infrastrukturni del, torej množico fizičnih gradnikov transportnega sistema, imenujemo **hrbtenica računalniškega omrežja**. Njena najpomembnejša storitev je zagotavljanje **povezljivosti**.*

**TRANSPORTNI SISTEM:**

*Transportni sistem si lahko predstavljamo kot navidezno napravo, ki podpira fizičen pretok podatkov med informacijskimi storitvami.*

V splošnem je vsak izvor uporabniškega prometa lahko tudi njegov ponor. Količina prometa, ki vstopa v hrbtenico, je enaka količini, ki izstopa iz nje, saj je namenjen od uporabnikov k uporabnikom in ne hrbtenici.

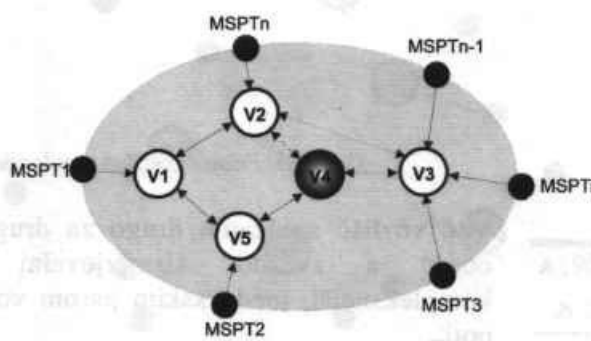
**TOPOLOGIJA omrežja**

**TOPOLOGIJA:**

*Topologija omrežja je množica vseh parov  $(V_i, V_j)$ , kjer sta  $V_i$  in  $V_j$ , vozlišči, med katerima obstaja neposredna fizična povezava.*

Vozlišča in povezave lahko tudi natančneje opišemo (tip vozlišča, kapaciteta povezave ...)

Slika prikazuje primer topologije s petimi vozlišči in dvanajstimi povezavami: lokalna vozlišča so V1, V2, V3, V5, omrežno vozlišče je V4. Vozlišča so **aktivni elementi** hrbtenice, ker izvajajo usmerjanje prometa, prenosni kanali pa so **pasivni elementi**, saj podatke, ne glede na njihovo vsebino med dvema točkama, le pasivno prenašajo.



Topologija omrežja močno vpliva na lastnosti hrbtenice, zato je na tem mestu primerno, da naštejemo najznačilnejše oblike.

## VRSTE TOPOLOGIJ:

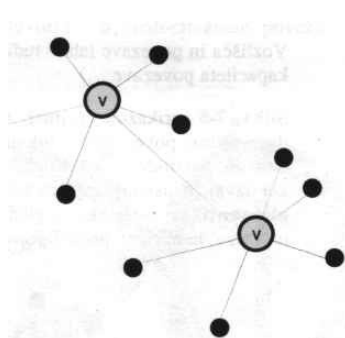
**ZVEZDNA topologija** ima samo eno vozlišče. Njena prednost je enostavnost, saj so usmerjevalni postopki trivialni: med katerima koli pristopnima točkama vodi namreč le ena pot. Veliko omrežij, ki so nastajala postopoma, je začelo s tako topologijo.

**Topologija OBROČA:** Več vozlišč nanizamo drugo za drugim v **obroč**. Primerjajmo obroč z zvezdo. Usmerjevalni postopki so nekoliko kompleksnejši, med vsakim parom vozlišč sta možni dve poti.

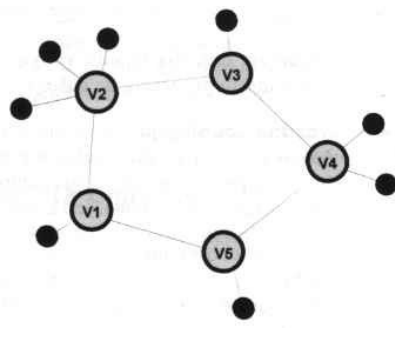
**DREVESNA topologija** zagotavlja enostavne usmerjevalne postopke. Lahko si jo predstavljamo kot večnivojsko zvezdo. V praksi jo srečamo le redko, navadno v omrežjih lokalnega dostopa (povezuje množico terminalov z med seboj povezanimi koncentracijami).

**POLNA topologija** zahteva neposredne povezave med vsemi pari vozlišč. Usmerjevalni postopki v taki topologiji so zelo zahtevni, saj število možnih poti med dvema točkama strmo narašča s številom vozlišč. Tudi njena cena je visoka. V praksi tako topologijo srečamo le redko, večinoma v omrežjih s posebnim namenom.

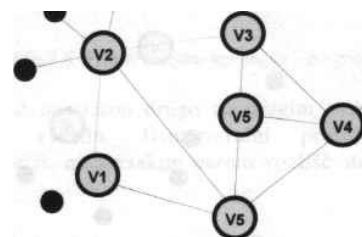
**SPLOŠNA topologija** je v praksi najpogostejša. Vsebuje poljubno izmed podmnožic povezav popolne topologije, ki še zagotavljajo povezanost omrežja. Njena kompleksnost je lahko visoka ali nizka, odvisno od tehnoloških in uporabniških zahtev, ki jih bomo natančneje pregledali v poglavju o načrtovanju omrežij.



Primer zvezdne topologija



Topologija obroča



Splošna topologija

## VZPOSTAVLJANJE KONČNE POVEZAVE

Lastnost povezljivosti hrbtenice zagotavlja možnost **končne povezave** med poljubnima dvema pristopnima točkama. Mehanizem, ki izvaja storitev povezljivosti, imenujemo **usmerjanje**.

Postopek usmerjanja ni enoličen in je odvisen od vrste tehnoloških in organizacijskih lastnosti hrbtenice. Namesto vozlišča se pogosto uporablja tudi izraz, ki meri na funkcionalnost vozlišča: **preklopnik** ali **komutator**. V vozlišču se informacijska enota - **paket** dejansko preklopi z vhodnega komunikacijskega kanala na enega od možnih izhodnih komunikacijskih kanalov. Odločitev za eno od možnih smeri preklopa je odvisna od

- cilja paketa,
- usmerjevalnega algoritma omrežja,
- ne nazadnje tudi od trenutnega stanja v omrežju.

## NAČINI PREKLAPLJANJA

**Preklapljanje** je osnovna naloga vsakega vozlišča. Za razlago različnih tipov preklapljanja moramo najprej opredeliti nekaj osnovnih pojmov, ki jih bomo pogosto uporabljali.

### SPOROČILO:

Omenili smo že, da informacijske storitve med seboj komunicirajo s pošiljanjem blokov podatkov, ki jih bomo v nadaljevanju imenovali **sporočila**. Oblika in obseg uporabniških podatkov največkrat ne ustrežata obliki in obsegu sporočil, ki so primerna za prenos skozi omrežje.

### PAKET:

Transportni nivo mora podatke, ki jih dobi od zgornjih plasti, preoblikovati v **pakete**. Sporočilo je v splošnem sestavljeno iz  $n$  paketov:

$$S = \{p_i\} ; i = 1, 2, \dots, n.$$

**Naslov** sporočila je podatek, ki ga potrebujejo vozlišča (komutatorji) za usmerjanje paketov.

V poglavju o kvaliteti storitev smo opredelili povezano in nepovezano storitev. S tega stališča je vzpostavljanje končne povezave v hrbtenici zelo jasno.

### POVEZAN IN NEPOVEZAN PRENOS:

- V primeru **povezane** storitve se skozi omrežje s pomočjo posebnega paketa vzpostavi povezava med izvorno in ponorno pristopno točko, naslednji paketi sporočila nato potujejo po isti, že vzpostavljeni poti s fiksno topologijo.
- Druga možnost, **nepovezana** storitev, pa opremi vsak paket z naslovom. Paketi se avtonomno prebijajo skozi omrežno hrbtenico do ponora, lahko po poteh z različno topologijo.

## PREKLAPLJANJE POVEZAV

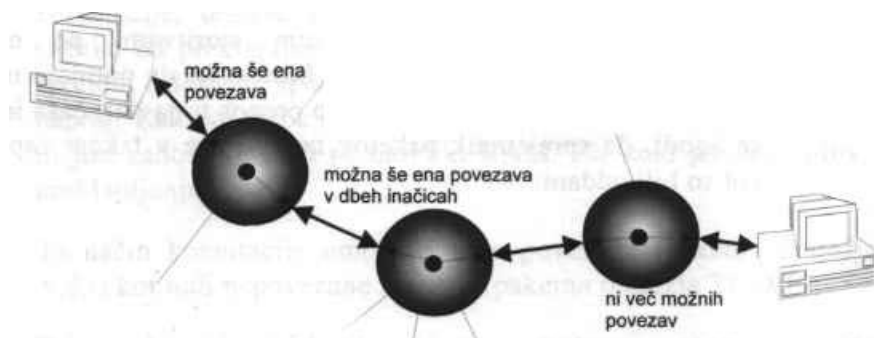
Vozlišče, ki preklaplja povezave, deluje podobno kot mehansko stikalo. Vhodna točka, vhodni prenosni kanal, se "galvansko" priklopi (poveže) na ustrezni izhodni prenosni kanal. Taka omrežja najpogosteje nudijo povezane omrežne storitve. Tipičen primer omrežja s preklapljanjem povezav je telefonsko omrežje, telefonska centrala pa je vozlišče.

Lep primer povezane storitve je telefonski pogovor.

1. Ko zavrtimo številko (torej definiramo naslov), se najprej vzpostavi "galvanska

zveza" med dvema telefonoma, na primer enim v Ljubljani in drugim v New Yorku (**faza vzpostavljanja zveze**).

2. Pogovor predstavlja sporočila, ki se brez usmerjanja prenašajo po že vzpostavljeni zvezi (**faza prenosa podatkov**).
3. Če po končanem prenosu podatkov zveze ne bi porušili, bi vse uporabljene linije omrežja, ki so sodelovale v prenosu, ostale za druge nedosegljive. Sproščanje uporabljenih prenosnih kanalov je nujno (**faza rušenja zveze**).



Namesto telefonov si lahko predstavljamo dva računalnika, ki med seboj izmenjujeta podatke. Tak način povezovanja je najbolj neposreden, saj v povezavi po vzpostavitvi ne sodelujejo vmesni aktivni elementi, kar lahko prispeva k hitrosti povezave. Slaba stran preklapljanja povezav je, da so prenosne poti na razpolago le enemu paru računalnikov, kar zmanjšuje prepustnost omrežja. Na sliki vidimo primer takega tipa povezave.

## TIPI PRENOSNIH KANALOV

Z obravnavo hrbtnice omrežja se je mesto prenosnega kanala natančneje izkristaliziralo. Opredelili smo **lokalne** in **omrežne** prenosne kanale: lokalni povezujejo računalnik z omrežnim vozliščem, omrežni pa povezujejo med seboj dve vozlišči. K prenosnem kanalu se bomo vrnil, ko bomo obravnavali fizične lastnosti prenosnih medijev, na tem mestu pa se bomo zadovoljili le s klasifikacijo prenosnih kanalov, kakršno bomo potrebovali pri obravnavi omrežnih hrbtnic. Glavne lastnosti prenosnega kanala so:

- smernost,
- organizacija podatkov,
- način kodiranja podatkov,
- način sinhronizacije posameznega byta in
- število priključkov na prenosnem mediju.

## SMERNOST PRENOSNEGA KANALA

Prenosni kanal je lahko dvosmeren ali enosmeren.

### **DVOSMEREN KANAL (duplex):**

Dvosmeren kanal ima dve različici:

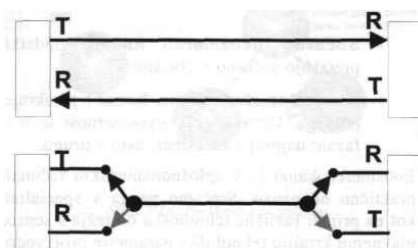
- **Sočasno dvosmeren kanal (ful duplex):** podatki se lahko pretakajo sočasno v obe smeri.
- **Izmenično dvosmeren kanal (half duplex)** je okrnjena različica polnega, ker omogoča dvosmernost le v izmenjujočih fazah: najprej v eno smer, nato v drugo.

### **ENOSMEREN KANAL (simplex):**

Enosmeren kanal je v splošno namenskih računalniških omrežjih praktično nepoznan. Srečamo pa ga v specializiranih omrežjih.

Če na izmenično dvosmernem kanalu pride do napak pri sinhronizaciji oddajnika in sprejemnika, je možno, da oba istočasno oddata podatke, ki se na kanalu zaletijo - pride do **trka ali kolizije**. Interferenca povzroči popačenje podatkov. Polovično dvosmeren kanal moramo poleg osnovne klasifikacije opredeliti tudi glede na možnost kolizije podatkov na prenosnem mediju:

- **Varen:** kolizijsko varen kanal sam odpravlja možnost kolizije.
- **Kolizijski:** kolizijsko nevaren kanal dopušča možnost kolizije.



*Polno dvosmeren kanal (zgoraj) in polovično dvosmeren kanal (spodaj).*

## ZAPOREDNI IN VZPOREDNI KANAL

Prenosni kanal lahko podatke prenaša na dva načina:

- **zaporedni (serijski) kanal:** zaporedno bit za bitom,
- **vzporedni (paralelni) kanal:** omogoča prenos več bitov hkrati (8, 16, 32 ...).

Vzporednih kanali se uporabljajo le v posebnih primerih za povezovanje računalnikov na oddaljenosti do nekaj metrov. Prepustnost vzporednega kanala je v primerjavi z zaporednim večja, vendar si ga na večje razdalje predvsem zaradi višje cene prenosnih medijev ne moremo privoščiti.

Ker interno računalnik uporablja paralelna vodila, je v primeru serijskega kanala potrebna **paralelno-serijska** konverzija na oddajni strani kanala in **serijsko-paralelna** konverzija na sprejemni strani (na sliki).



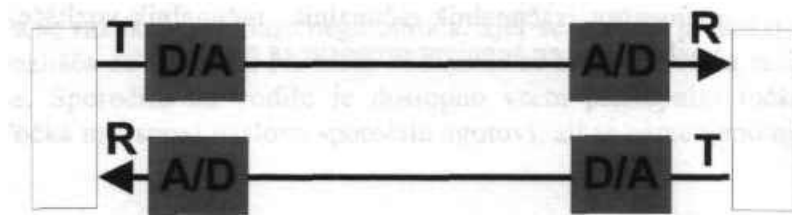
## KODIRANJE PODATKOV

V računalniških omrežjih se podatki vedno prenašajo v binarni obliki. Če sta logična ničla in logična enica kodirani z diskretnimi napetostnimi nivoji, govorimo o **digitalnih kanalih**, če pa sta ti dve logični vrednosti kodirani z analognimi signali, govorimo o **analognih kanalih**.

V primeru analognega kanala je na mestu priklopa računalnika potrebna **digitalno-analogna** konverzija (D/A) na oddajni strani in **analogno-digitalna** konverzija (A/D) na sprejemni strani prenosnega kanala. Današnja tehnologija prenosnih medijev nudi oba tipa prenosnih kanalov. Digitalni

kanali so vedno pogostejši in lahko pričakujemo, da bodo prej ali slej nadomestili večino analognih kanalov.

Glede na tip uporabljenih prenosnih kanalov govorimo o analognih ali digitalnih omrežjih. Kot primer lahko omenimo analogno in digitalno telefonsko omrežje. V svetu računalniških omrežij srečamo najpogosteje hibride obeh načinov.



Digitalno-analogna in analogno-digitalna kanverzija

## SINHRONIZACIJA PRENOSA

Glede na sinhronizacijo govorimo o asinhronih in sinhronih prenosnih kanalih.

- **Asinhroni kanal:** ob prenosu vsakega znaka (byta) se oddajnik in sprejemnik sinhronizirata. Najbolj znan je asinhroni format, pri katerem je vsak znak (7 ali 8 bitov) "oblečen" v en začetni sinhronizacijski bit ter enega ali dva končna sinhronizacijska bita. Tak kanal ima le okrog 50% (ali malo večjo) izkoriščenost. Tak način srečamo le še redko pri nekaterih terminalskih protokolih.
- **Sinhroni podatkovni kanal:** sinhronizacija oddajnika in sprejemnika se dogaja na nivoju paketa, ki ima tipično 128, 1024 ali tudi več bytov. Danes se v omrežjih ta način uporablja najpogosteje. Sinhronizacija se izvrši na podlagi začetnih in končnih 8 bitov vsakega paketa. Zato je izkoriščenost takega prenosnega kanala neprimerno večja.

## ŠTEVILO PRIKLJUČKOV NA PRENOSNI MEDIJ

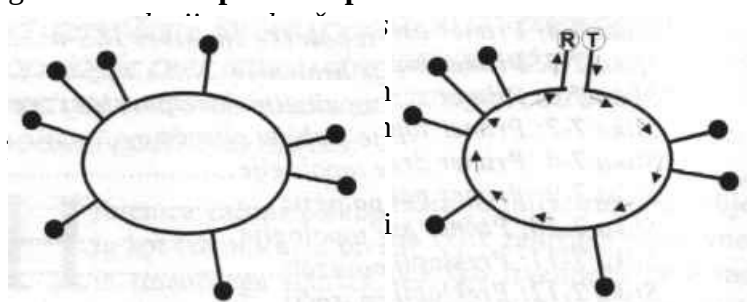
lasičen prenosni kanal povezuje dve točki, **oddajnik** in **sprejemnik**. Tako povezavo bomo imenovali **dvotočkovni prenosni kanal**, saj omogoča priključitev le dveh aktivnih elementov (računalnik-računalnik, računalnik-vozlišče, vozlišče-vozlišče) na en segment prenosnega medija.



Topologija vodila

Pojavom lokalnih ali krajevnih omrežij (LAN) se je spremenil klasični pristop priključevanja le dveh aktivnih elementov na segment prenosnega medija. Tehnologija LAN omogoča priključitev **več** aktivnih elementov (računalnikov, vozlišč) na en prenosni kanal:

govorimo o **skupinskem prenosnem kanalu**.



skupinskim medijem.

srečujemo oba tipa prenosnih kanalov, pogosta omrežju. Skupinski medij omogoča **topologija**

uporablja skupinski medij, pa je **topologija**



Ta se razlikuje od klasičnega obroča, kjer se podatki prenašajo od vozlišča do vozlišča. Na točki oddajnika se obroč na videz razklene. Sporočilo na vodilu je dostopno vsem pristopnim točkam. Točka na osnovi naslova sporočila ugotovi, ali je namenjeno njej.

Skupinski prenosni kanal je izmenično dvosmeren, izvedbe pa so lahko kolizijske ali nekolizijske.

### TRANSPORTNA PLAST

zagotavlja zanesljiv prenos podatkov višjim plastem – po idealnem kanalu – podatki prispejo neoporečni.

Kako poteka :

- višje plasti vzpostavijo končno povezavo prek A,B,C transportnih povezav, ki predstavljajo naslove izvora zahteve.
- Transportna plast identificira zahtevo, aktivira ustrezno omrežno povezavo (posreduje ustrezne podatke za vzpostavitev končne povezave) OSPT je sedaj zasedena, transportna B se preslika v omrežno entiteto
- Omrežna plast aktivira omrežne povezave, potrebne za vzpostavitev končne povezave. Denimo, da na izvorni strani aktivira linijo  $L2$
- Na ponorni strani se zveza vzpostavi prek linije  $Lj$ , kar pomeni, da imamo na povezovalni plasti entitetni par  $(L2, Lj)$ .
- Ta linija se na omrežni plasti preslika v omrežno pristopno točko  $p$ , torej imamo na omrežni plasti entitetni par  $(c, p)$ . Po podobnem razmisleku se na transportni plasti zgradi entitetni par  $(B, Z)$ , ki prek entitetnih parov  $(c, p)$  in  $(L2, Lj)$  fizično poveže odjemalčev in strežnikov proces. Med njima se vzpostavi logična povezava entitetnega para (uporabnik, imenski strežnik).

### LASTNOSTI TRANSPORTNE PLASTI

Razliko med višjimi in nižjimi plastmi pogloblja vsebinska heterogenost višjih plasti.

**Višje plasti vsebujejo:**

- informacijske storitve
- enkripcijo
- oddaljeno prijavljanje
- rešujejo sinhronizacijske probleme porazdeljenih procesov

**Na spodnjih plasteh:** pa se vse vrti okrog povezovanja:

povezovanje končnih strojev, povezovanje lokalnih vozlišč, povezovanje dveh sosednjih vozlišč ali računalnikov, povezovanje različnih tehnologij za prenos podatkov. Paralelo med zg in sp plastmi lahko potegnemo:

- **Zakasnitev pri vzpostavljanju zveze:**  
zajema čas od trenutka, ko je bila podana zahteva do trenutka odgovora nasprotne transportne entitete.
- **Verjetnost napake pri vzpostavljanju zveze:**  
odvisna je od stanja v omrežju, ne toliko od lokalnih možnosti izvornega in ponornega računalnika in transportnega procesa.
- **Prepustnost:**

je število uporabniških bytov, ki se v časovni enoti prenesejo prek transportne povezave. Na propustnost lahko vplivamo z vzpostavitvijo hitrejše, krajše, zanesljivejše končne povezave na omrežni plasti.

- **Zakasnitev pri prenosu:** je čas trajanja prenosa - tipičen parameter končne povezave, vključuje tudi zakasnitev lokalnega procesiranja na transportni plasti.
- **Razmerje ostanka napake:** je razmerje med številom napačno prenesenih paketov proti vsem prenesenim paketom. Na transportni plasti pričakujemo, da bo ta napaka teoretično enaka ničli, praktično pa zanemarljiva (idealni kanal)
- **Zaščita:** opredeljuje stopnjo zaščite, ki jo uporabnik zahteva od transportnega sistema. Spomnimo se na slepi promet, ki smo ga predvideli na predstavitveni plasti, ko smo si prizadevali zakriti obstoj določenih povezav.
- **Prioriteta:** omogoča, da uporabnik opredeli pomembnost posameznih povezav in s tem vpliva na parametre propustnosti in zakasnitve.
- **Robustnost:** opredeljuje število neuspešnih transportnih povezav zaradi internih problemov transportne plasti (ti so praviloma implementacijske narave). Vsebinsko pravzaprav opredeljuje stabilnost transportne plasti.

## MEDSEBOJNA ODVISNOST TRANSPORTNE IN OMREŽNE PLASTI

Omrežna in transportna plast sta med sabo povezani – sta komplementarni. – transportna plast opravlja naloge, ki jih omrežna ne podpira. Bogatejša omrežna plast – enostavnejša transportna. OSI model predlaga pet transportnih modelov.

### KLASIFIKACIJA OMREŽNIH STORITEV

- **tip A**
  - o idealna omrežna plast
  - o ni izgubljenih, podvojenih paketov
  - o ni potrebe po zahtevi za resetiranje omrežne povezave N(etwork) Reset
  - o v primeru napake resetiramo z T(ransport) Reset
- **tip B**
  - o zagotavlja dokaj dober prenos paketov brez izgubljanja
  - o včasih je omrežno povezavo treba resetirati z N-reset
  - o transportna povezava to zazna, interno sproži ponovno vzpostavljanje omrežne povezave
- **tip C**
  - o najnezanesljivejša omrežna storitev
  - o sprejemna sekvenca paketov ni zagotovljena pravilna
  - o možni ponovljeni, izgubljeni paketi
  - o izpad omrežne povezave se rešuje z N-reset

vloga transportne plasti pri teh treh tipih omrežnih storitev

#### tip A

- pri oddajniku se oblikujejo transportni paketi – iz podatkovnih blokov, ki jih dobi iz višje plasti – iz uporabniških podatkov tvori sporočila – prenašajo se po transportnem sistemu
- pri sprejemniku se transportna sporočila preoblikujejo v podatkovne enote – pričakujejo jih višje plasti
- transportna plast resetira transportno povezavo – rušenje končne povezave

**tip B**

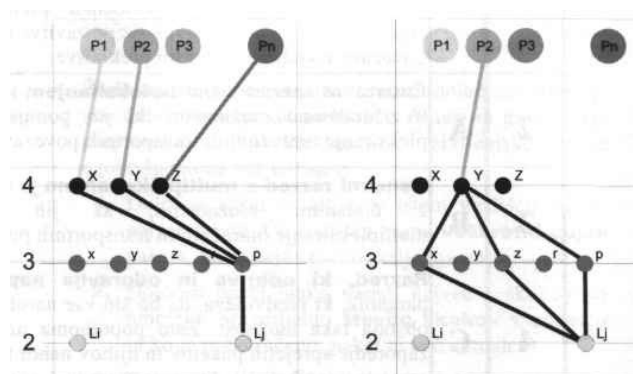
- poleg osnovnih funkcij (tip A) mora zaznavati reset omrežne povezave N-RESET in ustrezno ukrepanje
- sposobna mora biti obnoviti omrežno povezavo

**tip C**

- vsebuje vse zahteve tipa B, transportna plast mora še obvladati:
- urejanje paketov
- izločanje ponovljenih paketov
- ugotavljanje manjkajočih paketov in njihovo ponovno pošiljanje
- 

**TRANSPORTNI RAZREDI**

OSI predvideva storitev multipleksiranja transportnih in omrežnih povezav



omogoča da transportni povezavi pripada več omrežnih povezav – s tem se poveča prepustnost transportne povezave – lahko tudi obratno, le če niso preveč zahtevne (prepustnost, zakasnitev)

na osnovi teh treh tipov omrežnih storitev (A,B,C tipi) je po OSI modelu predvidenih 5 razredov transportnih storitev:

<i>Razred</i>	<i>Tip</i>	<i>Ime in opis</i>
<b>0</b>	<b>A</b>	<b>Enostavni razred</b> predvideva brezhibno delovanje omrežne plasti. Eventualno napako odpravi z resetiranjem transportne povezave T-RESET.
<b>1</b>	<b>B</b>	<b>Osnovni razred</b> odpravlja napake omrežne plasti, četudi se omrežna povezava resetira z N-RESET. Transportna povezava mora torej skrbeti poleg osnovnih funkcij še za evidenco prenesenega zaporedja, ki jo ob ponovni vzpostavitvi omrežne povezave nadaljuje na mestu prekinitve.
<b>2</b>	<b>A</b>	<b>Enostavni razred z multipleksiranjem</b> je razred 0 z dodanimi možnostmi, ki jih ponuja multipleksiranje omrežnih in transportnih povezav.
<b>3</b>	<b>B</b>	<b>Osnovni razred z multipleksiranjem</b> je razred 1 z dodanimi možnostmi, ki jih ponuja multipleksiranje omrežnih in transportnih povezav.
<b>4</b>	<b>C</b>	<b>Razred, ki odkriva in odpravlja napake</b> je paranoik, ki predvideva, da bo šlo vse narobe, če le obstaja taka možnost. Zato popolnoma nadzoruje zaporedje sprejetih paketov in njihov nabor (večkrat sprejeti, manjkajoči). Storitve razreda 1 so njegova podmnožica, multipleksiranja ni.

## OBRAVNAVANJE NEREGULARNOSTI

Teorija je eno, kruta realnost, drugo. Ko začne realnost, nastopijo težave (npr. zakasnitve paketov ob prometnih konicah). Poznamo več strategij za reševanje takih problemov:

- Različni naslovi za vsako povezavo – v nekem časovnem obdobju ne uporabimo istega naslova več kot enkrat – po isti transportni povezavi ne moremo vzpostaviti več omrežnih povezav. Zakasnjeni paketi so tako eliminirani
- Identifikator povezave – vsak paket ima identifikator – ni pripisovanja paketov napačnim povezavam. Slabost: transportna povezava mora skrbeti za evidenco določenih zgodovinskih informacij
- Življenjska doba - vsak paket ima svojo življenjsko dobo – ko jo prekorači, ga omrežje ubije. To lahko stori na več načinov:
  - Ko se določen paket na vozlišču pojavi največ K-krat (večkratni pojav)
  - Ko naredi največ K- skokov v omrežju ga izločimo
  - Določen paket je aktualen največ K – časovnih enot. Ko je njegova časovna znamka prekoračena, umre. Slabost: sinhronizacija ur v vseh vozliščih
- zaporedne številke paketov – pošiljanje se ureja po protokolu z drsečim oknom. Kako to deluje: osnovno različico korigiramo tako, da se zaporedja posameznih ciklov ne začenjajo vedno z 1, temveč tako, da se zaporedne številke posameznih ciklov pojavijo dovolj poredko, da je obstoj dveh paketov z isto zaporedno številko znotraj ene povezave praktično nemogoč. Odstranjena je tudi možnost, da bi ponovljeni paket razumeli kot nek drug paket.

## 9.2.1 SPLOŠNE UGOTOVITVE

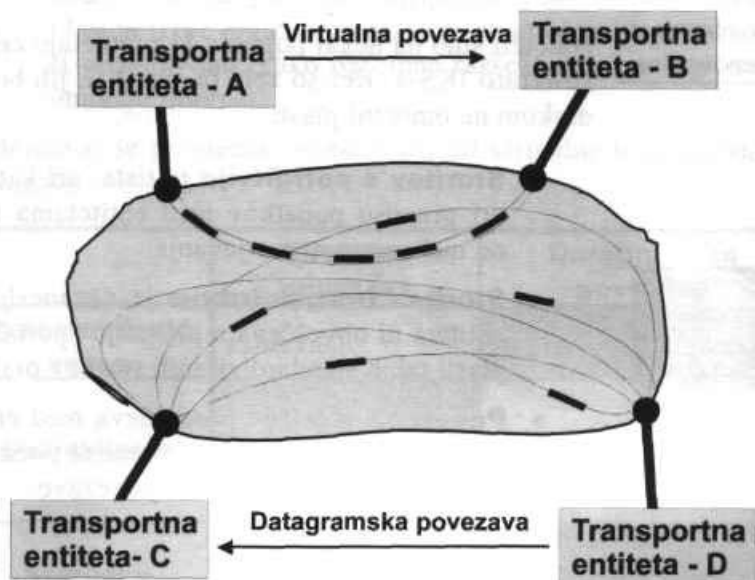
končna povezava – zaporedje vozlišč od izvirnega lokalnega vozlišča do ponornega lokalnega vozlišča – med sabo povezana z prenosnimi kanali.

V topologiji se lahko ustvari med parom točk več končnih povezav, kot jih na prvi pogled nudi fizično omrežje. Po isti fizični poti lahko namreč teče več povezav hkrati oziroma se sočasno pretakajo paketi različnih transportnih povezav - govori-mo o **logičnem omrežju**.

V logičnem omrežju se paketi lahko pretakajo **povezano** ali pa **nepovezano** (slika 9-6). V prvem primeru vsi paketi potujejo po isti fizični topologiji, v drugem primeru pa jo lahko uberejo po različnih fizičnih poteh, četudi pripadajo isti logični povezavi.

Slika 9-6: Povezan in nepovezan

prenos paketov.



Pri obravnavanju osnovnih lastnosti IKS-a in tudi pri klasifikaciji storitev smo omenjali povezane in nepovezane storitve. Kot povezano smo definirali storitev, pri kateri se sporočila med entitetama pretakajo po cevi, kar pomeni, da podatkovni paketi ne nosijo ponornega naslova, saj je "cevi" napeljana v pravo "vedro". Podatkovni paketi nepovezanih storitev so avtonomni in se tudi avtonomno pretakajo od izvora do ponora, zato mora vsak imeti naslov ponora.

Na omrežni plasti lahko zelo natančno opazujemo lastnost povezanosti, saj dobijo "cevi" in neodvisne poti bolj oprijemljivo

ozadje.

Slika 9-6 prikazuje logično omrežje, ki lahko zajema več hrbtenic. V njem sta vzpostavljeni dve logični povezavi. Prva je AB, ki je na omrežni plasti povezana in jo imenujemo **virtualna povezava**, druga pa je DC, ki je nepovezana in jo imenujemo **datagramska povezava**.

Naleteli smo na nekaj pojmov, ki prepletajo celotno arhitekturo in strukturo IKS-a. Ker so zelo pomembni, jih bomo ponovili s poučarom na omrežni plasti.

- **Storitev s potrditvijo** je tista, pri kateri entitetni par pri prenosu podatkov med entitetama uporablja enega od mehanizmov potrjevanja.
- **Storitev brez potrditve** je nezanesljiva saj oddajna entiteta ni obveščena o prispetju sporočila. Na omrežni plasti takih standardiziranih storitev praktično ni.
- **Povezana storitev** predvideva med entitetama "cevi", ki zagotavlja sekvenčnost prenosa podatkov. Posebnost te storitve so tri faze vsake povezave:
  1. faza vzpostavljanja,
  2. faza prenosa podatkov in
  3. faza rušenja povezave.
- Na omrežni plasti se povezana storitev imenuje **virtualna povezava**. V hrbtenici se "cevi" zgradi iz vozlišč in povezav, podatkovni paketi pa ne potrebujejo ponornega naslova.

- **Nepovezana storitev** je zasnovana na avtonomnosti podatkovnih paketov. Ti neodvisno potujejo od izvora do ponora, zato potrebujejo ponorni naslov. Smer napredovanja se na omrežni plasti ugotavlja sproti v vsakem vozlišču. Povezavo imenujemo **datagramska**.
- Fizično omrežje predstavlja množico med seboj povezanih vozlišč, ki oblikujejo topologijo omrežja. O fizičnem omrežju govorimo izključno v okviru omrežne plasti (struktura hrbtenice).
- **Logične povezave** so opredeljene z naslovnim prostorom določene protokolarne plasti. Logičnih povezav je možnih toliko, kolikor je naslovnih parov.
- **Logično omrežje** je množica logičnih povezav, ki jih lahko vzpostavimo na transportni plasti. Na drugih plasteh izraz logično omrežje ni udomačen. Poskusimo si predstavljati, kako ogromne razsežnosti ima logično omrežje Interneta.

Oglejmo si še pregledno tabelo lastnosti virtualne in datagramske povezave.

	<b>Virtualna povezava</b>	<b>Datagramska povezava</b>
<b>Vzpostavljanje zveze</b>	zahtevano	ni potrebno
<b>Naslavljanje</b>	podatkovni paket nosi le identifikacijo virtualne povezave	vsak paket nosi popoln naslov izvora in ponora
<b>Statusna informacija o zvezah</b>	vozlišča (omrežje) hrani podatke o vzpostavljenih virtualnih povezavah	vozlišča se ne ukvarjajo s končnimi povezavami, temveč s posameznimi paketi.
<b>Usmerjanje</b>	na nivoju povezave	na nivoju paketa
<b>Nadzor napak prenosa</b>	opravljen v omrežju	prepuščen transportni plasti - končnemu računalniku
<b>Kontrola pretoka med izvorom in ponorom</b>	zagotavlja jo omrežje	težko izvedljiva

Prednost virtualne povezave je očitno v preprostem napredovanju paketov v vozliščih. Podobno velja tudi za nadzor napak in pretoka, ki ga zagotavlja omrežje. Za omogočanje teh lepih lastnosti pa povezava v vozliščih zahteva možnost zaseganja in sproščanja podatkovnih struktur, potrebnih za ohranjanje virtualne povezave.

Datagramska povezava na nivoju vozlišča ne potrebuje podatkovnih struktur, saj usmerja pakete neodvisno. Velik del nujnih opravil, kot so odkrivanje napak in kontrola pretoka, je prepuščen transportni plasti. Bistvena prednost datagramske povezave je velika trdoživost v primeru odpovedi posameznih povezav ali vozlišč, saj se paketi le preusmerijo na del omrežja, ki ni prizadet. Prilagodljivost je velika tudi, če pride do preobremenjenih povezav ali vozlišč.

## POVEZOVALNE TABELE

Podatkovne strukture virtualne povezave, ki se zgradijo v vozliščih ob vzpostavljanju, imenujemo **povezovalne tabele**. Denimo, da se je v omrežju vzpostavilo sedem virtualnih povezav: štiri iz vozlišča A in tri iz vozlišča H. Nastajale so po vrstnem redu, kot so oštevilčene. V začetku ni vzpostavljena še nobena. Na osnovi izvornega in ponornega naslova na omrežni plasti ne moremo razlikovati paketov posameznih zvez, saj so naslovi v našem primeru enaki za zveze 1, 2 in 3 oziroma za zvezi 6 in 7. Preden nadaljujemo s pojasnjevanjem nastajanja virtualne povezave, definirajmo pojem virtualnega kanala.

*Virtualni kanal je identifikator, ki omogoča razlikovati pakete posameznih virtualnih zvez.*


Kot bomo videli kasneje, je virtualni kanal element prenosnega kanala, vozlišča pa informacijo o virtualnem kanalu uporabljajo v okviru virtualne zveze.

*Slika 9-7: Primer omrežja s sedmimi končnimi povezavami.*

Povezovalne tabele imajo naslednjo obliko: v prvem stolpcu je številka (identifikator) virtualne povezave, v drugem je oznaka vozlišča - izvora paketa, v tretjem oznaka logičnega kanala, po katerem paket prihaja v vozlišče, v četrtem je oznaka naslednjega vozlišča, kamor bo paket nadaljeval pot, v petem stolpcu pa je oznaka logičnega kanala, po katerem bo paket zapustil vozlišče.

Vozlišča morajo imeti za vsako končno povezavo v tabelah preslikovalne podatke med posameznimi virtualnimi kanali dveh prenosnih kanalov, ki se "stikata" na določenem vozlišču.

Ugotovimo tudi, da je potrebna faza sproščanja zveze. V tej **fazi** se podatkovne strukture spraznijo, kar pomeni, da se sprostijo virtualni kanali.

 5-HFEG

## USMERJEVALNI POSTOPKI

Nekaj osnovnih pojmov, povezanih z usmerjanjem:

- **Usmerjevalni postopek** predstavlja način določanja poti prometa od izvorne do ponorne točke v omrežju.
- **Statično** usmerjanje je časovno nespremenljivo, ne glede na trenutno obremenjenost povezav.
- **Dinamično** usmerjanje pri določanju poti upošteva trenutno obremenjenost povezav.
- Tako statično kot dinamično usmerjanje se lahko izvaja **po eni ali po več poteh** hkrati.
- Algoritem **najkrajše poti** je statičen usmerjevalni algoritem, pri katerem vsi paketi potujejo od izvora do ponora po najkrajši poti.
- **Optimalni** usmerjevalni algoritem je tisti, pri katerem so vse povezave približno enakomerno obremenjene.

## USMERJEVALNE TABELE

Podatkovne strukture (povezovalne tabele) v vozliščih se v fazi vzpostavljanja virtualne povezave vzpostavljajo dinamično, v fazi prenosa podatkov so statične, ob zaključku prenosa podatkov pa se sprostijo (spraznijo).

Povezovalne tabele nastanejo na osnovi **usmerjevalnega algoritma**. Usmerjevalni algoritem potrebuje določene podatke, ki so porazdeljeni po vozliščih hrbtnice.

Tabela 9-3 prikazuje usmerjevalne tabele za vsako vozlišče posebej.

3-1 Usmerjevalne tabele

	G	E	H→B	B		A→D	F
A→C	B	A→C	C			H→G	G
						H→B	B
<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>			<b>E</b>	

Tabela

A→D	H	A→D	E	A→D	D
	C			H→G	F
H→G	E			H→B	D
H→B	E				F
<b>F</b>		<b>G</b>		<b>H</b>	

9-

Enota podatka, ki vozlišču določa usmerjevalno informacijo, je sestavljena iz dveh delov:

- Prvi del definira izvorno in ponorno vozlišče. H->B pomeni, da gre za končno povezavo med vozliščema H in B.
- Drugi del opisuje smer nadaljevanja poti v posameznem vozlišču. V vozlišču F (H->B, E) pomeni, da paketi, ki pripadajo končni povezavi H->B, nadaljujejo pot iz vozlišča F v vozlišče E.

Oglejmo si natančneje usmerjevalno tabelo vozlišča F:

- če gre za končno virtualno povezavo A->D, pošlji paket v vozlišče H ali C (npr: naj gre v C vsak peti paket),
- če gre za povezavo H->G, pošlji paket v vozlišče E, prav tako pa tudi, če gre za povezavo H->B.

Pomembne ugotovitve o usmerjevalnih podatkih, ki jih lahko potegnemo iz tega primera, so:

- Ne glede na tip vzpostavljene končne povezave (virtualna ali datagramska) v vozliščih potrebujemo **usmerjevalne podatke** (Tabela 9-3).
- Le v primeru virtualne končne povezave potrebujemo tudi podatkovne strukture, ki opredeljujejo preslikavo paketov med vhom in izhodom vozlišča - **povezovalni podatki**.

## POPLAVLJANJE

Usmerjanje s poplavljanjem deluje tako, da vozlišče vsak sprejeti paket pošlje v vse možne smeri, na vse izhodne kanale. Od tod ime algoritma **poplavljanje**. Uporaben je samo v posebnih načinih delovanja in posebnih omrežjih (na primer v vojaškem omrežju, ko po vozliščih že padajo bombe in je potrebno čim dlje zagotoviti dostavo sporočil). Tak način usmerjanja podatkov predstavlja za omrežje veliko obremenitev, zato naj bi bila sporočila kratka (napad, organiziran umik, ...) Pomembno je tudi, da učinkovito uničujemo pakete, ki nimajo možnosti, da pridejo do cilja: če vemo, da je od izvora do ponora največ 5 skokov, uničujemo vse pakete, ki so opravili že več kot pet skokov.

Izpeljana različica je **selektivno poplavljanje**. Pakete se razpošilja le na linije, ki so približno v pravi smeri (proti ponoru), torej so najbolj obetavne. Mednje na primer ne sodi smer, od koder je prispel paket.

## MEDOMREŽNO POVEZOVANJE

Funkcijo medomrežnega povezovanja različnih hrbtenic opravlja omrežna plast. Naprave, ki povezujejo različne plasti med seboj, imenujemo:

- Na fizični plasti, kjer je prenesena informacijska enota en bit, je **repeater** (ponavljalnik). Njihova naloga je ojačevanje bitnih signalov, ki slabijo vzdolž prenosnega medija, ali pa konverzija signala na prehodu, na primer med bakrenim in optičnim prenosnim medijem.
- Na povezavni plasti, govorimo o **mostovih**. Njihova naloga je konverzija okvirov med različnimi protokoli in zavračanje kakorkoli poškodovanih okvirov.
- Na omrežni plasti govorimo o **usmerjevalnikih**, ki omogočajo medsebojno povezovanje različnih omrežij. Obvladujejo več deset različnih protokolov, ki so v praksi najpogostejši.



- Na višjih plasteh srečamo **protokolarne konverterje**, ki povezujejo različne transportne in aplikacijske protokole. Danes so zelo zanimivi konverterji med Internetovo elektronsko pošto SMTP in OSI-jevo inačico elektronske pošte X.400.

Posebna oblika povezovanja tehnološko različnih hrbtenic je **TUNELIRANJE**. Paketi se naložijo na poseben paket, ki po svojem protokolu omrežne plasti prenaša pakete med dvema pristopnima točkama. Z drugimi besedami, paket "tujega" protokola prenesemo kot podatke po pravilih domačega protokola.

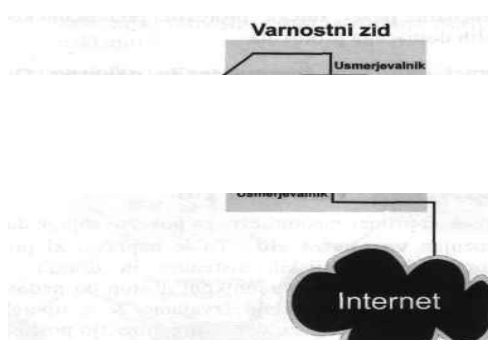
Drugi princip je **FRAGMENTACIJA PAKETOV**. Osnovni paketi se razdelijo, združijo ali preoblikujejo v obliko, ki je domača določenemu tipu omrežja. Na izstopu iz omrežja moramo pakete vrniti v izvorno obliko, ki jo pričakuje ponor.

## VARNOSTNI ZID

Poseben primer medomrežnega povezovanja je danes zelo pogosto omenjan **varnostni zid**. To je naprava, ki preprečuje zunanje zlorabe informacijskih sistemov in drugih virov IKS-a ter preprečuje svojim uporabnikom dostop do nedovoljenih zunanjih virov.

Slika 9-10 prikazuje izvedbo take naprave. Sestavljajo jo tri aparature, ki naj bi zagotovile varnost omrežja pred nepridipravi. Vsak paket je filtriran dvakrat, poleg tega pa gre še skozi aplikacijski konverter.

- **Filtra**, ki sta pravzaprav usmerjevalnika, sta namenjena nadzoru legalnosti paketov, ki se porajajo znotraj omrežja in ga zapuščajo, obenem pa onemogočata nekontroliran vdor nezaželenih paketov v omrežje.
- **Aplikacijski konverter** omogoča na aplikacijski plasti nadzor storitev, ki naj bi se uporabljale v omrežju. Na primer, če želimo, da ljudje med delovnim časom ne uporabljajo storitve WWW, bo aplikacijski konverter zavrgel vse pakete, ki pripadajo tej storitvi.



Slika 9-10:  
Izvedba varnostnega zidu.

## TVORJENJE OKVIROV

Eden od načinov tvorjenja okvirov je zasnovan na **štetju znakov** v okviru.

V glavi paketa označimo njegovo dolžino. Sprejemnik lahko s pomočjo štetja sprejetih znakov ugotovi, kdaj je okvira konec

7	p	e	r	i	c	a	5	r	e	ž	e	5	r	a	c	i	4	r	e	p	
7	p	e	r	i	c	a	4	r	e	ž	?	*	*	*	*	*	*	*	*	*	*

Dolžina paketa se lepo prilagaja vsebini, sprejemnik pa nima preveč dela z razpoznavanjem podatkov okvira. Najprej prebere število znakov in ko jih toliko tudi sprejme, pričakuje informacijo o dolžini naslednjega okvira.

Na žalost ima opisan pristop večjo slabost, če pride do napake pri prenosu na mestu, kjer pričakujemo informacijo o dolžini okvira. V našem primeru je petica (binarna ASCII koda 00110101)

prešla v štirico (binarna ASCII koda 00110100). Sprejemnik pričakuje na mestu znaka z informacijo o dolžini naslednjega okvira, zato se zmede in ni druge rešitve kot resetiranje zveze. Z drugimi besedami, ni mehanizma za reševanje take situacije.

Drugi način za določanje meja med okviri je uvedba posebnih znakov, ki označujejo **začetek in konec okvira**. Protokol BSC deluje na ta način: začetek paketa je označen z znakom DLE (*Data Link Escape*), ki mu sledi STX (*Start of Text*); konec paketa pa z DLE in ETX (*End of Text*). Mehanizem je na prvi pogled simpatičen, težave pa nastopijo, če se v podatkovnem delu pojavi DLE-STX ali DLE-ETX.

Oglejmo si tak primer okvira:

DLE	STX	B	L	A	DLE	STX	L	A	DLE	ETX
-----	-----	---	---	---	-----	-----	---	---	-----	-----

Sprejemnik ne bi znal interpretirati vmesnega DLE-STX. Težavo se uspešno obide z vrivanjem dodatnega znaka DLE vsakič, ko se v podatkih pojavi DLE. Po vrivanju dobi okvir obliko:

- Naloga **oddajnika** je torej, da pred vsak znak DLE v podatkovnem delu okvira (omrežni paket) vrine še dodaten znak DLE.

DLE	STX	B	L	A	DLE	DLE	STX	L	A	DLE	ETX
-----	-----	---	---	---	-----	-----	-----	---	---	-----	-----

- Ko **sprejemnik** naleti na DLE, ga izbriše;
- Če izbrisanemu znaku DLE sledi znak STX, gre za začetek okvira.
- Če izbrisanemu znaku DLE sledi znak ETX, gre za konec okvira.
- Če izbrisanemu znaku DLE sledi še en DLE, gre za DLE v okviru podatkov. Mehanizem deluje tudi pri sekvenci več znakov DLE.

V tem primeru je bil najmanjši informacijski delček znak. Pravimo, da je okvir **znakovno orientiran**. Kot informacijska enota v okviru lahko nastopajo tudi biti, tedaj pravimo, da je okvir **bitno orientiran**. V praksi se pogosto kot identifikator začetka in konca okvira uporablja bitna sekvenca **01111110**.

Primer (vrivanja bitov):

0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Tudi v tem okviru se med podatki pojavlja bitni vzorec, ki pomeni konec okvira. Oddajnik zato po vsaki peti zaporedni enici vrine dodatno ničlo - razen na začetku in koncu seveda. Zgornji okvir dobi obliko:

0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**In kako se mora obnašati sprejemnik?**

- Če naleti na šest zaporednih enic, je to gotovo začetek ali konec okvira.
- Če naleti na sekvenco petih enic, izloči ničlo, ki sledi.

Na ta način se zagotovi **transparentnost podatkov**, ki se prenašajo v podatkovnem delu okvira.

V primerih, ko se na fizični plasti posamezen bit kodira redundantno, na primer z dvema bitoma (1=01 0=10), lahko za označevanje začetka in konca paketa uporabimo "**prepovedane**" podatkovne kombinacije (00, 11).

## DVOTOČKOVNI PROTOKOLI

**Protokoli : HDLC, SLIP, PPP, ATM**

SLIP, PPP protokol serijske povezave (internet) – komunikacija poteka med dvema računalnikoma, prenos poteka po IP protokolu omrežne plasti.

- Statične povezave
- Dial-up (klicne) povezave

## ATM

Novejši protokol

Paketi – celice – dolžine 53B – prvih 5 celic glava- vsebuje CRC kodo

- Prenosi 155 Mbit/s, 622 Mbit/s
- Povezavno usmerjen prenos
- Vsebuje OSI storitve tretje in četrte plasti

## KOLIZIJSKI PROTOKOLI

Značilnosti teh protokolov:

1. borba oddajnikov za dostop do skupnega medija
2. trki paketov

## ALOHA

- Uveden 1970
- Oddajnik ne upošteva morebitne zasedenosti kanala – uničita se oba paketa
  - a. oddajnik oddaja kadar hoče
  - b. na cilju oddajnik ugotovi ali je prejet paket enak oddanemu
  - c. če ni se vklopi časovna kontrola
  - d. po njenem izteku ponovi oddajanje

+ preprost, poceni

- nizka prepustnost  
- 18% izkoristka

## RAZSEKANA ALOHA

- Oddajniki oddajajo paket v določenem taktu ne ob poljubnem času – paket je med oddajanjem varen.
- Uvedba dirigenta – centralizacija vitalne funkcije
  - o ranljivost (izpad dirigenta - izpad komunikacij)
  - o slabša prepustnost pri nizkem prometu
  - o 36 % učinkovitost

## CSMA

Kolizijski protokol s prisluškovanjem oddajnega kanala – preverja ali je kanal prost

- ne čaka oddajnega takta
- dirigent ni potreben
- učinkovit tudi pri nizkem prometu

## VZTRAJNI CSMA

Posluša ves čas. Obstaja možnost trka paketa, če le pride do tega, se vklop na oddajniku časovna kontrola, po njenem izteku ponovno pošlje podatke.

- 52% učinkovitost

## NEVZTRAJNI CSMA

Posluša v časovnih intervalih.

+ zelo učinkovit pri velikem prometu

- onemogoča sinhronizacijo trkov

## P-VZTRANI CSMA

Kombinacija obeh (vztrajnega/nevztrajnega)

Vztrajno prisluškuje – zazna prost kanal, z verjetnostjo p odda paket, ali pa z verjetnostjo p počaka in po izteku časovne kontrole odda paket. Ilustracija dva oddajnika želita oddati paket, zaznata prost kanal, vržeta kovanec – kanal ima tisti, ki vrže grb.

## CSMA/CD

Oddajniku omogočimo ugotavljanje trka – izgublja časa z oddajanjem, oddajanje takoj prekine.

- IEEE 802.3 – Ethernet

Pri CSMA protokolih poznamo dve fazi:

- faza prenosa paketa
- faza borbe za medij

## NEKOLIZIJSKI PROTOKOLI

Komunikacija se lahko odvija brez trkov. **Faza borbe** za prenosni medij zamenjamo s **fazo rezervacije**.

**Faza rezervacije:**

- vse udeležence obišče rezervacijski paket, v katerega se vpišejo kandidati, ki želijo komunicirati

**Faza prenosa podatkov:**

- po določenem vrstnem redu pridejo na vrsto, da oddajo pakete.
- vrstni red ni vedno stalen !
- + močan promet – zelo zmogljiv
- pri šibkem prometu – če oddaja le eden, mora čakati na rezervacijski paket

## PROTOKOL Z ŽETONOM (Žeton = paket = vagonček)

1. v začetki je žeton prazen
2. oddajnik naloži v žeton podatke – drugi oddajniki ne morejo oddajati
3. sprejemnik raztovori žeton – odda praznega
4. vsak oddajnik, ki ne želi oddajati, odda prazen žeton naprej

dve izvedbi:

**token ring** – obroč z žetonom

**token bus** – vodilo z žetonom

## PROTOKOLI Z OMEJENO KOLIZIJO

Kolizijski protokoli so dobri pri nizkem prometu, rezervacijski pa so učinkoviti pri gostem prometu.

Protokol z omejeno kolizijo:

- pri šibkem prometu – se obnaša kot kolizijski
- pri gostem prometu – se obnaša kot rezervacijski protokol

Če pride do trka med dvema postajama, jim onemogočimo pravico do oddajanja. Komercialno ta protokol ni razširjen, zanimiv le za specifična omrežja.

## PROTOKOLI ZA HITRE KOMUNIKACIJE

Hitre komunikacije – 100 Mbit/s in več

FDDI – povezave 100 Mbit/s do 200 km daleč brez uporabe ponavljalnika (repeater – ki upočasnjuje prenos, ko ojačuje signal)

Različica IEEE 802.5 – token ring

Dvojni obroč zato, da se zmanjša ranljivost pri izpadu enega od obročev. Komercialno ime: Fast Ethernet

## BREŽŽIČNA IN RAČUNALNIŠKA OMREŽJA

### BREŽŽIČNA IN MOBILNA RAČUNALNIŠKA OMREŽJA

**Brezžično omrežje – prenosni medij ni fizično oprijemljiv**

- radijski valovi
- svetloba

**Mobilno omrežje**

- ni nujno brezžično
- podpira selitve računalnikov

## FIZIČNA PLAST

- najnižja plast
- povezovanje različnih elementov IKS
- urejajo prenos digitalnih podatkov po prenosnem mediju

### NALOGE FIZIČNE PLASTI

- problem prenosa digitalnih informacij
- usklajevanje električnih in drugih standardov na sprejemniku in oddajniku
- pretvarjanje električnih signalov v oblike primerne za prenos po brezžičnem mediju
- problem kodiranja digitalnih signalov z analognimi vrednostmi
- fizični vmesniki (konektorji) za priključitev računalnika na prenosni medij

Fizična plast vpliva na zmogljivost sistema, sodi na področje komunikacijskih naprav. Osnova fizične plasti je prenosni medij, ki povezuje ali več računalnikov po notranjem vodilu ali drugih elementov IKS-a. Prenosni mediji so naprave, ki omogočajo prenos podatkov od oddajnika do sprejemnika – tako je bistvena lastnost prenosnega kanala njegova frekvenčna karakteristika.

## TIPI PRENOSNIH MEDIJEV

### PRENOSNI POMNILNI MEDIJI

Mednje sodijo:

- trakovi, kasete
- prenosljivi diski
- diskete, CD-ji

### PARICA & ZVITA PARICA

Najpreprostejši, najcenejši, če sta dve bakreni žici zviti druga okoli druge, potem je to zvita parica (UTP), Uporaba :

- 10 Mbit/s relativno dolge povezave z uporabo modernih tehnologij
- kot analogna povezava 33kBit/s, 64 kBit/s do 2Mbit/s, na zelo kratkih razdaljah pa do 100 Mbit/s

### **KOAKSIALNI KABEL**

Zgradba: jedro iz bakrene žice, ovito v izolacijo, oklop iz bakrene mrežice

- zelo odporen proti zunanjim elektronskim motnjam, kabel sam ne povzroča elektromagnetnega sevanja
- 1-2 Gbit/s do 1 km daleč, uporaba v Ethernet tehnologiji

### **OPTIČNO VLAKNO**

- najzanesljivejši prenosni medij, prevajajo svetlobne signale, zelo majhno slabljenje signala, nemogoče prisluškovanje mediju
- do 100 Gbit/s

### **BREZŽIČNE POVEZAVE**

Vse te tehnologije širijo elektromagnetno sevanje v prostor.

- radijske zemeljske povezave
- mikrovalovne usmerjene povezave
- infrardeče povezave
- satelitske povezave

### **TELEFONSKA OMREŽJA**

To **največje svetovno omrežje** je velikega pomena pri vzpostavljanju povezavah, ki jih potrebuje IKS za svoje omrežje.

### **ANALOGNA TELEFONSKA OMREŽJA**

Klasična vloga: prenos govora. Pretvorbo iz analognega v digitalni na oddajnikovi strani v analogno iz digitalne na sprejemnikovi strani opravlja naprava, modem imenovana. Da razlikujemo med logično 0 in 1 moramo osnovni pisk spreminjati – modulirati.

### **VRSTE MODULACIJ**

- amplitudna
- frekvenčna
- fazna modulacija

### **DIGITALNA OMREŽJA IN ISDN**

Razlogi za digitalizacijo telekomunikacij:

- pretvorba signala (digitalna/analogna) je zahtevala veliko denarja, energije, prostora
- ponavljalniki analognih povezav so bolj zahtevni kot ponavljalniki digitalnih povezav, ki so hitrejši, cenejši, zanesljivejši.

**PCM standard** omogoča pretvorbo govora v digitalno obliko. Po Nyquistu to pomeni, da moramo imeti vzpostavljen kanal s hitrostjo 64 kbit/s

### **ISDN**

ISDN je mednarodni standard s široko paleto komercialnih izdelkov, katerega cilj je integracija različnih uporabniških storitev:

- prenos govora, slike, gibajoče slike, podatkov s pripadajočimi storitvami
- kanala A,B – kapaciteta 64 kBit/s , servisni kanal D – kapaciteta 16 Kbit/s

- ni pretvorbe iz analognega v digitalni in vice versa, s tem se spremeni filozofija »omreževanja«

## NAJETE IN KOMUTIRANE LINIJE

**Komutirane linije** (klicne linije) imenujemo običajne analogne in ISDN telefonske linije, le da namesto telefona nanje priključimo računalnik.

Klicni modem je tisti, ki opravlja A/D in D/A pretvorbo ter vrtenje telefonskih števil in dviganje slušalke, hitrosti dosegajo do 56 kBit/s.

V primeru ISDN povezave nimamo modema ampak poseben ISDN vmesnik, v osnovni varianti 64 kBit/s

Priključevanje po komutiranih linijah radi uporabljajo tisti, ki nimajo neposrednega dostopa do hrbtenice računalniškega omrežja in tisti, ki samo občasno potrebujejo omrežne storitve in dostop do oddaljenega računalnika.

Najete linije so tiste, ki omogočajo stalno povezavo z določeno informacijsko in komunikacijsko opremo. Na njih lahko zgradimo prenosni kanal med dvema računalnikoma ali pa celo hrbtenico določene topologije.