

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**PROBLEMATIKA ZAGOTAVLJANJA PODATKOV PRI
PORAZDELJENIH PODATKOVNIH BAZAH**

Ljubljana, september 2003

Milan Jeličić

IZJAVA

Študent Milan Jeličič izjavljam, da sem avtor tega magistrskega dela, ki sem ga napisal pod mentorstvom doc. dr. Jurija Jakliča, in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, 24.9.2003

Podpis:

KAZALO

1	Uvod	1
1.1	Opis problematike	1
1.2	Namen in cilji magistrskega dela	3
1.3	Metode dela	3
1.4	Zasnova magistrskega dela	3
2	Porazdeljene podatkovne baze	4
2.1	Podatkovna baza	5
2.2	Površinski podatkovni modeli	6
2.3	Arhitektura porazdeljenih podatkovnih baz	7
2.3.1	Sistem za upravljanje porazdeljenih podatkovnih baz	9
2.4	Porazdelitev podatkov	10
2.4.1	Centralna postavitev podatkov	11
2.4.2	Replikacija podatkov	12
2.4.3	Razdelitev podatkov	13
2.5	Velike podatkovne baze	14
2.5.1	Porazdelitev podatkov glede na ključ	14
2.5.2	Porazdelitev podatkov glede na lastnosti	15
3	Zagotavljanje celovitosti podatkov v podatkovnih bazah	16
3.1	Interferenca med sočasnimi odvisnimi transakcijami	18
3.1.1	Izguba spremembe podatka	19
3.1.2	Sprememba pomena podatkov	19
3.1.3	Napačno branje podatka	21
3.2	Zaseganje podatkov	21
3.2.1	Zaseganje podatkov v dveh fazah	22
3.2.1.1	Mrtva zanka	23
3.2.1.2	Navidezna mrtva zanka	25
3.2.1.3	Živa zanka	26
3.2.1.4	Strategije za odkrivanje mrtvih zank	26
3.2.1.5	Kaskadna razveljavitev transakcij	27
3.3	Časovno označevanje	28
3.3.1	Osnovno časovno označevanje	29
3.3.2	Previdno časovno označevanje	30
3.4	Optimistične metode	30
3.5	Metoda večine	31
3.6	Zagotavljanje podatkov ob napakah med izvajanjem transakcij	32
3.6.1	Zaključevanje v dveh fazah	33
3.6.2	Zaključevanje v treh fazah	35
4	Računalniška omrežja	36
4.1	Omrežni model ISO/OSI	37
4.2	Omrežni model TCP/IP	39
4.3	Omrežne tehnologije po slojih modela ISO/OSI	40

5	Omrežna podpora komunikaciji enega z več sprejemniki	41
5.1	Usmerjeno in razpršeno oddajanje.....	41
5.2	Pregled tehnologije za večsmerno oddajanje.....	42
5.2.1	Optimalna uporaba pasovne širine komunikacijskih povezav	42
5.2.2	Manjša obremenitev strežnikov	43
5.2.3	Manjša obremenitev omrežja in omrežnih elementov	44
5.2.4	Pomanjkljivosti tehnologije večsmernega oddajanja	44
5.3	Aplikacije, ki uporabljajo večsmerno oddajanje	45
5.4	Usmerjanje paketov pri večsmernem oddajanju.....	46
5.4.1	Posredovanje paketov glede na njihov izvor	47
5.4.2	Gradnja distribucijskega drevesa za večsmerno oddajanje	49
5.5	Omrežni protokoli za usmerjanje podatkovnih paketov večsmernega oddajanja	51
5.5.1	Protokol DVMRP.....	51
5.5.2	Protokol MOSPF	52
5.5.3	Protokol CBT.....	54
5.5.4	Protokol PIM	55
5.5.4.1	Zgoščen način delovanja protokola PIM.....	56
5.5.4.2	Razpršen način delovanja protokola PIM	57
5.5.4.3	Dvosmerni način delovanja protokola PIM	58
5.5.4.4	Oddajniško usmerjen način delovanja protokola PIM	59
5.5.5	Protokol MBGP	60
5.6	Zagotavljanje zanesljivosti prenosa podatkov pri večsmernem oddajanju ...	60
5.6.1	Protokol RMTP	62
5.6.2	Protokol PGM.....	63
5.7	Zagotavljanje varnosti prenosa podatkov pri večsmernem oddajanju	65
6	Analiza mehanizmov za učinkovito zagotavljanje podatkov v porazdeljenih podatkovnih bazah	66
6.1	Porazdeljeni sistemi brez podvajanja podatkov.....	66
6.1.1	Komunikacija med porazdeljenimi sistemi brez podvajanja podatkov.....	67
6.2	Porazdeljeni sistemi s podvajanjem podatkov.....	69
6.2.1	Pohlepna replikacija.....	69
6.2.2	Zakasnjena replikacija	70
6.2.2.1	Zakasnjena skupinska replikacija	71
6.2.2.2	Zakasnjena centralizirana replikacija.....	72
6.2.3	Dvonivojska replikacija	72
6.2.4	Komunikacija med porazdeljenimi sistemi s podvajanjem podatkov.....	74
7	Sklep.....	75
8	Literatura.....	77
9	Viri	79

Priloga

1 UVOD

1.1 Opis problematike

Podjetja v svojem poslovnem procesu uporabljajo različne vire podatkov. Pri izvajanju poslovnih procesov v podjetju podatki tudi nastajajo in zaželeno je, da prihaja do njihovega izmenjevanja. Vse te podatke je zato treba nekje hraniti, da so enostavno in hitro dostopni vsem uporabnikom. Z uporabo informacijskih tehnologij in sredstev sodobna podjetja pri svojem poslovanju ustvarjajo in uporabljajo najrazličnejše podatkovne baze za shranjevanje podatkov. Te baze so lahko najenostavnejše kadrovske evidence ali evidence poslovnih sredstev, lahko pa so zelo kompleksne baze, ki služijo podpori poslovnih procesov podjetja.

Količina ustvarjenih in pridobljenih podatkov iz dneva v dan narašča. Posledično se podatkovne baze povečujejo in zasedajo čedalje več pomnilniškega prostora. Poleg tekstovnih podatkov, ki zasedajo relativno malo pomnilniškega prostora in jih lahko s pomočjo tehnik zgoščevanja podatkov še dodatno zmanjšamo, se čedalje pogosteje shranjujejo podatki v multimedijški obliki, pa naj gre pri tem za besedila, oblikovana s sodobnimi urejevalniki besedil, ali slikovne oziroma zvokovne materiale. Tudi če za podatke v multimedijški obliki uporabimo tehnike zgoščevanja podatkov, še vedno zasedajo veliko količino pomnilniškega prostora (Compression FAQ, 2002). Shranjevanje take količine podatkov na enem samem mestu postaja čedalje večji problem zaradi omejene količine pomnilniškega prostora v računalnikih.

Podjetja razširjajo področja svojega delovanja s fizično geografsko razpršenostjo, kjer so lahko posamezne enote celo na drugih celinah. Pojavi se potreba po deljenih informacijskih sistemih in posledično po porazdelitvi (decentralizaciji) podatkovnih baz. To privede do novih težav. Omrežni segmenti, na katerih se nahaja informacijska oprema, ki jo uporabljamo za shranjevanje podatkov, so lahko oddaljeni več tisoč kilometrov in povezani s počasnimi komunikacijskimi povezavami. Prenos podatkov po komunikacijskih povezavah ne obsega samo medsebojne komunikacije porazdeljenih podatkovnih baz, ampak jih uporabljajo tudi druge aplikacije in programske rešitve. Tako pogosto prihaja do zasičenja komunikacijskih povezav (Comer, 1991, str. 309). Zaradi napake na omrežju ponudnika ali izpada vmesnika na omrežni napravi lahko pride do pretrganja komunikacij. Zaradi teh težav se postavlja več vprašanj in možnih rešitev, kako uporabnikom zagotoviti nemoten dostop do podatkov in s tem izvajanje poslovnih procesov. Rešitev je več: od enostavnega povečanja zmogljivosti komunikacijskih povezav, vpeljave dodatnih neodvisnih komunikacijskih povezav, ki preidejo v

uporabo, ko se primarna povezava pretrga ali zasiti, do uporabe prioritizacije prometa v omrežju (Vegesna, 2001, str. 6).

V podjetjih narašča zahteva po čim hitrejšem dostopu do podatkov. Vsako povpraševanje po podatkih, shranjenih v podatkovni bazi, zahteva tudi določen procesorski čas, odvisno od zahtevnosti same dejavnosti. Več ko imamo shranjenih podatkov v podatkovni bazi, počasnejši in zahtevnejši je dostop do njih (Sloman, 1994, str. 63).

Pri porazdelitvi podatkov na več strežnikov dosežemo, da so hitreje dosegljivi. Vsak strežnik dobi le del zahtev, ki bi jih sicer dobil en sam strežnik. Pri tem ni nujno, da so strežniki na enem samem lokalno povezanem omrežnem segmentu. Z razpršitvijo strežnikov po oddaljenih lokacijah zmanjšamo zakasnitev pri dostopu do podatkov. Lokalni uporabniki dostopajo do podatkov na strežnikih, ki so jim po lokaciji ali hitrosti omrežnih povezav najugodnejši, in zato najhitreje odgovarjajo na njihove zahteve. Seveda se pri porazdeljenih podatkovnih bazah srečamo še z večjimi problemi, kot jih sicer srečamo v primerih, ko imamo samo centralni strežnik s centralno bazo podatkov (Martin, 1981, str. 322-347):

- isti podatek se lahko hkrati spremeni na različnih lokacijah;
- če se podatek prebere prej, kot je bila upoštevana njegova sprememba, se uporabi napačna vrednost;
- zaporedje zaseganja podatkov na globalni ravni lahko privede do mrtve zanke (angl. deadlock);
- zaradi izpada omrežja se spremembe na podatkih ne morejo izmenjati med vsemi porazdeljenimi podatkovnimi bazami;
- zakasnitve v omrežju upočasnijo hitrost izvajanja transakcij.

S porazdelitvijo strežnikov po oddaljenih lokacijah se pojavita vprašanji, kako izvesti medsebojno komunikacijo teh strežnikov in katere protokole uporabiti, da kar najmanj obremenimo omrežje, kljub temu pa dosežemo optimalno komunikacijo med posameznimi strežniki. Pri tem sta pomembni predvsem organizacija in konfiguracija omrežja (Williamson, 2000, str. 34-53), saj vsa omrežja ne podpirajo vseh mogočih načinov komunikacije in niso prilagojena za uporabo določenih aplikacij. Seveda moramo pri tem biti pozorni tudi na same porazdeljene podatkovne baze, ki morajo upoštevati prednosti in omejitve, ki jih postavljajo način komuniciranja na omrežni ravni in uporabljeni protokoli.

1.2 Namen in cilji magistrskega dela

Cilji magistrskega dela so raziskati načine za porazdelitev podatkovnih baz tako na aplikacijskem kot tudi omrežnem sloju, predstaviti smernice za učinkovito implementacijo porazdeljenih baz v produkcijskem okolju ter opisati mehanizme za zagotavljanje celovitih in pravih podatkov v poslovnem okolju za odločanje in operativno delo (izvajanje poslovnih procesov).

V prvem delu bom opisal tehnologije, ki omogočajo porazdelitev na določnem sloju ISO/OSI omrežnega modela. Predstavil bom zahteve aplikacij in protokolov za zagotavljanje celovitosti in varnosti informacije, prenesene po komunikacijskih povezavah. Opisal bom tudi protokole, ki omogočajo najučinkovitejši prenos podatkov po komunikacijskih povezavah, saj to področje pri nas še ni dovolj opisano.

V drugem delu, v katerem se poleg literature opiram na lastno delo in izkušnje na področju večsmerne komunikacije ter porazdeljenih podatkovnih baz, bom predstavil najpogostejše probleme, ki jih v poslovnem svetu srečamo pri porazdelitvi podatkovnih baz, in podal smernice za učinkovite rešitve.

1.3 Metode dela

V magistrskem delu izhajam iz teoretičnih spoznanj, pridobljenih s študijem problematike podatkovnih baz in podatkovnih omrežij. Analiziral bom razlike in učinkovitost obstoječih protokolov ter podal smernice za njihovo izboljšanje, kjer je to mogoče. Pri tem bom uporabil bogate izkušnje, pridobljene z dolgoletnim praktičnim delom z omrežno opremo ter podatkovnimi bazami in večletnim predavanjem o aktualni tematiki na področju večsmerne oddajanja (angl. multicast) tako v Sloveniji kot tudi po svetu.

Zaradi relativno novega področja, ki ga odpira problematika porazdeljenih podatkovnih baz, se bom pri tem opiral na najnovejša spoznanja, obstoječe raziskave, logično sklepanje in lastne izkušnje na tem področju.

1.4 Zasnova magistrskega dela

Magistrsko delo je razdeljeno na šest poglavij. Uvodu sledi poglavje o pomenu podatkov ter njihovi povezavi z informacijami in znanjem. Opisana sta površinski podatkovni model in arhitektura porazdeljenih podatkovnih baz. Pri tem so

nakazani problematika in načini za reševanje problema porazdelitve podatkov. Prikazana je tudi problematika načrtovanja velikih podatkovnih baz, ki jo delno srečamo že pri klasičnih podatkovnih bazah.

V tretjem poglavju so opisani najpogostejši problemi pri dostopu do podatkov, ki jih srečamo pri porazdeljenih podatkovnih bazah. Podane in razčlenjene so rešitve teh problemov, ki zagotavljajo celovitost podatkov. Na koncu sta opisani še dve metodi za zagotavljanje podatkov ob napakah, ki lahko nastanejo med izvajanjem transakcij na porazdeljenih podatkovnih bazah.

V četrtem poglavju so predstavljeni računalniška omrežja in protokoli, ki omogočajo komunikacijo med oddaljenimi računalniki. Slojni omrežni model omogoča neodvisno izdelavo strojne in programske opreme na določenem sloju z natančno definiranimi vmesniki tako proti višjim slojem kakor tudi proti nižjim slojem. Opisana sta dva omrežna modela in njune značilnosti.

V petem poglavju je natančno opisana posebna vrsta komunikacijskega protokola, ki omogoča hkratno komunikacijo enega z več sprejemniki – večsmerno oddajanje. Tak način komunikacije zagotavlja optimalno izkoriščanje pasovne širine komunikacijskih povezav in najmanjšo mogočo obremenitev strežnikov. Opisani so omrežni protokoli, ki so nujno potrebni, da je tak način komunikacije sploh mogoč. Podane so tudi možnosti za učinkovito zagotovitev varnosti in zanesljivosti komunikacije, ki je v javnih omrežjih vedno nujna.

V šestem poglavju je predstavljena ideja o združitvi večsmernega oddajanja s porazdeljenimi podatkovnimi bazami. Analizirane so smernice za učinkovite rešitve tako za podatkovne baze brez podvajanja podatkov kakor tudi za podatkovne baze, ki uporabljajo podvajanje za povečanje razpoložljivosti podatkov. Pri tem so podani primeri učinkovite uporabe različnih rešitev.

Magistrsko delo končujem s pomembnejšimi ugotovitvami, do katerih sem prišel med preučevanjem literature, pri praktičnem delu ter uporabi opisanih metod in protokolov.

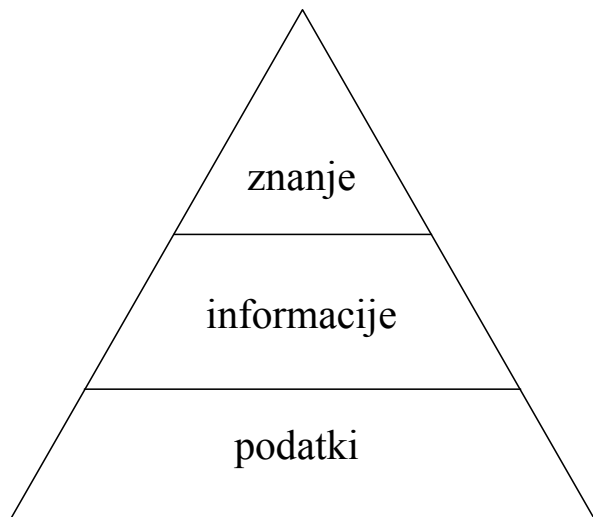
2 PORAZDELJENE PODATKOVNE BAZE

Podjetja potrebujejo za svoje poslovanje čedalje več podatkov. Zaradi lažjega dostopa uporabnikov do podatkov so ti shranjeni v različnih oblikah. To so lahko kartotečni sistemi, dokumentirani postopki in pravila, poslovne knjige, zapisana dejstva, različne podatkovne baze itd.

Podatkovna baza je računalniško podprt model okolja, ki je namenjen izvajanju akcij in lažjemu sprejemanju odločitev. S svojo organizacijo omogoča hitrejši in preglednejši dostop do podatkov, kakor če bi ti bili shranjeni kako drugače (Mohorič, 1992, str. 9).

Povezava med podatki in informacijami je včasih zabrisana in se termina pogosto uporabljata kot sinonima, vendar med njima obstaja bistvena razlika. Podatek je le dejstvo, koncept ali instrukcija, ki je primeren za obdelavo ali interpretacijo. Podatek postane informacija šele, ko ga ovrednotimo ali mu pripišemo neki pomen. Iz množice primernih informacij lahko razberemo neko uporabno znanje, ki ga potem uporabimo v delovnem procesu. Povezavo med podatkom, informacijo in znanjem lahko prikažemo v obliki piramide (slika 1).

Slika 1: Povezava med znanjem, informacijami in podatki.



Vir: Bezjak, 2002, str. 14.

2.1 Podatkovna baza

Za nastanek in razvoj podatkovnih baz je pomembno poznati zgodovino njihovega nastanka. Na začetku razvoja računalništva je bilo vse podrejeno računalniku. Uporabniki so se morali prilagajati njegovim zmožnostim. Sestava podatkov je bila prilagojena za lažjo uporabo na računalnikih. Programska oprema je bila tesno integrirana s podatki, včasih celo do te mere, da so bili podatki neposredno zakodirani v programu. Sprememba podatkov je zato večkrat zahtevala tudi spremembo programa.

Po podatkovni revoluciji so v ospredje prišli podatki. Glavni razlog za podatkovno revolucijo je bil razvoj pomnilniških medijev. S čedalje nižjimi cenami in hkrati s čedalje večjo zmogljivostjo magnetnih diskov se je spremenila tehnika programiranja. Zmanjševati se je začel semantični prepad med programskimi jeziki in človeškim mišljenjem, programi pa so se začeli prilagajati človeku in postajajo čedalje bolj neodvisni od računalnika. Ko podatki stopijo v ospredje, postane podatkovna baza osnova za nadzor in vodenje poslovnih procesov, za sprejemanje odločitev in za podporo dejavnosti podjetij (Mohorič, 1992a, str. 6-7).

Med programsko opremo in podatke stopi sistem za upravljanje s podatkovno bazo, ki mora zagotoviti razpoložljivost podatkov in nadzor nad njihovo uporabo tako uporabnikov kot tudi programske opreme. Pri tem sistem za upravljanje s podatkovno bazo skrbi za:

- celovitost in trajnost podatkov,
- učinkovit dostop do podatkov v okviru določenih pravic ter
- omogoča sočasnost uporabniških in sistemskih dostopov.

2.2 Površinski podatkovni modeli

Podatkovni model je mehanizem, s katerim opišemo zgradbo podatkovne baze. Danes uporabljamo štiri vrste podatkovnih modelov (Database Models, 2003):

1. hierarhične, ki temeljijo na drevesnih strukturah,
2. mrežne, ki so podobni hierarhičnim, vendar podpirajo več staršev in temeljijo na mrežnih strukturah,
3. relacijske, ki temeljijo na relacijah in
4. objektne, ki so se razvili s konceptom objektnega programiranja.

Hierarhični in mrežni podatkovni model sta vezana na fizično organizacijo podatkovne baze. To je razumljivo, saj sta nastala še v času, ko je bil dostop do podatkov omejen na zaporedno branje magnetnih trakov. Ker je struktura zapisa v bazi predvidljiva, je dostop do podatkov zelo enostaven. Zapisi so organizirani v hierarhijo starš-otrok (angl. parent-child), kjer mrežni podatkovni model dovoljuje tudi dva ali več staršev v razmerju starš-otrok. Ker moramo za vsako strukturo napisati program za dostop do nje, je programiranje ob pogostih spremembah strukture podatkov v hierarhičnih podatkovnih bazah precej zamudno.

Relacijski model izrablja prednost neposrednega dostopa do podatkov, kjer dostopni čas do podatka na mediju ni več bistveno odvisen od tega, kje na mediju se podatek fizično nahaja (Kodek, 1988, str. 178). Podatki in relacije med njimi so urejeni v tabele. Vsaka vrstica v tabeli predstavlja neki zapis. Vsebina tabel je

uporabnikom dostopna s pomočjo povpraševalnih jezikov, ki temeljijo na relacijski algebri ali relacijskem računu.

Objektni podatkovni model je na neki način razširitev mrežnega. Podatki se shranjujejo v obliki objektov. Vsak objekt ima lahko svoje attribute, ki ga opisujejo, kakor tudi metode, ki dodajajo objektu določeno obnašanje (Cattell, 1991, str. 37-53).

Prednost relacijskega podatkovnega modela je v njegovi neodvisnosti od strukture podatkov. Podatki so dostopni prek tabel, kjer ima vsaka vrstica enak format zapisa. S predstavitvijo podatkov s tabelami se do njih lažje in hitreje dostopa ter jih je lažje spreminjati (Maier, 1983). Zaradi razširjenosti, hitrosti in prilagodljivosti bom v nadaljevanju dela pri raziskovanju uporabljal predvsem relacijski podatkovni model.

2.3 Arhitektura porazdeljenih podatkovnih baz

Vsaka podatkovna baza je sestavljena iz (Welzer-Družovec, 2002, str. 142-146):

1. podatkov,
2. sistema za upravljanje podatkovne baze,
3. uporabnikov in uporabniških vmesnikov ter
4. administratorjev podatkovne baze.

Podatkovni del podatkovne baze je sestavljen iz fizičnega dela, kjer so shranjene vrednosti podatkovnih elementov, in metapodatkovnega dela, kjer so shranjeni opisi fizičnih podatkov, kaj pomenijo in kako so dostopni uporabnikom.

Metapodatkovna baza vsebuje tri vrste opisov fizičnih podatkov:

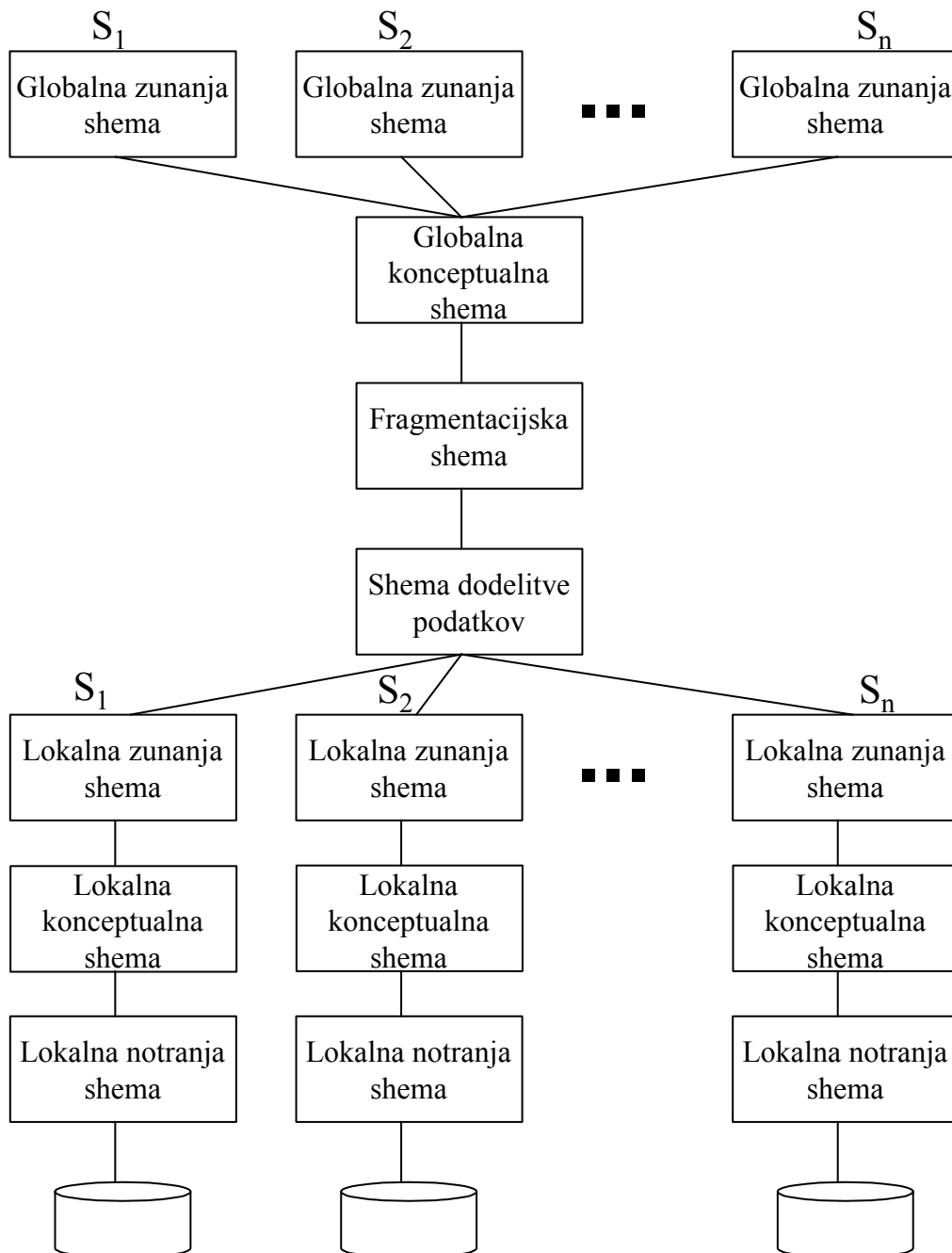
1. notranjo shemo,
2. konceptualno shemo in
3. zunanje sheme.

Notranja shema omogoča preslikavo fizičnih datotek v logične zapise. Obsega podatke, ki definirajo, kje se nahajajo podatkovni elementi, ki tvorijo logični zapis. Konceptualna shema obsega opise pravil, ki jih določene vrednosti podatkov lahko zavzamejo. Obsega tudi opise lastnosti in povezave entitet iz modela okolja, ki ga predstavlja. Na zunanji ravni se s pomočjo opisov kaže kot uporabnikov model okolja.

Porazdeljene podatkovne baze na našete vrste opisov dodajo še (slika 2):

1. globalne zunanje sheme,
2. globalno konceptualno shemo,
3. fragmentacijsko shemo in
4. shemo dodelitve podatkov.

Slika 2: Predstavitev metapodatkovne baze za porazdeljene podatkovne baze.



Vir: Welzer-Družovec, 2002, str. 147.

Globalne zunanje sheme predstavljajo poglede različnih uporabnikov. Globalna konceptualna shema omogoča fizično neodvisnost podatkov od porazdeljenega okolja. Fragmentacijska shema in shema dodelitve podatkov določata, kako so podatki logično porazdeljeni in kje se nahajajo fizični podatki (Welzer-Družovec, 2002a, str. 147).

2.3.1 Sistem za upravljanje porazdeljenih podatkovnih baz

Z vidika upravljanja je podatkovna baza organizirana centralno ali porazdeljeno. O centralizirani organizaciji govorimo, ko podatkovno bazo podpira sistem, ki se nahaja na istem računalniškem sistemu kot celotna podatkovna baza. Pri porazdeljeni podatkovni bazi so podatki shranjeni na fizično ločenih računalniških sistemih, funkcionalno pa deluje kot celovita centralizirana podatkovna baza. Porazdeljena podatkovna baza hrani logično povezane podatkovne baze na dveh ali več različnih računalnikih, ki so med seboj povezani s komunikacijskimi povezavami. Lokalne sisteme za upravljanje s podatkovno bazo združuje in kontrolira globalni sistem.

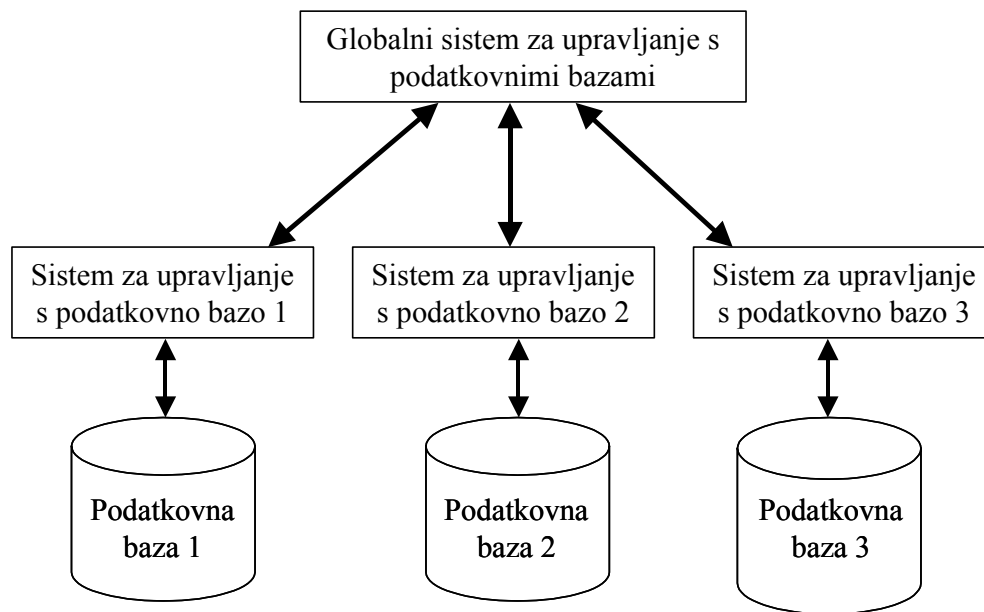
Porazdeljene podatkovne baze delimo na dve različni skupini:

1. homogene in
2. heterogene.

Pri homogenih porazdeljenih podatkovnih bazah uporabljamo enake lokalne sisteme za upravljanje s podatkovnimi bazami in enake podatkovne baze. Podatke porazdelimo po enakih podatkovnih bazah, ki se nahajajo na več različnih strežnikih v omrežju. Vsi uporabniki dostopajo do podatkovne baze prek globalnega sistema za upravljanje s podatkovnimi bazami (slika 3). Globalni sistem za upravljanje s podatkovnimi bazami uporabnikom transparentno zagotovi podatke s tistega strežnika, na katerem se nahaja podatkovna baza z zahtevanimi podatki.

Pri heterogenih porazdeljenih podatkovnih bazah gre za povezavo različnih sistemov za upravljanje s podatkovnimi bazami in s tem tudi podatkovnih baz. Za integracijo različnih sistemov prav tako poskrbi globalni sistem za upravljanje. Vendar mora v tem primeru globalni sistem za upravljanje poleg informacije, kje se kakšen podatek lahko nahaja, upoštevati še dejstvo, da imamo lahko združene različne tipe sistemov za upravljanje s podatkovnimi bazami, ki imajo drugačne predstavitve podatkov.

Slika 3: Homogene porazdeljene podatkovne baze.



Vir: Martin, 1981.

V nadaljevanju dela se bom posvetil problematiki homogenih porazdeljenih podatkovnih baz. Problemi, ki se pojavljajo pri slednjih, so aktualni tudi pri heterogenih, vendar je pri heterogenih poudarek predvsem na integraciji različnih sistemov za upravljanje s podatkovnimi bazami, ki se nahajajo na različnih operacijskih sistemih.

2.4 Porazdelitev podatkov

Z naraščanjem količine podatkov se pojavlja vprašanje, kje in kako jih hraniti, da bodo kadar koli dostopni za vse uporabnike. Ko implementacija podatkovne baze zahteva porazdeljen sistem, se postavi vprašanje, na kakšen način porazdeliti podatke po posameznih podatkovnih bazah.

Poznamo tri načine za postavitev podatkov:

- centraliziran,
- repliciran in
- razdeljen.

Pri vseh pristopih gre za kompromise med naslednjimi spremenljivkami (Bray, 1983):

- cena pomnilniškega prostora,
- velikost pomnilniškega prostora na posameznem sistemu,
- stroški komunikacijskih povezav,
- hitrost dostopa do podatkov in
- zanesljivost dostopa do podatkov.

Če želimo zmanjšati ceno, ki jo plačamo za pomnilniški prostor, hranimo v bazi samo eno kopijo podatka. Podvajanje podatkov med strežniki podatkovnih baz prinaša zahteve po dodatnem pomnilniškem prostoru, kar zviša ceno rešitve. Vendar podvajanje podatkov ni nujno slaba rešitev. Omogoča večjo zanesljivost in dostopnost sistema v primeru izpada enega ali več strežnikov.

Ko uporabniki dostopajo do podatkovne baze na strežniku, ki se nahaja na istem omrežnem segmentu, je dostopni čas krajši, kot če morajo uporabniki dostopati do strežnikov na oddaljenih omrežnih segmentih. Z oddaljenostjo strežnikov od uporabnikov in v odvisnosti od hitrosti komunikacijskih povezav med njimi se podaljšuje dostopni čas. Zato je smiselno, da postavimo strežnike na isti omrežni segment ali na segment, ki je zelo blizu uporabnikom. V primeru izpada omrežnega segmenta, na katerem se nahajajo strežniki, je uporabnikom onemogočen dostop do podatkov, saj so pretrgane komunikacijske povezave. Da se takim situacijam izognemo, strežnike ločimo in jih postavimo na neodvisne omrežne segmente. S tem zagotovimo redundanco komunikacijskih povezav. V primeru izpada enega omrežnega segmenta lahko uporabniki še vedno dostopajo vsaj do nekaterih podatkov. Postavitve strežnikov na več ločenih segmentov omogoči tudi krajši dostopni čas večjemu številu uporabnikov.

2.4.1 Centralna postavitvev podatkov

Pri centraliziranem pristopu so vsi podatki podatkovne baze shranjeni na centralnem strežniku. Včasih je bil to edini mogoč način postavitve podatkov. Danes je ta postavitve primerna samo za manjše podatkovne baze, kjer se uporabniki nahajajo na istem omrežnem segmentu, ali pa je segment s strežnikom dosegljiv prek hitrih komunikacijskih povezav.

Centraliziran sistem je najenostavnejši za upravljanje in ne potrebuje sistema za sinhronizacijo podatkov, vendar so ob izpadu sistema ali omrežnih povezav vsi podatki nedosegljivi. Edino podvajanje podatkov dosežemo z izdelavo varnostnih kopij, ki ponavadi niso neposredno in hitro dostopne uporabnikom baze. Dostop do centralne lokacije podatkov je za oddaljene uporabnike počasnejši kot za tiste,

ki se nahajajo na lokalnem omrežnem segmentu ali so z njim povezani s hitrimi komunikacijskimi povezavami.

2.4.2 Replikacija podatkov

Replikacija je proces kopiranja in vzdrževanja podatkovnih objektov, kot so zapisi in tabele, med podatkovnimi bazami, ki sestavljajo porazdeljen sistem. Razloga za uporabo replikacije podatkov med strežniki sta večja dostopnost podatkovnih baz (če izpade en ali več strežnikov) in hitrejši dostop do podatkov (več strežnikov lahko opravi več transakcij na časovno enoto). Replicirane podatkovne baze so ponavadi razdeljene na lokacijsko neodvisne računalniške sisteme (Martin, 1981a, str. 245-251).

Računalniški sistemi, ki uporabljajo replikacijo, se lahko nahajajo na istem ali ločenih omrežnih segmentih. S tem ko jih postavimo na isti omrežni segment, dosežemo predvsem hitrejši dostop do podatkov. Tako ustvarjena množica sistemov omogoča izvajanje večjega števila transakcij. Če so uporabniki podatkov locirani na več segmentih, je smiselno postaviti podatke v njihovo neposredno bližino. Na omrežne segmente, kjer zaradi količine uporabnikov pričakujemo pogoste dostope do podatkov, postavimo strežnik (ali več) in podatke repliciramo med oddaljenimi segmenti. Tako dosežemo višjo hitrost dostopa do podatkov in redundanco strežnikov na ravni omrežja. Vsaka replicirana baza dovoljuje dostop do vseh podatkov. S pomočjo mehanizma replikacije pa si prizadevamo, da vsebujejo vse porazdeljene podatkovne baze vedno enake podatke.

Največja prednost replikacije je dosegljivost podatkov v primeru izpada enega izmed strežnikov. Uporabniki so v tem primeru enostavno in hitro preusmerjeni na enega (ponavadi najbližjega) izmed drugih strežnikov in izpada ne opazijo. Edina občutna sprememba zanje je lahko daljši dostopni čas. Druga velika prednost repliciranih podatkovnih baz je lokalnost dostopov do podatkov. Če so podatki postavljeni na strežnike na lokalnih omrežnih segmentih, je dostopni čas bistveno krajši v primerjavi s povprečnim dostopnim časom od centralno lociranih podatkov. Ker uporabniki dostopajo do lokalno postavljenih podatkov, se tudi omrežje razbremeni in omogoči boljše odzivne čase drugim aplikacijam.

Pomanjkljivost replikacije je podvajanje podatkov. Vse replicirane podatkovne baze morajo vsebovati določene ali celo vse podatke, ki se pojavljajo v kateri koli izmed njih. Pomanjkljivosti replikacije je tudi kompleksnost upravljanja in medsebojne komunikacije med repliciranimi podatkovnimi bazami.

Delno replicirane podatkovne baze vsebujejo le podmnožico vseh podatkov, ki jih porazdeljene baze hranijo. S tem se zmanjša količina potrebnega pomnilniškega prostora. Rešitev je kompromis med ceno popolne replikacije (zasedanje pomnilniškega prostora) in dostopnostjo podatkov ob izpadu omrežja (določeni podatki so nedosegljivi).

2.4.3 Razdelitev podatkov

Če so podatki izrazito lokalne narave (ko se podmnožica podatkov večinoma uporablja na eni lokaciji), lahko podatkovne baze organiziramo tako, da podatke lokalne narave porazdelimo po tistih lokalnih podatkovnih bazah, kjer pričakujemo njihovo najpogostejšo uporabo. Posamezne razdeljene baze hranijo le podmnožico vseh podatkov, pri tem pa so podatki razdeljenih baz še vedno dosegljivi vsem uporabnikom. Vse razdeljene baze skupaj tvorijo celotno podatkovno bazo.

Razdeljene baze združujejo prednosti centralno lociranih podatkovnih baz in repliciranih podatkovnih baz. Razdelitev podatkov zmanjšuje stroške hranjenja podatkov, ker lahko vedno hranimo samo eno kopijo podatkov in ker jih lahko shranimo na nosilcih z bistveno manjšo zmogljivostjo, kot če bi jih hranili centralno (Inmon, 1981, str. 95-101).

Z razdelitvijo podatkov po različnih strežnikih povečamo dostopnost podatkov, saj izpad ene razdeljene podatkovne baze ne vpliva na druge, katerih podatki so še vedno dostopni. Seveda so podatki baze, pri kateri je prišlo do izpada, nedosegljivi. Prizadeti so predvsem lokalni uporabniki podatkovne baze, ki je hranila večino podatkov, do katerih so pogosto dostopali. Če pride do izpada komunikacijskih povezav, lokalni uporabniki to občutijo le kot nedosegljivost nekaterih manj pogosto rabljenih podatkov. Ker uporabniki najpogosteje dostopajo do podatkov lokalne podatkovne baze, so dostopni časi kratki in uporaba omrežnih povezav je manjša.

Problem, ki ga srečamo pri razdelitvi podatkov glede na lokalnost dostopov, je, kako učinkovito določiti, kateri podatki so lokalne narave. To je zelo pomembno za zmanjšanje števila dostopov do podatkov na oddaljenih podatkovnih bazah. Včasih uporabljamo rešitev, kjer podatki niso strogo ločeni po razdeljenih podatkovnih bazah, ampak se pojavljajo na več lokacijah. V tem primeru moramo zagotoviti ažurnost podatkov z replikacijo ali kakšnim drugim mehanizmom.

Baze, kjer je ločitev podatkov na več različnih lokacij nujna, so velike podatkovne baze. Te obsegajo toliko podatkov, da jih ne moremo shraniti na enem samem strežniku, ne da bi bila pri tem močno degradirana hitrost dostopa do njih.

2.5 Velike podatkovne baze

Količina podatkov, shranjena v podatkovni bazi, je praktično omejena samo s fizičnim prostorom, ki ji ga odmerimo na strežniku. Če nam ga začne primanjkovati, dodamo dodatne pomnilniške zmogljivosti. Če strežnik fizično ne zmore več zagotoviti zmogljivosti, potrebne za podatkovno bazo, ali ko je cena ustreznega strežnika previsoka, lahko v sistem dodamo nov strežnik. Sistema za upravljanje s podatkovno bazo se ustrezno povežeta z globalnim sistemom za upravljanje in del podatkov, ki se nahajajo na obstoječem strežniku, prerazporedimo na novega. Porazdelitev podatkov po strežnikih lahko opravimo glede na ključ (angl. indeks) ali lastnosti podatkov.

2.5.1 Porazdelitev podatkov glede na ključ

Pri porazdelitvi podatkov ene podatkovne baze na več podatkovnih baz lahko določimo količino podatkov in vsebino posameznih podatkovnih baz glede na ključ. Tako lahko bazo, ki vsebuje zapise s ključi od 1 do 100, porazdelimo na dve bazi, kjer prva vsebuje podatke s ključi od 1 do 50, druga pa podatke s ključi od 51 do 100.

Prednosti porazdeljenih podatkovnih baz glede na ključ so (Inmon, 1981a, str. 97-99):

- če pride do napake na eni od porazdeljenih podatkovnih baz, lahko glede na ključ preprosto ugotovimo, kateri drugi podatki so še vedno dosegljivi,
- upravljanje s tako porazdeljenimi podatkovnimi bazami je razdeljeno na upravljanje s posameznimi podatkovnimi bazami.

Slabe strani porazdeljenih podatkovnih baz glede na ključ so:

- sinhronizacija dostopa do podatkovnih baz postane zelo zahtevna,
- odločitev, kje razdeliti podatke, da bodo posamezne porazdeljene podatkovne baze enakomerno obremenjene, ponavadi ni enostavna,
- vsaka posamezna podatkovna baza dodatno zaseda prostor s svojim sistemom za upravljanje in definicijami tabel,
- nevarnost, da se vsaka porazdeljena podatkovna baza, če je porazdelitev glede na glavni ključ vezana na določeno kategorijo (lokacija, oddelek,

področje ...), začne razvijati v svoji smeri glede na druge porazdeljene podatkovne baze.

2.5.2 Porazdelitev podatkov glede na lastnosti

Pri porazdelitvi podatkov glede na lastnosti podatke s podobnimi lastnostmi hranimo v skupni podatkovni bazi. Pri tem je lastnost podatkov lahko lokacija njihovega nastanka, ciljna skupina uporabnikov, skupina produktov itd. Ker so lastnosti včasih skupne tudi dvema ali več lokacijam, se podatki lahko znajdejo v več podatkovnih bazah hkrati.

Prednosti porazdelitve podatkov glede na lastnosti (Inmon, 1981b, str. 99-101):

- upravljanje s tako porazdeljenimi podatkovnimi bazami je razdeljeno na upravljanje s posameznimi podatkovnimi bazami in
- lažje zagotovimo, da so podatki fizično bližje uporabnikom, ki do njih pogosto dostopajo.

Slabosti porazdelitve podatkov glede na lastnosti:

- sinhronizacija dostopa do podatkovnih baz postane zelo kompleksna,
- porazdeljevanje podatkov je zahtevnejše in
- redundanca podatkov je večja v primerjavi s porazdelitvijo glede na ključ.

V obeh primerih porazdelitve podatkov je bila delitev izvedena horizontalno, torej po vrsticah tabel relacijske podatkovne baze. Mogoči delitvi podatkov sta še vertikalna in mešana. Pri vertikalni delitvi posamezne baze hranijo stolpce tabel, pri mešani pa se uporabljata obe delitvi hkrati (posamezni vertikalni segmenti so razdeljeni še horizontalno ali nasprotno).

Možnost za razdelitev podatkov je tudi lokacija njihovega nastanka. Podatkovne baze hranijo le lokalno ustvarjene podatke. Tako bistveno skrajšamo dostopni čas, saj študije kažejo, da je 90 odstotkov dostopov do podatkov lokalnih glede na lokacijo nastanka (Bell, 1992, str. 2). Razlog za tako visoko lokalnost dostopov je dejstvo, da skupine ali oddelki znotraj podjetja, ki določene podatke vnesejo ali ustvarijo, do teh podatkov potem tudi najpogosteje dostopajo, čeprav so dostopni celotni organizaciji.

3 ZAGOTAVLJANJE CELOVITOSTI PODATKOV V PODATKOVNIH BAZAH

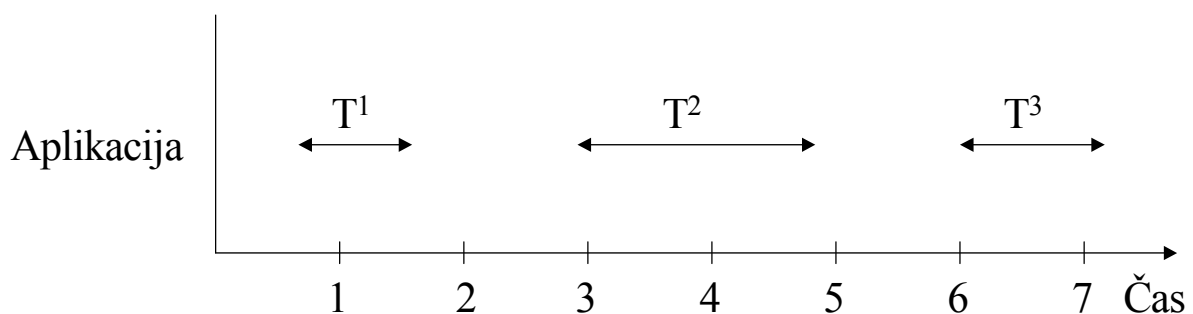
Transakcija je definirana kot množica akcij, ki pripadajo enemu uporabniku ali aplikaciji podatkovne baze, ki so nedeljiva celota. Transakcije imajo štiri lastnosti (ACID, 2003):

1. **nedeljivost:** transakcija je nedeljiva celota (izvesti se morajo vse akcije, ki jih transakcija obsega, ali nobena),
2. **veljavnost:** transakcije spreminjajo podatkovno bazo iz ene veljavne oblike v drugo, po izvedbi transakcije je podatkovna baza spet v veljavnem stanju,
3. **neodvisnost:** transakcije se izvajajo neodvisno druga od druge (delni rezultati izvajanja podmnožice akcij ene transakcije niso vidni drugim transakcijam) in
4. **trajnost:** rezultati uspešno zaključene transakcije so trajno shranjeni v podatkovni bazi.

Transakcije preoblikujejo podatkovno bazo iz enega veljavnega (celovitega) stanja v drugo veljavno stanje. Če pride do napake pri izvajanju transakcije, lahko podatkovna baza ostane v neveljavnem stanju. Sistem za upravljanje s podatkovno bazo mora vedeti, ali je podatkovna baza v neveljavnem stanju, zato mora vedeti, kdaj je transakcija zaključena. Vsaka transakcija se zato s posebnim ukazom zaključi (angl. commit). Če transakcije ni mogoče uspešno končati, aplikacija to sporoči sistemu za upravljanje s podatkovno bazo s posebnim ukazom prekini (angl. abort). V določenih primerih lahko sistem sam zazna napako in povzroči razveljavitev transakcije. V takih primerih in v primeru ukaza prekini mora sistem povrniti podatkovno bazo v predhodno veljavno stanje.

Pri transakcijah, ko se vsaka naslednja transakcija izvede popolnoma za predhodno, pravimo, da se izvajajo zaporedno (slika 4).

Slika 4: Zaporedno izvajanje transakcij.



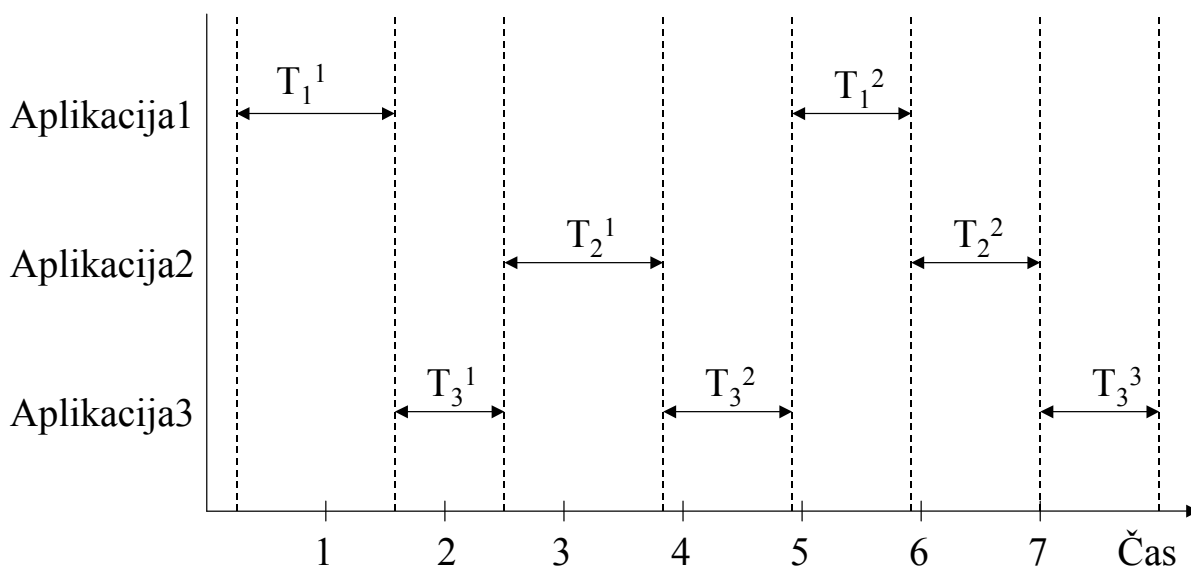
Vir: prirejeno po Bell, 1992, str. 166.

Prednost zaporednega izvajanja transakcij je, da medsebojno ne tekmujejo za podatke. Vsaki transakciji so na voljo vsi podatki in lahko od njih dostopa brez omejitev. Vendar je izključno zaporedno izvajanje transakcij v sodobnih večopravnih sistemih omejujoče do drugih transakcij in potratno glede procesorskega časa, saj so sposobni izvajati več transakcij hkrati.

Čeprav imamo v sistemu več transakcij, ki dostopajo in spreminjajo popolnoma druge podatke iste podatkovne baze, jih ne moremo izvesti vseh hkrati, ker je dovoljen dostop do podatkov samo eni transakciji. S tem ko morajo druge transakcije ali aplikacije oziroma uporabniki čakati, da se predhodna transakcija izvede, prihaja do nepotrebnih zastojev pri izvajanju transakcij.

Sočasno izvajanje transakcij poveča izkoristek procesorja in skrajša čakanje na proste podatke (slika 6). Primerjava slik 5 in 6 pokaže, da je čas, potreben za izvedbo aplikacij, v primeru sočasnega izvajanja transakcij veliko krajši kot v primeru, ko jih izvajamo zaporedno.

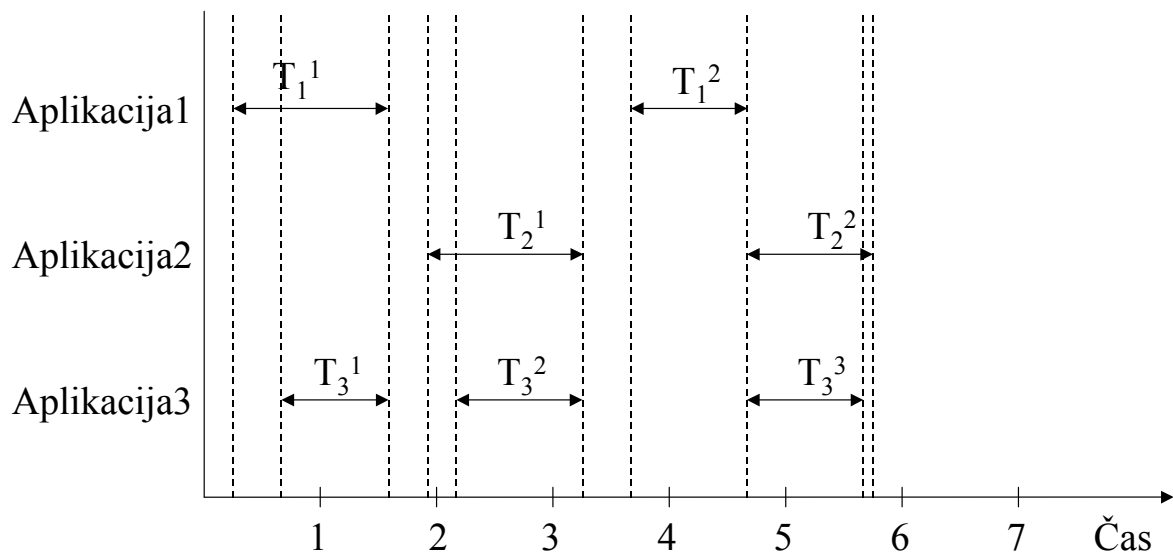
Slika 5: Zaporedno izvajanje več transakcij.



Vir: prirejeno po Bell, 1992, str. 167.

V sistemih, kjer lahko več aplikacij izvajamo hkrati, jih lahko izvedemo več v krajšem času. To je mogoče, saj je vsem transakcijam dovoljen sočasen dostop do podatkov. S tem se izognemo čakanju naslednjih transakcij, da se predhodne izvedejo v celoti. Medtem ko ena aplikacija zapisuje podatke, lahko druga izvaja kompleksne izračune ali čaka na uporabnikov vnos.

Slika 6: Sočasno izvajanje več transakcij.



Vir: prirejeno po Bell, 1992, str. 167.

Transakcijam, ki dostopajo do popolnoma različnih podatkov in sprememba le-teh ne vpliva na druge transakcije, ki se izvajajo hkrati, pravimo neodvisne transakcije. Te se lahko izvajajo sočasno z drugimi medsebojno neodvisnimi transakcijami, ne da bi pri tem spravile podatkovno bazo v neveljavno stanje. Njihovo nasprotje so odvisne transakcije. Te spreminjajo ali dostopajo do istih podatkov in lahko spravijo bazo v neveljavno stanje. Zato moramo biti zelo previdni, kdaj in na kakšen način bomo dovolili odvisnim transakcijam spremembe zapisov v podatkovni bazi.

3.1 Interferenca med sočasnimi odvisnimi transakcijami

Zaporedne odvisne in neodvisne ter sočasne neodvisne transakcije vedno pustijo podatkovno bazo v veljavnem stanju. Pri njih ne prihaja do hkratnih dostopov do istih podatkov. Sočasne odvisne transakcije, ki se izvajajo hkrati in dostopajo do istih podatkov, pa lahko ogrozijo neodvisnost podatkov in spravijo podatkovno bazo v neveljavno stanje.

Pri odvisnih sočasnih transakcijah lahko pride do več različnih težav:

1. izguba spremembe podatka,
2. kršitev nedeljivosti transakcij ali
3. napačno branje podatka.

3.1.1 Izguba spremembe podatka

Do izgube spremembe podatka pride, ko ena transakcija izvaja spremembo podatka, pri tem pa tudi druga transakcija spremeni isti podatek.

Za primer lahko vzamemo situacijo, prikazano na sliki 7.

Slika 7: Časovni potek izvajanja transakcij, pri katerih pride do izgube spremembe podatka.

Čas	Transakcija T ₁	Transakcija T ₂
1	Preberi stanje	
2	stanje = stanje – 100	Preberi stanje
3	Zapiši stanje	stanje = stanje + 100
4		Zapiši stanje

Če je bila začetna vrednost podatka stanje pred začetkom transakcij 100, bi morala biti tudi končna vrednost tega podatka 100. Transakcija T₁ je odvzela 100 enot, transakcija T₂ pa je dodala 100 enot. Vendar je prišlo do sočasnega dostopa do začetnega stanja. Obe transakciji sta prebrali vrednost 100, prva transakcija je odvzela 100 enot in zapisala vrednost 0. Druga transakcija je prebrani vrednosti 100 dodala 100 enot in zapisala vrednot 200. Ker je obveljalo zadnje pisanje, je trenutna vrednost stanja 200, kar pa je napačen podatek.

3.1.2 Sprememba pomena podatkov

Pri spremembi pomena podatkov nastane problem, ko različne transakcije spremenijo isto vrstico relacijske podatkovne baze, vendar ne istega podatka. Rezultat transakcij je izguba pomena podatkov, tako da ti niso več smiselni.

Za primer lahko vzamemo situacijo, prikazano na sliki 8.

Slika 8: Časovni potek izvajanja transakcij, pri katerih pride do spremembe pomena podatkov.

Podatkovna baza Predavatelji:

Predavatelj	Predmet
Janko Novak	Podatkovne baze
Janko Novak	Operacijski sistemi
Miha Ban	Operacijski sistemi

Podatkovna baza Predavanja:

Predavatelj	Predmet	Datum
Janko Novak	Operacijski sistemi	30.12.2003

Časovni potek transakcij:

Čas	Transakcija T ₁	Transakcija T ₂
1	Preberi Predavanja, kjer je datum='30.12.2003'	Preberi Predavanja, kjer je datum='30.12.2003'
2	Preberi Predavatelji, kjer je Predavanja.Predavatelj = Predavatelj.Predavatelj in Predmet='Podatkovne baze'	Preberi Predavatelji, kjer je Predavanja.Predmet=Predavatelj.Predmet in Predavatelj='Miha Ban'
3	Če ne obstaja, zaključi transakcijo	Če ne obstaja, zaključi transakcijo
4	Predavanja.Predmet='Podatkovne baze'	Predavanja.Predavatelj='Miha Ban'
5	Zaključi	Zaključi

Podatkovna baza Predavatelji vsebuje informacije o predavateljih in predmetih, ki jih predavajo. Podatkovna baza Predavanja vsebuje informacije o predavateljih, ki bodo predavali določen predmet na določen dan.

Zaradi zamenjave urnika predavanj transakcija T₁ poišče vsa predavanja na dan 30.12.2003, nato poišče pri vseh predavateljih, ki predavajo na ta dan (Janko Novak), ali lahko predavajo tudi predmet 'Podatkovne baze', in zamenja predavani predmet v 'Podatkovne baze'.

Transakcija T₂, ki se izvaja hkrati, poišče vsa predavanja na dan 30.12.2003, nato poišče pri vseh predmetih, ki se predavajo na ta dan (Operacijski sistemi), ali jih lahko predava predavatelj 'Miha Ban', in zamenja predavatelja v 'Miha Ban'.

Če bi se vsaka od teh transakcij izvedla sama zase, potem bi obe uspešno spremenili bazo in jo pustili v celovitem stanju. Ker pa se transakciji izvedeta hkrati, je rezultat necelovito stanje, saj naj bi 30.12.2003 predaval 'Miha Ban' predmet Podatkovne baze, čeprav 'Miha Ban' lahko predava le predmet Operacijski sistemi.

3.1.3 Napačno branje podatka

Ponavadi smo pri sočasnih transakcijah pozorni na spreminjanje podatkov, vendar lahko pride do težav tudi, če transakcija samo bere podatke. Situaciji, kjer transakcija prebere vrednost podatka, ki se kasneje popravi, pravimo napačno branje podatka (angl. dirty read).

Kot primer vzemimo transakcijo T_1 , ki bere podatke o gibanju stanja zalog. Medtem ko transakcija bere spremembe zalog, lahko druge transakcije stanje zalog posameznih postavk spreminjajo. Če transakcija T_1 ne bo upoštevala vseh vmesnih sprememb v bazi, bo prišla do napačnega končnega podatka o stanju zalog.

Vsekakor se težavam, do katerih prihaja pri sočasnem odstopu do podatkov, lahko izognemo. Obstaja več metod za zagotavljanje podatkov v podatkovnih bazah (Holliday, 1999, str. 1):

1. zaseganje podatkov (angl. locking),
2. časovno označevanje (angl. time stamping),
3. optimistične metode (angl. optimistic methods) in
4. metode večine (angl. quorum consensus).

3.2 Zaseganje podatkov

Zaseganje podatkov je najbolj razširjena metoda za zagotavljanje podatkov v sistemih za upravljanje s podatkovnimi bazami. Ko neka transakcija želi dostop do podatkov, mora predhodno opraviti zaseganje podatkov za branje ali pisanje (odvisno od akcij, ki jih želi izvesti). Ko določena transakcija zaseže podatke za branje, lahko tudi druge transakcije dostopajo do teh podatkov in jih zasežejo za branje. Ker bralni dostop podatkov ne spremeni, transakcije, ki berejo isti podatek, ne vplivajo druga na drugo. Drugače pa ravnamo v primeru zaseganja podatkov za pisanje. Ko določena transakcija zaseže podatke za pisanje, nobena druga transakcija ne more dostopiti do tako zaseženih podatkov. Pri tem je onemogočen dostop tako za branje kot pisanje, dokler jih transakcija, ki je zasegla podatke za pisanje, ne sprostí.

Nekateri sistemi dovoljujejo nadgraditev zaseženih podatkov za branje na zaseganje podatkov za pisanje. To je mogoče le, ko je transakcija edina, ki je zahtevala zaseganje podatkov za branje. Prav tako je mogoče tudi nasprotno: zaseganje podatkov za pisanje je mogoče vedno spremeniti na zaseganje podatkov samo za branje.

Glede na transakcije, ki jih aplikacija izvaja, lahko zasežemo logični objekt:

1. zapis,
2. množica zapisov,
3. tabela.

ali fizični objekt:

1. fizični blok ali stran,
2. fizična datoteka ali
3. celotna podatkovna baza.

Ko transakcija zaseže celotno podatkovno bazo, druge transakcije ne morejo spreminjati podatkov v njej, dokler je transakcija ne sprost. Na prvi pogled se zdi, da je zaseganje samo posameznih zapisov v podatkovni bazi veliko boljši način. Vendar lahko pride do težav, ko transakcija obdeluje veliko zapisov ali skoraj vse zapise podatkovne baze. Čas, ki ga transakcija potrebuje, da zaseže celotno bazo, je bistveno krajši kot čas, ki bi bil porabljen za zaseganje vseh potrebnih posameznih zapisov znotraj baze. Dodaten čas se porabi tudi za čakanje na druge transakcije, ki morajo sprostiti zasežene podatke. V tem primeru je bolje, da transakcija zaseže celotno bazo in opravi svoje delo, kakor da poskuša zaseči vsak potreben zapis posebej. Zato mora sistem za upravljanje s podatkovno bazo dovoljevati različne velikostne stopnje zaseganja podatkov.

3.2.1 Zaseganje podatkov v dveh fazah

Eden izmed najbolj znanih in najpogosteje uporabljenih mehanizmov za zagotavljanje podatkov je zaseganje podatkov v dveh fazah. Ime izvira iz dveh faz, ki jih pri svojem izvajanju uporablja: **fazo naraščanja** in **fazo krčenja**. V fazi naraščanja transakcija poskuša zagotoviti podatke z zaseganjem, v fazi krčenja pa transakcija sprošča zasežene podatke. Pri tem mora transakcija upoštevati naslednja pravila:

1. transakcija mora najprej podatek zaseči, šele nato lahko nad podatkom izvede operacijo,
2. pri zaseganju transakcija ne sme zaseči podatka, ki je že zasežen za pisanje, ali zaseči podatka za pisanje, če je ta že zasežen za branje,
3. ko transakcija sprost kateri koli zaseženi podatek, ne sme opraviti nobenega zaseganja več,
4. ko se transakcija zaključi, morajo biti vsi zaseženi podatki sproščeni hkrati.

Zaseganje podatkov v dveh fazah zagotavlja pravilno zaporedje izvajanja transakcij s sinhronizacijo dostopa do podatkov. S tem omogoča sistemu za

upravljanje s podatkovno bazo manj kompleksno, vendar še vedno učinkovito kontrolo nad izvajanjem sočasnih transakcij.

Pri porazdeljenih podatkovnih bazah morajo posamezni sistemi za upravljanje izmenjati informacijo o podatkih, ki jih transakcija želi zaseči. Če je podatek že zasežen v kateri izmed porazdeljenih baz, potem transakcija ne sme dobiti dovoljenja za zaseganje. Pri tem se s številom podatkovnih baz v porazdeljenem sistemu povečuje število sporočil, ki jih je treba izmenjati. Za vsako zaseganje podatka v bazi tako potrebujemo $4*N$ sporočil, kjer je N število porazdeljenih podatkovnih baz, in sicer:

1. N sporočil koordinatorja na začetku transakcije vsakemu porazdeljenemu sistemu,
2. N sporočil iz porazdeljenih sistemov o tem, da se je transakcija izvedla (s tem sporočajo pripravljenost za zaključek transakcije),
3. N sporočil koordinatorja za zaključek transakcije vsakemu porazdeljenemu sistemu in
4. N sporočil iz porazdeljenih sistemov, da se je transakcija uspešno zaključila.

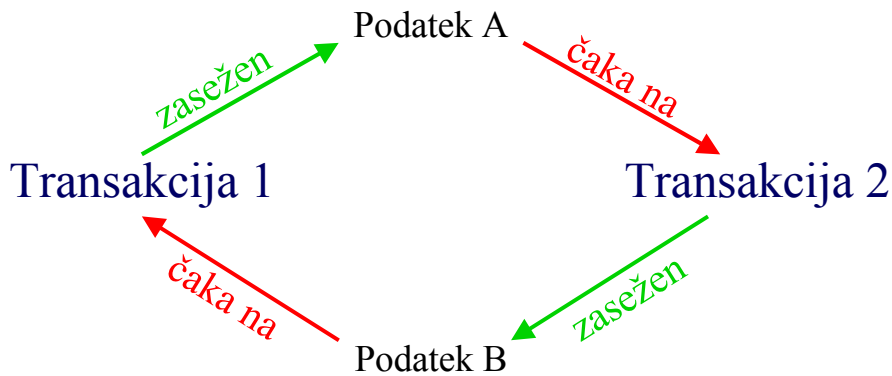
Če se upošteva še sporočilo za potrditev prejema vsakega posameznega sporočila, naraste število izmenjanih sporočil za dvakrat. Hitro lahko sklenemo, da tak način komunikacije med podatkovnimi bazami porazdeljenega sistema ne izrablja komunikacijskih povezav najučinkoviteje. Za zmanjšanje količine sporočil potrebujemo način komunikacije, kjer bo koordinator z enim samim sporočilom dosegel vse porazdeljene sisteme.

3.2.1.1 Mrtva zanka

V sistemu, kjer uporabljamo zaseganje podatkov, morajo transakcije najprej podatek zaseči, preden lahko do njega dostopijo (ga preberejo) ali ga spremenijo. Na primer, da imamo v sistemu dve transakciji. Prva transakcija (T_1) se bo izvedla na podatku A in nato na podatku B. Druga transakcija (T_2) se bo izvedla na podatku B in nato na podatku A. Če se obe transakciji sprožita hkrati, potem transakcija T_1 zaseže podatek A in transakcija T_2 zaseže podatek B. Sedaj poskuša transakcija T_1 zaseči podatek B, vendar mora najprej počakati, da ga transakcija T_2 sprostí. Hkrati pa poskuša transakcija T_2 zaseči podatek A, ki je že zasegla transakcija T_1 . Ta čaka, da transakcija T_2 sprostí podatek, transakcija T_2 pa čaka, da transakcija T_1 sprostí podatek. Problem medsebojnega čakanja, ki v tem trenutku nastopi, imenujemo **mrtva zanka** (angl. deadlock).

Medsebojno čakanje transakcij, da sprostijo podatke, je pogost problem, ki se pojavlja pri zaseganju podatkov. Sistem za upravljanje s podatkovno bazo mora poskrbeti za ugotavljanje in reševanje zaseganj, pri katerih lahko pride do mrtve zanke (slika 9). Za ugotavljanje, ali je prišlo do mrtve zanke, se najpogosteje uporabljajo grafi "čaka-na" (angl. wait-for graphs) ali Petrijeve mreže (Concurrency: Deadlock and Starvation, 2003, str. 1-24).

Slika 9: Enostavna mrtva zanka.



Vir: Concurrency: Deadlock and Starvation, 2003, str. 6.

Grafi "čaka-na" nam z usmerjenimi povezavami med transakcijami prikažejo, katera transakcija čaka na drugo transakcijo, da sprostijo podatek. Če v grafu obstaja cikel, potem vemo, da je prišlo do mrtve zanke. Grafi "čaka-na" so lahko zelo kratki. Kot primer lahko vzamemo dve transakciji, kjer prva čaka na drugo in druga na prvo, da sprostijo podatek:

$$T_1 \rightarrow T_2 \rightarrow T_1$$

Lahko imamo tudi daljše grafe "čaka-na", ki zajemajo medsebojno odvisnost zaseganj transakcij na vseh porazdeljenih sistemih.

Sistem za upravljanje s podatkovno bazo ugotovi, da je prišlo do mrtve zanke, ko najde cikel v grafu "čaka-na". Takrat lahko eno izmed transakcij prekine, razveljavi vse njene akcije in jo izvede kasneje. S tem drugim transakcijam omogoči izvajanje brez nastopa mrtve zanke.

3.2.1.2 Navidezna mrtva zanka

Pri porazdeljenih sistemih prihaja do različno dolgih zakasnitev v medsebojni komunikaciji. Komunikacija med sistemi je pomembna za preprečevanje mrtvih zank. Centralni sistem, ki skrbi za preprečevanje mrtvih zank, zbira podatke o transakcijah in podatkih, nad katerimi se izvajajo. Kadar pride v omrežju do zasičenja podatkovnih povezav, njihovega pretrganja ali vzpostavljanja novih poti za podatke, je komunikacija otežena in lahko pride do težav, opisanih v naslednjem primeru (Bernstein, 1987, str. 64-67).

Transakcija T_1 zaseže podatek A in sistem za upravljanje s podatkovno bazo to sporoči centralnemu sistemu za preprečevanje mrtvih zank. Ker je pri dostopu do podatka A transakcija T_1 edina, ji centralni sistem dovoli zaseganje podatka in izvedbo akcije. Po izvedeni akciji transakcija T_1 podatka A ne potrebuje več, zato ga sprosti. V tem trenutku pa pride do zakasnitve v prenosu sporočila centralnemu sistemu za preprečevanje zank, hkrati pa v sistem pride transakcija T_2 , ki mora prav tako izvesti akcijo nad podatkom A, zato ga poskuša zaseči. Ker centralni sistem še vedno ni dobil informacije o sprostitvi podatka A, transakciji T_2 ne dovoli zaseči podatka A in transakcija T_2 se ne more izvesti, čeprav je transakcija T_1 že lokalno sprostila podatek. Takšno situacijo imenujemo **navidezna mrtva zanka** (angl. phantom deadlock) in povzroča nezaželene zakasnitve transakcij ali celo njihovo razveljavitev.

Primer poteka transakcij in komunikacije s centralnim sistemom za preprečevanje mrtvih zank je prikazan na sliki 10.

Slika 10: Primer navidezne mrtve zanke.

Čas	Transakcija T_1	Transakcija T_2
1	zahteva za zaseganje podatka A	
2	posredovanje zahteve za zaseganje centralnemu sistemu	
3	prejeto dovoljenje za zaseganje podatka A	
4	zaseganje podatka A	
5	izvedba transakcije nad podatkom A	
6	posredovanje zahteve za sprostitvev podatka A	zahteva za zaseganje podatka A
7		posredovanje zahteve za zaseganje centralnemu sistemu
8		ker centralni sistem še ni dobil zahteve za sprostitvev podatka A od transakcije T_1 , je zahteva transakcije T_2 zavrnjena

3.2.1.3 Živa zanka

Na primer, da imamo transakcijo T_1 , ki za svoje potrebe zasega veliko količino podatkov v podatkovni bazi. Poleg nje se v bazi izvaja tudi več krajših transakcij, ki zasegajo malo podatkov in se zato izvajajo hitreje. Zaradi preprečevanja mrtvih zank sistem prekine in razveljavi akcije transakcije T_1 in jo prestavi na kasnejši čas. Ker se v sistemu izvaja veliko drugih transakcij, ki zasegajo podatke, katere potrebuje tudi transakcija T_1 , obstaja možnost, da se transakcija T_1 ne bo nikoli izvedla. Vsako njeno izvajanje privede do nastopa nove mrtve zanke, zato jo sistem prekine.

Prekinitev transakcije in poskus njenega ponovnega izvajanja kasneje se zdi učinkovita rešitev problema mrtve zanke, vendar lahko privede do novega problema, ki ga imenujemo **živa zanka** (angl. livelock). Pri živi zanki se transakcija ne more izvesti, ker vsako njeno ponovno izvajanje privede do mrtve zanke in je vedno znova razveljavljena s strani sistema za upravljanje s podatkovno bazo. Za preprečitev problema žive zanke mora sistem za upravljanje s podatkovno bazo upoštevati prioriteto transakcije, ki je vedno višja, dlje ko je transakcija v sistemu. Če sistem razveljavi transakcijo z najnižjo prioriteto, se zmanjša možnost za nastanek žive zanke zaradi daljših transakcij.

3.2.1.4 Strategije za odkrivanje mrtvih zank

V centraliziranih sistemih je mogoče odkrivanje mrtvih zank z uporabo grafov "čaka-na". Tudi v porazdeljenih sistemih lahko odkrivamo mrtve zanke z njimi, vendar moramo zgraditi globalni graf "čaka-na", ki upošteva vse transakcije v porazdeljenem sistemu.

Transakcija T_1 dostopa do podatka A, zato ga zaseže na lokalnem porazdeljenem sistemu S_1 . Hkrati se izvaja tudi transakcija T_2 , ki zaseže podatek B na sistemu S_2 . V naslednjem koraku transakcija T_1 poskuša zaseči podatek B na sistemu S_2 , hkrati pa transakcija T_2 poskuša zaseči podatek A na sistemu S_1 . Graf "čaka-na" na sistemu S_1 ne vsebuje zanke: $T_2 \rightarrow T_1$. Tudi na sistemu S_2 graf "čaka-na" ne vsebuje zanke: $T_1 \rightarrow T_2$. Očitno porazdeljena sistema sama z uporabo lokalnih grafov "čaka-na" ne moreta ugotoviti, da sta se transakciji znašli v globalni mrtvi zanki. Zato moramo sestaviti globalni graf "čaka-na", tako da združimo vse lokalne grafe. V tem primeru dobimo nov graf, iz katerega lahko ugotovimo, da sta transakciji T_1 in T_2 v mrtvi zanki: $T_1 \rightarrow T_2 \rightarrow T_1$.

Poznamo tri strategije za odkrivanje mrtvih zank v porazdeljenih podatkovnih bazah:

1. centralizirana,
2. hierarhična in
3. porazdeljena.

Pri centraliziranem odkrivanju mrtvih zank izberemo enega izmed porazdeljenih sistemov, ki postane odgovoren za odkrivanje mrtvih zank. Ta sistem zbira informacije in ustvari globalni graf "čaka-na". Ker mora imeti izbrani sistem informacijo o vseh zaklenjenih podatkih, predstavlja komunikacija med njim in drugimi porazdeljenimi sistemi zaradi zakasnitev v omrežju velik problem. Komunikacijske povezave, ki vodijo do centralnega sistema za odkrivanje mrtvih zank, morajo tako biti ustrezno dimenzionirane, prav tako pa mora biti strežnik ustrezno zmogljiv, da zmore poleg upravljanja baze poskrbeti za pravočasno gradnjo grafov "čaka-na" in izvajati akcije v primeru nastopa mrtve zanke.

Če razporedimo porazdeljene sisteme v hierarhijo, kjer ima lahko vsak sistem svoj nadrejen sistem in podrejene sisteme, potem lahko uporabimo hierarhičen sistem za odkrivanje mrtvih zank. V tem primeru sistem, ki je nadrejen, zbira vse grafe "čaka-na" od podrejenih sistemov, jih vključi v svojega in pošlje nadrejenemu sistemu. Tako zmanjšamo količino komunikacije v primerjavi s centralno strategijo odkrivanja mrtvih zank, še vedno pa se večja količina potrebne obdelave informacij, ko gremo proti višjim mestom v hierarhiji.

Obstaja tudi nekaj predlogov za porazdeljeno strategijo odkrivanja mrtvih zank (Rob, 2001, str. 505). Pri slednjih je problem, da ima vsak sistem samo delno informacijo o globalnem grafu "čaka-na", tako je spet treba izmenjati precej informacij, preden določen sistem ugotovi, ali je prišlo do mrtve zanke ali ne.

3.2.1.5 Kaskadna razveljavitev transakcij

Kadar se transakcija T_1 konča z razveljavitvijo, mora sistem za upravljanje s podatkovno bazo povrniti podatke v stanje, preden jih je transakcija spremenila. Če je transakcija T_2 prebrala podatek, katerega je transakcija T_1 spremenila, potem mora biti razveljavljena tudi transakcija T_2 . Zaradi razveljavitve transakcije T_1 je transakcija T_2 prebrala napačen podatek. Razveljavitev sprememb podatkov ene transakcije posledično povzroči razveljavitev vseh transakcij, ki so dostopale do istih podatkov. Taki situaciji pravimo kaskadna razveljavitev transakcij.

Sistem za upravljanje s podatkovno bazo mora hraniti dovolj informacij o tem, katera transakcija je dostopila do katerih podatkov in kako jih je spreminjala, da

lahko povrne podatkovno bazo v celovito stanje. V primeru kaskadne razveljavitve mora sistem opraviti veliko dela, da razveljavi vse prizadete transakcije. Zato je kaskadna razveljavitev transakcij zelo nezaželena situacija (Garcia-Mollina, 2001).

Na srečo se kaskadni razveljavitvi lahko izognemo. Sistem mora zagotoviti, da transakcije lahko preberejo samo tiste podatke, ki so bili spremenjeni pri uspešno zaključenih transakcijah. Ker vemo, da se je transakcija že zaključila, lahko brez nevarnosti za kaskadno razveljavitev beremo ali spreminjamo podatke, do katerih je dostopala. Če pa se transakcija še ni zaključila, potem moramo z branjem ali pisanjem počakati. Tako preprečimo situacijo, ko razveljavitev predhodne transakcija vpliva na naslednje transakcije.

3.3 Časovno označevanje

Metode časovnega označevanja se zanašajo na zaporedje transakcij, označenih s časovno informacijo. Ker vsaka transakcija ob nastanku dobi časovno informacijo nastanka, sistem vedno ve, kakšno je pravo zaporedje izvajanja transakcij. Pri tem ne nastajajo mrtve zanke, ker ni zaseganja podatkov. Če pride do medsebojnega konflikta dveh ali več transakcij, se prva izvede tista z najstarejšo časovno oznako, druge pa so prekinjene in predstavljene na kasnejšo izvedbo. Časovna oznaka transakcije določa prioriteto transakcije. Starejša ko je transakcija, prej bo izvedena (Jensen, 2001, str. 3-4).

Zaporedje transakcij, ki ga dobimo s časovnim označevanjem, je enako zaporedju, ki bi ga imeli pri zaporednem izvajanju transakcij. Pri tem moramo zagotoviti, da nobeni transakciji nimata enake časovne oznake. Za označevanje se uporablja kar sistemska ura strežnika ali poseben števec, ki se poveča po vsaki označeni transakciji.

V porazdeljenem okolju lahko pride do odstopanja sistemskega časa. Rešitev z globalnim števcem je na prvi pogled privlačnejša, vendar bi bil v primeru izpada sistema z globalnim števcem transakcijski sistem onemogočen. Rešitev z redundantnim števcem, ki prevzame dolžnosti, če primarni izpade, je mogoča, vendar če izpad prizadene samo del omrežja, lahko pride do težav s sinhronizacijo med glavnim in pomožnimi števci. Zato so pogostejše rešitve, ki uporabljajo sistemski čas za označevanje transakcij. Pri tem je zelo pomembno, da imajo vsi strežniki, ki pri tem sodelujejo, enak sistemski čas. Sinhronizacija se tako lahko izvaja na podlagi prejetih časovnih oznak, ko strežniki sprejmejo transakcijo ali pa uporabijo način sinhronizacije sistemskega časa z uporabo omrežnih protokolov.

Omrežni protokol NTP (Network Time Protocol) omogoča sinhronizacijo systemskega časa glede na določen centralni časovni strežnik. Ponavadi se centralni časovni strežnik sinhronizira z lastno atomsko uro in tako zagotovi točen čas. NTP deluje uspešno tudi pri velikih oddaljenostih od centralnega časovnega strežnika in v omrežjih s počasnimi komunikacijskimi povezavami, pri tem pa povzroča zanemarljivo malo prometa (The NTP FAQ and HOWTO, 2003).

Pri zaseganju podatkov je nedeljivost transakcije zagotovljena z zaseganjem podatkov, nad katerimi se transakcija izvaja. Časovno označevanje ne zagotovi nedeljivosti transakcij, temveč samo njihovo pravilno zaporedje izvajanja. Zato imamo pri časovnem označevanju posebne vmesne pomnilnike, kamor transakcije zapisujejo rezultate sprememb, ki jih transakcija naredi na podatkih. Ko se transakcija izvede in uspešno konča, se spremembe prenesejo iz vmesnih pomnilnikov v bazo.

Ko dve ali več transakcij spremenijo isti podatek, pride do konflikta med transakcijami. Takrat se dovoli prepis spremenjenih podatkov iz vmesnega pomnilnika v bazo samo transakciji z najstarejšo časovno oznako. Druge transakcije se prekinejo in poskusijo izvesti kasneje. Ker prekinjene transakcije ne prepišejo vrednosti iz vmesnih pomnilnikov v bazo, je s tem zagotovljena celovitost podatkov.

Izvajanje vsake transakcije poteka v treh fazah:

1. faza branja podatkov,
2. faza preverjanja podatkov in
3. faza zapisovanja podatkov.

V fazi branja podatkov transakcija zbira podatke, potrebne za njeno izvedbo, in nad njimi opravlja izračune. Pri teh izračunih se nove vrednosti zapisujejo v vmesne pomnilnike. V fazi preverjanja podatkov transakcija preveri, ali spremembe, ki jih je naredila nad podatki, ne bodo ogrozile integritete podatkov in veljavnosti podatkovne baze. Če faza preverjanja ne ugotovi nobenih konfliktov, sledi faza zapisovanja podatkov, kjer se novi podatki prenesejo iz vmesnega pomnilnika v podatkovno bazo. S tem se transakcija zaključi.

3.3.1 Osnovno časovno označevanje

Pri metodi osnovnega časovnega označevanja lahko pride do konflikta pri transakcijah, ki so svoje spremembe že vnesle v vmesni pomnilnik. V tem primeru se izvede samo najstarejša transakcija. Druge transakcije v konfliktu se prekinejo in poskusijo izvesti kasneje (Bernstein, 1987a, str. 114-116).

Ko transakcija T_i poskusi spremeniti podatek A, mora sistem preveriti, ali ni kakšna starejša transakcija T_j , ki se še ni zaključila, tudi spremenila podatka A in se njegova nova vrednost nahaja v vmesnem pomnilniku. V tem primeru mora transakcija T_i počakati, da se transakcija T_j zaključi in se nova vrednost podatka A zapiše v bazo. Če je kakšna novejša transakcija T_k vmes že prebrala podatek A, jo moramo prekiniti in izvesti kasneje.

3.3.2 Previdno časovno označevanje

Problem pri osnovnem časovnem označevanju je ponavljanje transakcij pri konfliktih. Previdno časovno označevanje poskuša zmanjšati potrebo po ponavljanju konfliktnih transakcij na račun zmanjšanja možnosti hkratnega izvajanja transakcij.

Previdno časovno označevanje zakasni transakcijo, dokler ne ugotovi, da pri njenem izvajanju ne bo prišlo do konflikta s kakšno starejšo transakcijo, ki se že izvaja. To zagotovi s posebnimi vrstami, kamor se vstavljajo transakcije glede na njihovo časovno oznako. Preden se transakcijo uvrsti v vrsto, sistem preveri, če transakcija pri izvajanju sprememb podatkov ne bo posegla v podatke transakcij, ki so že v vrsti. V primeru konflikta se transakcija enostavno zakasni (Bernstein, 1987b, str. 116-117).

Z zakasnitvijo transakcije zaradi mogočega konflikta se izognemo ponovnemu izvajanju transakcije, vendar zmanjšamo število transakcij, ki bi se lahko hkrati izvajale v sistemu. Poleg tega pa morajo vsi porazdeljeni sistemi periodično oddajati "prazne" transakcije zaradi razpoznavanja pravičnega vrstnega reda, ko določena transakcija dejansko prispe v izvedbo.

3.4 Optimistične metode

Optimistične metode za zagotavljanje podatkov dovoljujejo neovirano izvajanje vseh transakcij brez čakanja ali sinhronizacije. Pri tem se zanašajo na dejstvo, da prihaja do konfliktov med transakcijami zelo poredko.

Zaradi dovoljenega izvajanja vseh transakcij pri optimističnih metodah se rezultati transakcij hranijo v lokalnih kopijah, ki se ustvarijo za vsako transakcijo posebej. Ko se transakcija zaključi, sistem preveri, ali je prišlo do konflikta med transakcijo,

ki je zaključena, in med drugimi transakcijami. Če se konflikt pojavi, se transakcija prekine in izvede znova kasneje.

Vsekakor se odvečno delo, ki ga sistem opravi pri konfliktnih transakcijah, pozna na zmanjšani zmogljivosti sistema, vendar se optimistične metode uporabljajo takrat, ko pričakujemo zelo malo konfliktov. V tem primeru se transakcije izvajajo hitro in brez večjih zakasnitev.

Zaradi omejitev pri izvajanju transakcij podatkovne baze le poredko uporabljajo to rešitev za zagotavljanje podatkov.

3.5 Metoda večine

Metoda večine (angl. quorum consensus) je novejša metoda, ki jo uporabljamo za zagotavljanje podatkov v porazdeljenih sistemih, predvsem pri repliciranih podatkovnih bazah. Preden lahko transakcija izvede akcijo (dostop ali spremembo podatka), mora dobiti dovoljenje skupine porazdeljenih sistemov (Bernstein, 1987c, str. 298-301).

Metodo večine lahko primerjamo z volitvami, kjer je glas posameznika utežen in kjer je določen prag izvolitve. Večinska skupina je definirana kot množica sistemov, katerih skupna volilna moč presega volilni prag. Transakcija dobi dovoljenje za izvedbo akcije le, če zbere dovolj glasov, da je presežen volilni prag. Pri tem je določeno minimalno število sistemov, ki morajo poslati dovoljenje, da je akcija odobrena. Večinski prag za branje je manjši kot večinski prag za spreminjanje podatkov, seštevku obeh pa mora biti večji kot seštevku volilne moči vseh sistemov skupaj. S tem je zagotovljeno, da ima vsaka večinska skupina za dovolitev pisanja vsaj enega člana iz katere koli večinske skupine za branje in katere koli večinske skupine za pisanje.

Težave, ki se pojavljajo pri metodi večine, izvirajo predvsem iz količine informacije, ki jo je treba prenesti prek omrežja, da se odloči, ali bo transakcija dobila pravico izvedbe. Čeprav gre samo za bralni dostop do podatkov, moramo dobiti dovoljenje večine sistemov. Zato ponavadi izberemo dva volilna praga: enega za bralni dostop in enega za pisalni dostop do podatkov.

Težave lahko povzroči tudi nepravilno izbran volilni prag. Če je ta previsok, potem bodo transakcije večinoma zavržene. Če je prenizek, se lahko zgodi, da razpade na dva ali več skupin, znotraj katerih je mogoče dosegati volilni prag. Transakcije lahko tako skupno bazo spravijo v neveljavno stanje, ko sistem spet postane dosegljiv v celoti.

Dobra stran metode večine je, da celoten sistem lahko deluje, tudi če pride do izpada dveh ali več strežnikov. Slabost metode je čas, ki je potreben, da transakcija ugotovi, ali ima večina dovoljenje za izvedbo operacije. Prav tako se pojavi težava, ko imamo malo sistemov, ki sodelujejo v metodi večine. Če pride do izpada enega sistema, potem potrebujemo vsaj še dva delujoča sistema, če pride do izpada dveh sistemov, potrebujemo vsaj tri delujoče sisteme.

Obstaja več načinov za izvedbo sistemov, ki uporabljajo metodo večine (drevesni sistem, mrežni sistem, hierarhični sistem). Pri vseh metodah gre za zmanjševanje števila glasov, potrebnih za doseganje večine, in s tem zmanjševanje količine potrebne komunikacije med porazdeljenimi sistemi. Z manjšo količino komunikacije in s tem čakanja na pridobitev dovoljenja za izvedbo transakcije se poveča hitrost obdelave transakcij.

3.6 Zagotavljanje podatkov ob napakah med izvajanjem transakcij

Opisane metode: zaseganje podatkov, časovno označevanje, optimistične metode in metode večine rešujejo problem, kaj narediti s podatki, ki jih je transakcija spremenila, oziroma kako zagotavljati podatke med izvajanjem transakcij. Naslednji metodi pa se ukvarjata s tem, kako zagotoviti podatke, ko pri izvajanju transakcije pride do konflikta ali napake.

Do napake lahko pride zaradi (Bell, 1992a, str. 232):

1. napake med izvajanjem transakcije,
2. izpada računalniškega sistema,
3. izpada pomnilniškega medija ali
4. izpada omrežja.

Napake med izvajanjem transakcije so lahko: enostaven konflikt med dvema ali več izvajajočimi se transakcijami, napaka, do katere pride ob programski izvedbi transakcije (napačni podatki, napačna operacija), ali napaka, do katere pride zaradi hrošča v programu. V teh primerih se transakcija razveljavi.

Do izpada računalniškega sistema lahko pride zaradi izpada električnega omrežja, napake na vodilu, procesorju ali perifernih enotah. Napaka ob izpadu sistema prizadene vse transakcije, ki se izvajajo. Ob ponovnem zagonu sistema mora sistem za upravljanje s podatkovno bazo ugotoviti, v kakšnem stanju je bil sistem

tik pred napako, in poskusiti na podlagi dnevnika transakcij razveljaviti spremembe tistih transakcij, ki se niso uspele končati.

Prav tako so prizadete vse transakcije, ki se izvajajo na sistemu, če pride do izpada pomnilniškega medija – trdi disk ali, manj verjetno, glavni pomnilnik računalnika. Izpad trdega diska lahko rešujemo z arhiviranjem podatkov in z zrcaljenjem podatkov (angl. mirroring) na več neodvisnih trdih diskov. Pri arhiviranju lahko podatkovno bazo postavimo v stanje, v katerem je bila v trenutku izdelave arhiva. Pri uporabi zrcaljenja lahko sistem neovirano deluje naprej, skrbnik pa mora čim prej zamenjati okvarjen trdi disk.

Danes so omrežja zelo zanesljiva. Do izgube podatkov, ki je včasih bila bolj pravilo kot izjema, danes praktično ne prihaja več. Sodobni usmerjevalni protokoli omogočajo hitro odkrivanje pretrganih komunikacijskih povezav in preusmerjanje prometa na vzporedne poti. Težave se pojavijo šele, ko povsem izgubimo povezljivost med dvema ali več omrežnimi segmenti. V tem primeru zaradi onemogočene komunikacije ne moremo ugotoviti, ali je prišlo do izpada komunikacijskih povezav ali pa je prišlo do izpada strežnika.

Transakcija, ki se izvaja nad porazdeljenimi podatkovnimi bazami, izvaja operacijo spremembe podatka. Na dveh sistemih se uspešno konča, na tretjem pa pride do napake. Ker se je transakcija na prvih dveh sistemih uspešno končala, so bili njeni rezultati trajno zapisani v podatkovno bazo v skladu s pravili o nedeljivosti in trajnosti transakcij. Podatke končanih transakcij lahko spravimo v stanje pred izvedbo transakcije, vendar lahko to povzroči kaskadno razveljavitev transakcij. Zato moramo v porazdeljenih sistemih poskrbeti, da do takšnih situacij ne pride.

Dve metodi, ki poskušata rešiti težave z zaključevanjem transakcij v porazdeljenih sistemih, sta:

1. zaključevanje v dveh fazah in
2. zaključevanje v treh fazah.

3.6.1 Zaključevanje v dveh fazah

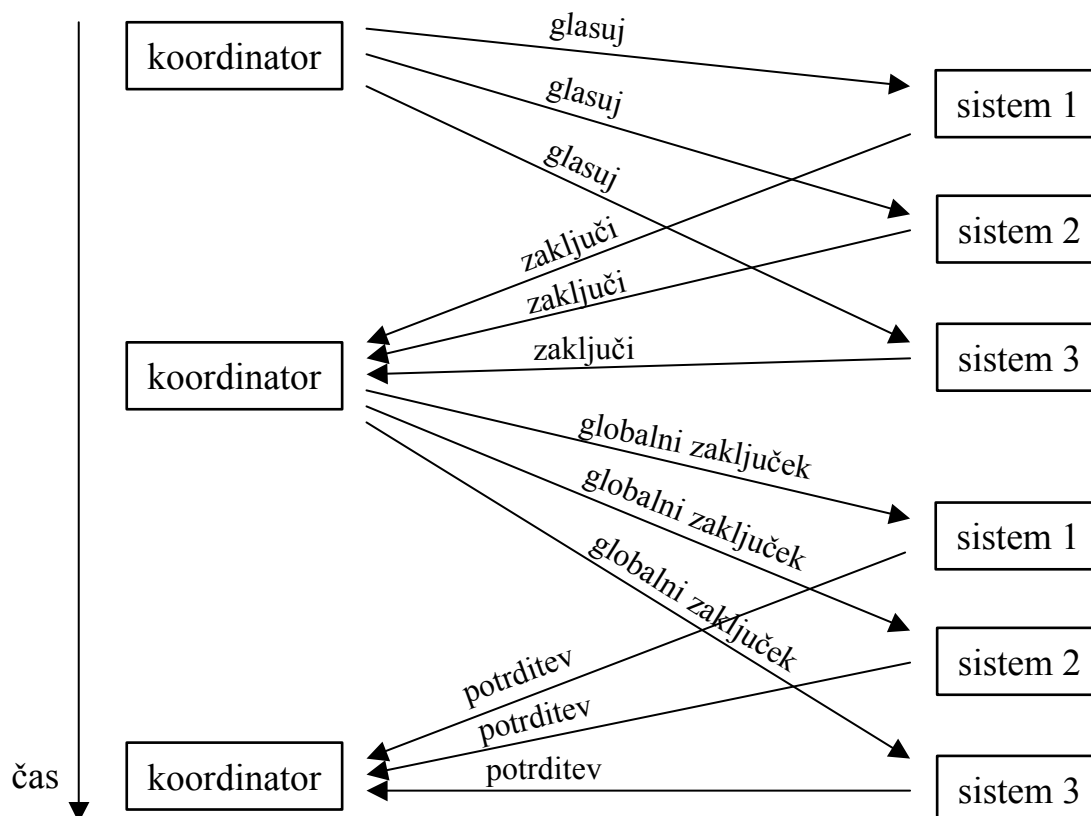
Kot že ime metode pove, gre pri zaključevanju v dveh fazah za izvedbo dveh operacij pri zaključevanju transakcij:

1. faza glasovanja in
2. faza odločitve.

V fazi glasovanja koordinator porazdeljenih sistemov vpraša posamezne sisteme, na katerih se izvaja transakcija, ali so pripravljeni na njen zaključek. Če v tem

trenutku kateri koli izmed sistemov glasuje za prekinitev transakcije ali če koordinator ne dobi odgovora, pošlje ukaz za razveljavitev transakcije in podatki se povrnejo v prvotno stanje. Če vsi sistemi glasujejo za zaključek transakcije, koordinator pošlje ukaz za globalni zaključek transakcije (Bell, 1992b, str. 248-256). Potek zaključevanja v dveh fazah prikazuje slika 11.

Slika 11: Zaključevanje v dveh fazah.



Vir: prirejeno po Bell, 1992, str. 250.

Sistemi morajo po oddaji glasu čakati, da koordinator zbere vse glasove in se odloči, ali bo dovolil zaključek transakcije ali ne. Če koordinator ne dobi odgovorov dovolj hitro, se lahko odloči za razveljavitev transakcije. Dokler sistemi ne dobijo odgovora koordinatorja, ali lahko končajo transakcijo ali ne, morajo čakati. Če je vmes prišlo do izpada koordinatorja, bi sistemi čakali v nedogled. Da se izognemo čakanju, po določenem času izvolimo novega koordinatorja, ki poskrbi, da procesi spet stečejo normalno.

Opisani postopek ponazarja centralno metodo zaključevanja v dveh fazah. Pri linearni topologiji se informacija o globalni odobritvi ali zavrnitvi transakcije prenaša od sistema do sistema, dokler ne doseže zadnjega sistema in ne zakroži naprej do prvega sistema. Pri porazdeljeni topologiji pa se glasovi izmenjajo med

vsemi sodelujočimi sistemi in se tako lahko sami odločijo, ali bodo transakcijo zaključili ali prekinili.

Zaključevanje v dveh fazah je najenostavnejša in najpogosteje uporabljena metoda. Velika pomanjkljivost, ki je bila povod za iskanje boljših metod, je čakanje na odločitev koordinatorja, ne da bi sistem vedel, ali lahko transakcijo zanesljivo zaključi ali naj jo razveljavi.

3.6.2 Zaključevanje v treh fazah

Metoda zaključevanja v treh fazah poskuša rešiti problem čakanja brez ustrezne odločitve o nadaljnjih akcijah pri zaključku transakcije.

Pri zaključevanju v treh fazah imamo dodatno vmesno fazo predodločitve med fazo glasovanja in fazo odločitve. Tako imamo kar pet možnih trenutkov, ko procesi čakajo na odločitev (Bell, 1992c, str. 256-265):

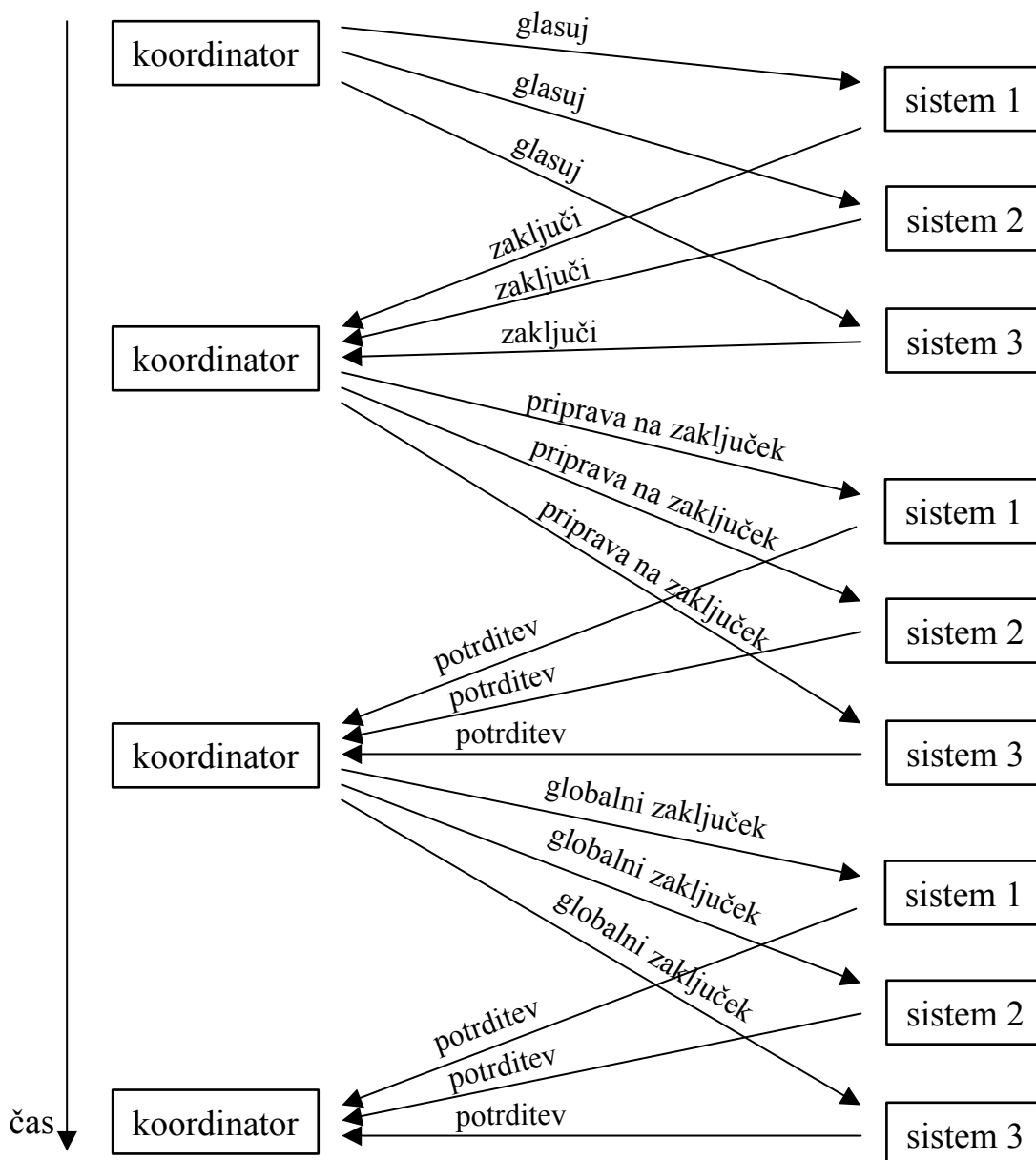
1. ko sistemi čakajo na koordinatorjevo zahtevo za glasovanje,
2. ko koordinator čaka na glasove sistemov,
3. ko se sistemi odločajo o zaključku ali razveljavitvi transakcije,
4. ko koordinator čaka na odločitve in
5. ko sistemi čakajo na potrditev zaključka transakcije.

Če pride do izpada sistemov ali koordinatorja v prvem in drugem primeru, potem vsi sistemi po poteku določenega časa vedo, da noben ni mogel dobiti potrditve zaključka transakcije, zato izvedejo njeno razveljavitev.

V tretjem in petem primeru imamo lahko največ težav s čakanjem na odločitev. V tem primeru so transakcije ustavljene, dokler sistemi ne uspejo izvoliti novega koordinatorja in pridobiti potrditve akcije.

V četrtem primeru koordinator ne dobi vseh odločitev sistemov. Mogoče je, da je prišlo do izpada enega ali več sistemov. Če so v tem primeru vsi drugi sistemi glasovali za zaključek, koordinator ukaže izvedbo zaključka transakcije. Izpadli sistemi bodo ob ponovni postavitvi ugotovili, da je prišlo do napake pri izvajanju transakcije, in bodo poskusili vzpostaviti celovito stanje v sodelovanju z drugimi sistemi.

Slika 12: Zaključevanje v treh fazah.



Vir: prirejeno po Bell, 1992, str. 260.

Zaključevanje v treh fazah (slika 12) vpeljuje bolj deterministično delovanje metode v primerjavi z zaključevanjem v dveh fazah, vendar pri tem doda novo fazo, ki poveča kompleksnost in količino izmenjanih sporočil. To je glavni razlog, da jo kljub prednostim redko srečamo v porazdeljenih podatkovnih bazah.

4 RAČUNALNIŠKA OMREŽJA

Doseganje velike procesorske moči in pomnilniškega prostora na enem samostojnem računalniškem sistemu je sicer mogoče, vendar je to zelo draga

rešitev, ki si jo lahko privoščijo predvsem velika podjetja. Cenejša rešitev je povezava več manj zmogljivih, vendar bistveno cenejših računalniških sistemov v večji sistem. Pri tem zmogljivost ne narašča linearno zaradi dodatnega procesiranja, ki ga zahtevata povezovanje in koordiniranje, zato pa je cena sistema bistveno nižja od enako zmogljivega samostojnega računalnika. Poleg cene je prednost več sistemov tudi večja razpoložljivost v primeru napak na sistemih ali omrežju.

Temeljna naloga računalniških omrežij je prenos podatkov med oddaljenimi računalniškimi sistemi. Računalniška omrežja morajo zagotoviti pravilno usmerjanje podatkov od izvora do ponora in s tem omogočiti izmenjavo podatkov.

V zadnjem desetletju smo pričali nenehni hitri rasti in razvoju na področju podatkovnih komunikacij in računalniških omrežij. Čedalje več prometa v omrežjih je povezano s stroški za nakup dodatnih zmogljivosti. Aplikacije, ki uporabljajo prenos podatkov, morajo zato čim bolj učinkovito izkoriščati pasovno širino, ki jim je na voljo.

S ciljem omogočiti čim bolj enostaven in neodvisen razvoj komunikacijskih sistemov in aplikacij je svetovna organizacija za standarde (International Organization for Standardization - ISO) razvila odprt omrežni model za povezovanje (Open System Interconnection - OSI).

4.1 Omrežni model ISO/OSI

Slojna arhitektura omrežnega modela omogoča ločenost in neodvisnost posameznih slojev z dobro definiranimi vmesniki tako navzgor proti višjim slojem kakor tudi navzdol proti nižjim slojem.

Prednosti slojne arhitekture so (Criticisms of the ISO OSI Reference Model, 2003):

1. spreminjamo lahko kateri koli sloj, ne da bi pri tem vplivali na druge sloje,
2. modularni slojni pristop poenostavlja celotno arhitekturo sistema,
3. posamezne sloje lahko razvijajo različne razvojne skupine,
4. tehnologijo na posameznih slojih lahko zamenjamo s popolnoma drugačno tehnologijo,
5. funkcije nižjega sloja si lahko delijo in uporabljajo različne funkcije višjih slojev in
6. še posebej na nižjih slojih lahko funkcionalnost prenesemo iz programske opreme na strojno opremo, kar prinese večjo hitrost.

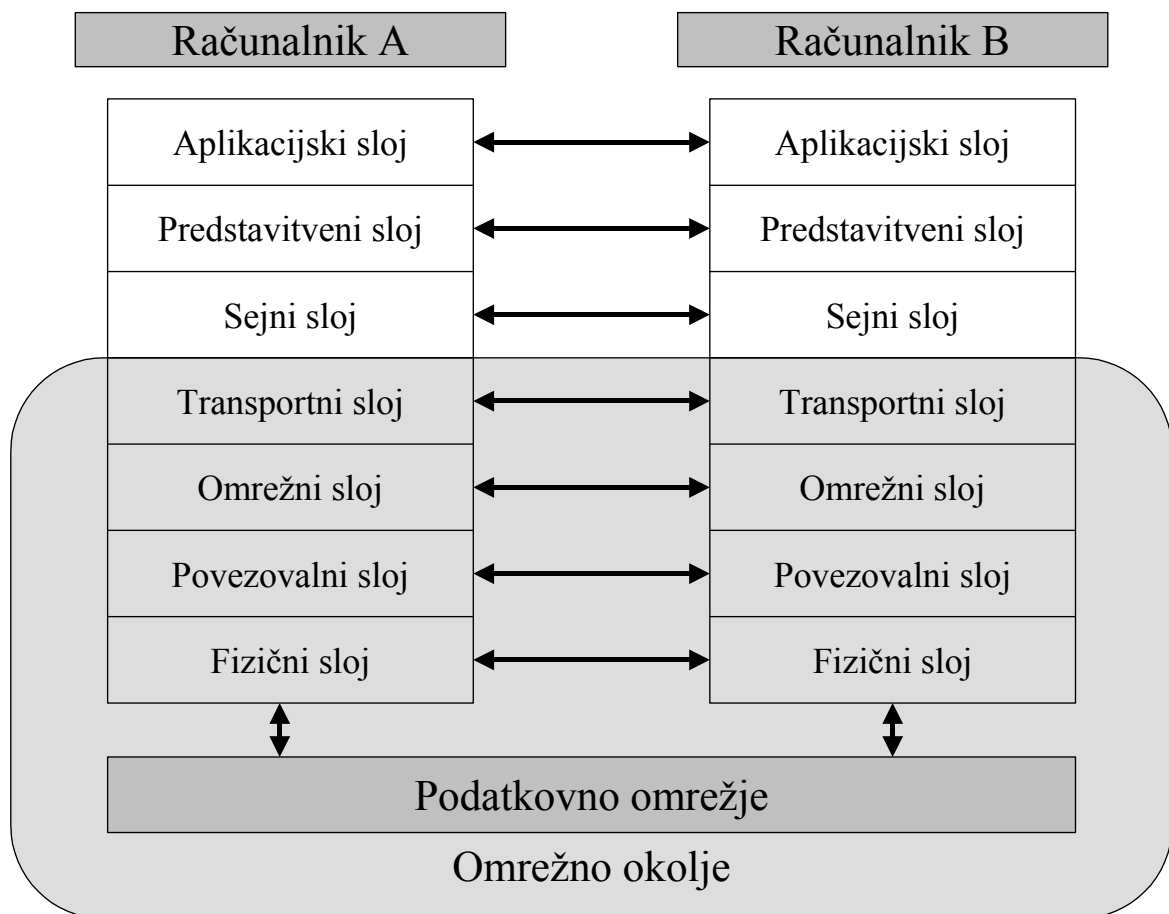
Natančna ločitev funkcij v slojni arhitekturi ima tudi slabe lastnosti:

1. vsak sloj doda svojo informacijo zaradi identifikacije, kar povzroči povečanje količine informacije, ki se mora prenesti,
2. zaradi delitve na neodvisne in samostojne sloje vsak sloj podvaja določene funkcije drugih slojev in
3. z napredkom tehnologije se določene funkcije znajdejo na sloju, kjer njihova uporaba ni najučinkovitejša.

Omrežni model ISO/OSI sestavlja sedem slojev (slika 13):

1. fizični sloj,
2. povezovalni sloj,
3. omrežni sloj,
4. transportni sloj,
5. sejni sloj,
6. predstavitveni sloj in
7. aplikacijski sloj.

Slika 13: Omrežni model ISO/OSI.



Vir: Sloman, 1994, str. 19.

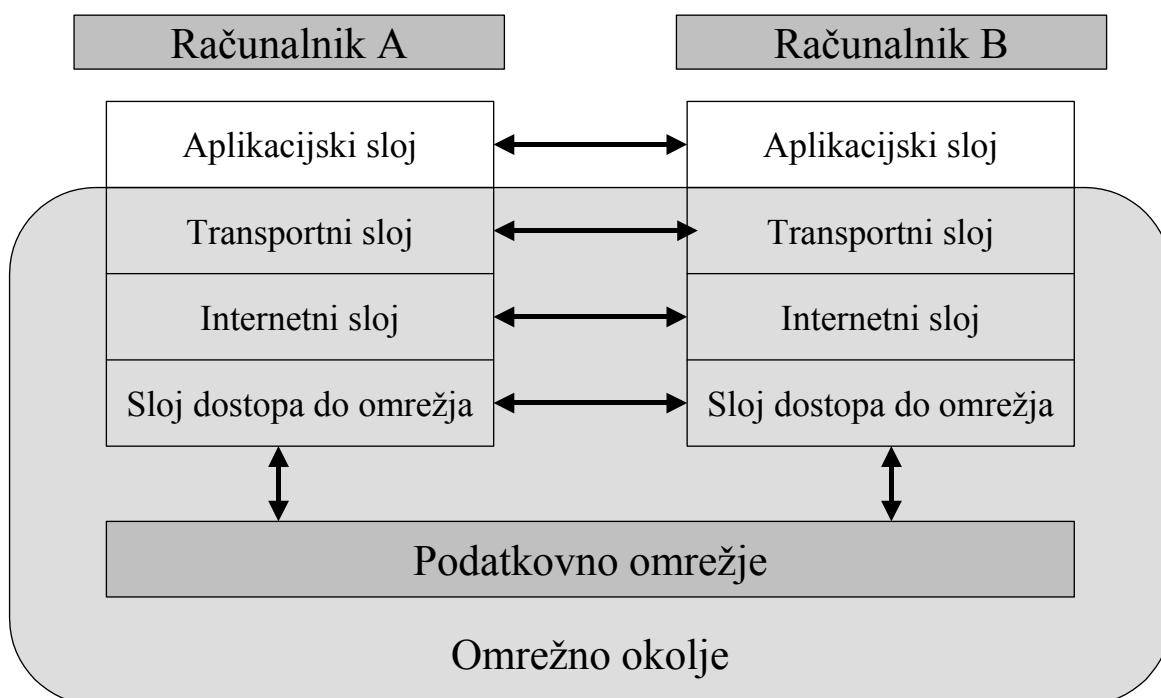
Neodvisnost slojev se kaže v dejstvu, da neki aplikaciji na aplikacijskem sloju ni treba skrbeti za varnost prenesene informacije, ker za varnost lahko poskrbi kodiranje podatkov na nižjem, predstavitvenem sloju.

Ravno zaradi dodatne kompleksnosti modela, ki je počasi nastajal pod okriljem organizacije ISO, in podvajanja funkcij med sloji se je razvil konkurenčni omrežni model TCP/IP (Transmission Control Protocol/Internet Protocol).

4.2 Omrežni model TCP/IP

Omrežni model TCP/IP je dobil ime po imenih protokolov, ki jih uporablja na transportnem in omrežnem sloju. Za razliko od omrežnega modela ISO/OSI ima tri sloje manj, kar mu omogoča večjo fleksibilnost. Ravno večja fleksibilnost, enostavnost za razvoj aplikacij in nizka cena ali celo brezplačnost le-teh so razlogi, da je danes omrežni model TCP/IP prevladujoč model pri končnih uporabnikih. Model IOS/OSI uporabljamo pogosteje in skoraj izključno v omrežjih nacionalnih operaterjev telekomunikacijskih storitev.

Slika 14: Omrežni model TCP/IP.



Vir: prirejeno po Postel, 1981, str. 8.

Omrežni model TCP/IP sestavljajo štirje sloji (slika 14):

1. sloj dostopa do omrežja,
2. medmrežni sloj,

3. transportni sloj in
4. aplikacijski sloj.

4.3 Omrežne tehnologije po slojih modela ISO/OSI

Če za referenco vzamemo omrežni model ISO/OSI, potem gre pri najnižjem, prvem ali fizičnem sloju za definicijo ravni signalov in povezav, ki prek medijev (bakrene žice, optičnih vlaken ali radijskih valov) prenašajo podatke. Za prepoznavanje teh signalov na obeh straneh komunikacijske povezave prvi sloj določa, kateri podatek je predstavljen glede na prejet signal. Za pravilno predstavitev podatkov glede na signal morata obe strani pošiljati ustrezne signale in pravilno razpoznavati prejete signale. Ker je fizični sloj ločen od višjih slojev, lahko po kateri koli fizični povezavi pošljamo podatke katerega koli protokola višjega sloja (Minoli, 1993, str. 85-88).

Drugi, povezovalni sloj zagotavlja zanesljiv prenos podatkov prek fizičnega sloja. Povezovalni sloj določa tudi fizično naslavljanje, fizično topologijo omrežja, omogoča zaznavanje ter odpravljanje napak, ki nastajajo na komunikacijski povezavi na nižjem sloju, kontrolo pretoka in označevanje zaporedja podatkovnih blokov, ki jih na tem sloju imenujemo okvirji (angl. frames).

Tretji, omrežni sloj omogoča višjim slojem neodvisnost od podatkovnih poti in topologije omrežja. Omrežni sloj zagotavlja usmerjanje podatkov s ciljem združevanja oddaljenih omrežij. Na omrežnem sloju imamo zato logične naslove, na podlagi katerih prihaja do odločitev, kam bo kateri podatkovni blok, ki se na tem sloju imenuje paket (angl. packet), usmerjen. Omrežni sloj zagotavlja dostavo paketov tudi pri izpadu določenega segmenta iz omrežja, če imamo na voljo alternativne poti. Omrežni elementi, ki se na tem sloju imenujejo usmerjevalniki, si izmenjujejo informacije o oddaljenih omrežjih in gradijo usmerjevalne tabele, ki omogočajo prenos paketov od enega oddaljenega omrežja do drugega.

Četrty, transportni sloj zagotavlja zanesljiv in transparenten prenos med eno lokalno točko in dvema ali več oddaljenima točkama. Na transportnem sloju gre za kontrolo prenosa, multipleksiranje sej, upravljanje z navideznimi povezavami ter kontrolo in ugotavljanje napak. Pri modelu TCP/IP imamo na tem sloju dva protokola. Protokol TCP (Transmission Control Protocol) za prenos podatkov med dvema oddaljenima točkama najprej vzpostavi sejo z uporabo rokovanja v treh korakih (angl. three way handshake) in šele nato začne s pošiljanjem podatkov. Protokol UDP (User Datagram Protocol) pri svojem delovanju ne vzpostavi seje med dvema oddaljenima točkama, ampak zagotavlja le osnovno transportno

funkcijo. Pri svojem delovanju omogoča le multipleksiranje sej in ugotavljanje napak.

Peti, sejni sloj skrbi za vzpostavljanje, vzdrževanje in zaključevanje seje med dvema aplikacijama na podlagi modela zahteve in odgovora (angl. request-response model). Včasih se seja vzpostavi samo za prenos podatka v eno smer, ponavadi pa seje trajajo dlje in omogočijo izmenjavo več informacij, dokler je ena ali obe strani ne prekineta. Za aplikacije medmrežja je seja povezana z vrati (angl. port), ki jih odpre aplikacija. Primer so vrata številka 80, na katerih ponavadi čaka na zahtevo strežnik HTTP.

Na šestem sloju, imenovanem predstavitveni sloj, gre za predstavitev podatkov (kodiranje) in za pretvorbo podatkov, ki izvirajo iz enega sistema v podatke, ki jih drugi sistem razume. Ponavadi je predstavitveni sloj del operacijskega sistema.

Na zadnjem, sedmem, aplikacijskem sloju, ki je najbližje uporabniku, gre za kontrolo komunikacije med dvema oddaljenima aplikacijama.

5 OMREŽNA PODPORA KOMUNIKACIJI ENEGA Z VEČ SPREJEMNIKI

Na eni strani komunikacijskega spektra je usmerjeno oddajanje (angl. unicast). Pri tem gre za komunikacijo, kjer oddajnik pošilja podatke enemu sprejemniku. Na drugi strani komunikacijskega spektra je razpršeno oddajanje (angl. broadcast). Pri tem gre za komunikacijo, kjer oddajnik pošilja podatke vsem mogočim sprejemnikom. Vmes pa je večsmerno oddajanje (angl. multicast), kjer oddajnik pošilja podatke samo določeni skupini sprejemnikov.

5.1 Usmerjeno in razpršeno oddajanje

Usmerjeno oddajanje je podprto v celotnem medmrežju. To je namreč način, kako računalniki, ki so povezani v medmrežje, komunicirajo med seboj. Ko želi oddajnik poslati podatke, jih vstavi v podatkovni paket medmrežnega protokola (slika 15). Pri oddaji paketa v omrežje vstavi vanj naslov sprejemnika, kateremu je paket namenjen. Podatkovni paket vsebuje tudi naslov oddajnika, ki omogoča dvosmerno komunikacijo.

Slika 15: Podatkovni paket medmrežnega protokola.

Naslov oddajnika	Naslov sprejemnika	Podatki
------------------	--------------------	---------

Vir: Wright, 1995, str. 326.

Za razliko od usmerjenega oddajanja, kjer je sprejemnikov naslov vedno znan vnaprej, pri razpršenem oddajanju naslov sprejemnika niti ni pomemben. Oddajnik pri razpršenem oddajanju želi, da bi podatkovni paket prejeli vsi mogoči sprejemniki, ki se ponavadi nahajajo na istem omrežnem segmentu. V tem primeru oddajnik za naslov sprejemnika navede poseben naslov, ki ga vsi sistemi upoštevajo kot lastni naslov. Namesto da bi oddajnik ustvaril podatkovni paket za vsakega sprejemnika posebej, lahko na način razpršenega oddajanja naslovi paket na vse sprejemnike. Od slednjih pa je odvisno, ali bodo poslani paket sprejeli in upoštevali.

5.2 Pregled tehnologije za večsmerno oddajanje

Pri večsmernem oddajanju gre za kombinacijo usmerjenega oddajanja z razpršenim oddajanjem. Večsmerno oddajanje združuje prednosti metod na obeh koncih komunikacijskega spektra in se umešča vmes med usmerjeno in razpršeno oddajanje. Prednost, ki jo vzame iz usmerjenega oddajanja, je doseganje oddaljenih sprejemnikov, kajti usmerjevalniki ne dovoljujejo razpršenega oddajanja zunaj lokalnega omrežnega segmenta. Prednost, ki jo vzame iz razpršenega oddajanja, pa je pošiljanje ene same kopije podatkov več sprejemnikom, kar je nemogoče doseči z usmerjenim oddajanjem, kjer moramo za vsak sprejemnik poslati nov paket z istimi podatki.

Prednosti večsmerne oddajanja, ko moramo poslati podatke več sprejemnikom, so:

1. optimalna uporaba pasovne širine komunikacijskih povezav,
2. manjša obremenitev strežnikov in
3. manjša obremenitev omrežja in omrežnih elementov.

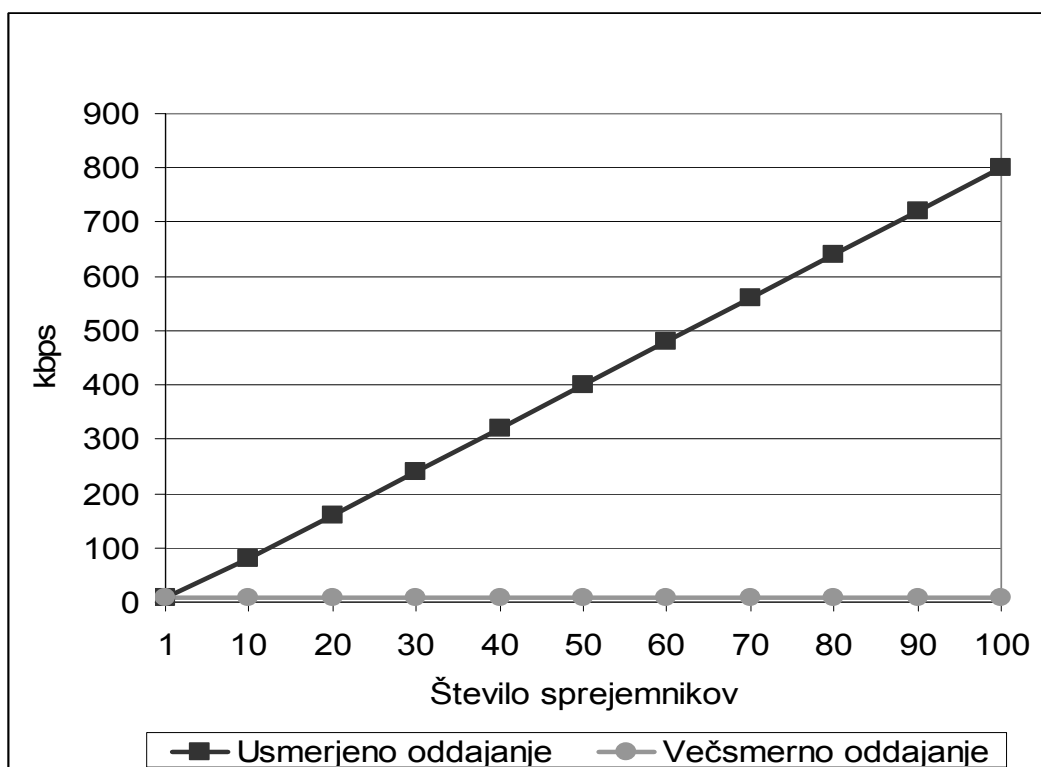
5.2.1 Optimalna uporaba pasovne širine komunikacijskih povezav

Z naraščanjem števila uporabnikov omrežja (pri tem mislim predvsem na število aplikacij, ki uporabljajo podatkovne povezave za prenos podatkov) narašča količina podatkov, ki se prenaša po omrežju. Z uporabo tehnologije večsmerne oddajanja, kjer pošljemo podatek več naslovnikom hkrati, namesto da pošiljamo

podatek vsakemu naslovníku posebej, zmanjšamo količino po nepotrebnem prenesenih kopij istih podatkov in s tem razbremenimo komunikacijske povezave.

Za primerjavo vzemimo prenos govora, ki zaseda komunikacijsko povezavo z 8 kbps (kilobitov na sekundo). Če pošiljamo podatke s tehnologijo usmerjenega oddajanja, moramo za vsakega sprejemnika poslati svoj podatkovni paket. To pomeni, da bo obremenitev omrežja naraščala linearno s številom sprejemnikov. Ko bo sprejemnik samo en, bo omrežje obremenjeno s prometom 8 kbps, ko pa bo v omrežju 10 sprejemnikov, bo omrežje obremenjeno z 80 kbps. Pri večsmernem oddajanju oddajamo samo eno kopijo podatkovnega paketa ne glede na število sprejemnikov, saj je paket naslovljen na skupino sprejemnikov. To pa pomeni, da bo omrežje obremenjeno s prometom 8 kbps ne glede na število sprejemnikov (slika 16).

Slika 16: Prikaz zasedenosti komunikacijske povezave v odvisnosti od načina oddajanja in števila sprejemnikov.



Vir: Williamson, 2000, str. 8.

5.2.2 Manjša obremenitev strežnikov

Strežniki, ki oddajajo pakete v omrežje na način usmerjenega oddajanja, morajo pri tvorbi vsakega paketa vstaviti tudi naslov sprejemnika. Če je sprejemnikov več

tisoč ali več milijonov, je procesorski čas, ki ga mora še tako hiter strežnik vložiti v tvorbo paketov, nezanemarljiv. Upoštevati moramo tudi zasedenost pomnilniškega prostora, kjer mora strežnik hraniti tabelo naslovov vseh sprejemnikov.

Če strežnik uporablja večsmerno oddajanje, teh težav ni, kajti za vsak poslan podatek mora strežnik ustvariti samo en paket, ki ga naslovi na skupino sprejemnikov.

Poleg zasedenosti zmogljivosti strežnika moramo upoštevati tudi čas sprejema paketa sprejemnika. Pričakovano je, da sprejemniki, ki so na istem omrežnem segmentu, prejmejo pakete ob približno istem času. Ko mora oddajnik naslavljati tisoče ali milijone sprejemnikov, bo časovna zakasnitev istega podatkovnega paketa pri prvem in zadnjem sprejemniku občutna. Za določene aplikacije (transakcijski sistemi), kjer je pravočasnost informacije nujna, si take zakasnitve ne moremo privoščiti.

5.2.3 Manjša obremenitev omrežja in omrežnih elementov

Manj ko imamo podatkovnih paketov v omrežju, manjša je obremenitev podatkovnih poti in omrežnih elementov, ki skrbijo za usmerjanje prometa po omrežju. Tako se izognemo slabemu odzivnemu času zaradi ozkih grl in izgube paketov zaradi prezasedenosti omrežnih elementov. Na daljši rok pa se izognemo stroškom, ki so povezani z nadgraditvijo zmogljivosti podatkovnih poti in zmogljivosti omrežnih elementov.

5.2.4 Pomanjkljivosti tehnologije večsmerne oddajanja

Pri naštevanju vseh prednosti tehnologije večsmerne oddajanja se postavi vprašanje, zakaj ni bolj razširjena v primerjavi z drugimi tehnologijami. Glavni razlog je dejstvo, da je večina aplikacij, ki se danes uporabljajo v omrežjih, tipa strežnik-odjemalec. Pri takih aplikacijah gre izključno za komunikacijo enega z enim, zato večsmerno oddajanje ni potrebno.

Druga bistvena omejitev večsmerne oddajanja je, da ne moremo vzpostaviti seje med oddajnikom in sprejemniki. Vzpostavitev seje bi pomenila spremembo v usmerjeno oddajanje. Zaradi tega se pri večsmerne oddajanje ne moremo zanašati na prednosti, ki jih ponuja vzpostavitev seje med dvema oddaljenima točkama (Williamson, 2000a, str. 11-15):

1. nimamo možnosti potrjevanja podatkov, saj strežnik ne ve, kdo vse so sprejemniki, potrditve tisoč ali milijon sprejemnikov pa bi povzročile zasičenje omrežja in oddajnika,
2. izgube podatkov zaradi težav v omrežju ni enostavno ugotoviti in popraviti s ponovno oddajo izgubljenih podatkov ter
3. ni kontrole pretoka, kar pomeni, da lahko oddajnik povzroči zasičenje sprejemnika ali komunikacijske povezave med njima.

5.3 Aplikacije, ki uporabljajo večsmerno oddajanje

Večsmerno oddajanje se pogosto povezuje z aplikacijami za videokonference. Čeprav so bile ravno videokonference ene prvih aplikacij, ki so izkoriščale prednosti večsmerne oddajanja, so to le ena izmed mnogih aplikacij, ki temeljijo na večsmerni oddaji. Posamezne tipe aplikacij delimo glede na način oddajanja podatkovnih paketov, in sicer na:

1. aplikacije, kjer imamo en oddajnik in več sprejemnikov,
2. aplikacije, kjer imamo več oddajnikov in en sprejemnik ter
3. aplikacije, kjer imamo več oddajnikov in več sprejemnikov.

Najenostavnejše aplikacije so tiste, kjer imamo samo en oddajnik. Pri teh aplikacijah gre predvsem za podaljšanje dosega klasičnega razpršenega oddajanja, ker slednje omrežni elementi ne posredujejo naprej z lokalnega segmenta. Tipična uporaba samo enega oddajnika je pri aplikacijah, ki nekaj oznanjajo (omrežni čas, seznam sej, radijski ali televizijski prenosi, oznanjanje dogodkov, borzne informacije, stanje zalog ...) ali nekaj kontrolirajo (senzorji, varnostni mehanizmi, nadzor omrežja ...).

Aplikacije, kjer imamo več oddajnikov in en sprejemnik, najpogosteje temeljijo na modelu zahteva-odgovor (angl. request-response). Primeri teh aplikacij so iskanje virov (predvsem v omrežnih sistemih) in zbiranje podatkov (prenos podatkov na centralno lokacijo, arhiviranje podatkov).

Najzahtevnejše so aplikacije, kjer imamo več sprejemnikov in več oddajnikov. Ponavadi so oddajniki hkrati tudi sprejemniki. Pri teh aplikacijah je največji problem sinhronizacija. To so aplikacije, ki najbolje izkoriščajo vse prednosti usmerjenega oddajanja (Makofske, 2002). Najpogostejši primeri teh aplikacij so avdio- in videokonference. Druge aplikacije so še porazdeljeno in vzporedno procesiranje, posodabljanje podatkov v podatkovnih bazah, učenje na daljavo in interaktivne simulacije.

5.4 Usmerjanje paketov pri večsmernem oddajanju

Protokole za usmerjanje prometa usmerjenega oddajanja srečamo na vsaki točki medmrežja, v vsakem povezanem podatkovnem omrežju, protokoli za usmerjanje prometa večsmerne oddajanja pa niso tako pogosti v omrežjih. Glavni razlog za manjšo razširjenost protokolov usmerjanja večsmerne oddajanja je v njihovem kasnejšem razvoju. Pojavili so se precej kasneje kot protokoli za usmerjanje prometa usmerjenega oddajanja.

V medmrežju imamo več razredov naslovnih prostorov. Prvi trije razredi: A, B in C, so namenjeni usmerjenemu oddajanju. Četrty razred naslovov, razred D, pa je namenjen večsmernemu oddajanju.

Slika 17: Naslovni prostor medmrežnega protokola.

Razred	Naslovni prostor	Namen uporabe
A	1.0.0.0 – 126.255.255.255	Usmerjeno oddajanje
B	128.0.0.0 – 191.255.255.255	Usmerjeno oddajanje
C	192.0.0.0 – 223.255.255.255	Usmerjeno oddajanje
D	224.0.0.0 – 239.255.255.255	Večsmerno oddajanje
E	240.0.0.0 – 255.255.255.255	Naslovi, rezervirani za eksperimentalno uporabo

Vir: prirejeno po Deering, 1989, str. 5.

Pri usmerjanju podatkovnih paketov, ki so naslovljeni na določen naslov iz naslovnega prostora usmerjenega oddajanja, usmerjevalniki na podlagi informacije o omrežjih usmerijo paket proti cilju. Ker je cilj vedno en sam in ker usmerjevalniki s pomočjo usmerjevalnih protokolov izmenjujejo informacijo o tem, kje se odločena omrežja nahajajo, lahko vedno določimo pot, ki vodi proti naslovníku.

Pri večsmernem oddajanju so podatkovni paketi naslovljeni z naslovom skupine, ki jih želi prejemati (slika 18). V tem primeru ne moremo uporabiti protokolov usmerjanja usmerjenega oddajanja, saj se skupinski naslov ne pojavlja v informaciji, ki si jo usmerjevalni protokoli izmenjujejo. Zato so razvili protokole, ki si izmenjujejo informacijo o sprejemnikih prometa večsmerne oddajanja.

Slika 18: Podatkovni paket medmrežnega protokola pri večsmernem oddajanju.

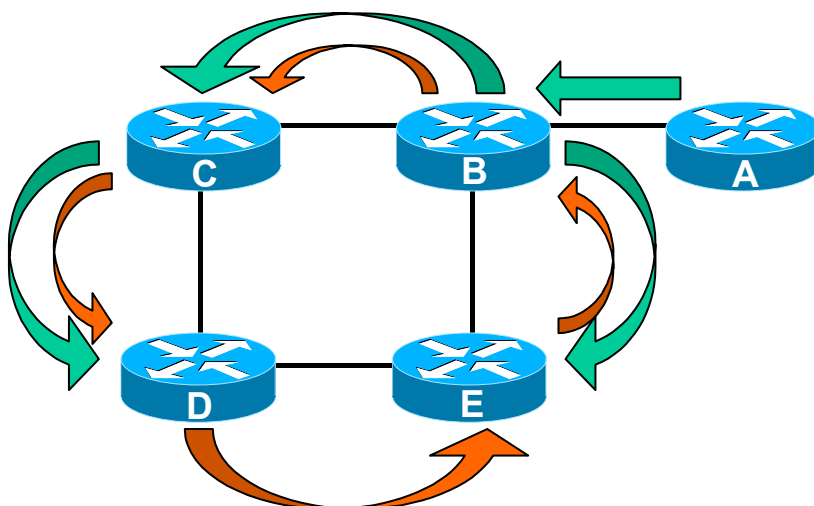
Naslov oddajnika	Naslov skupine sprejemnikov	Podatki
------------------	-----------------------------	---------

Vir: Wright, 1995, str. 341.

5.4.1 Posredovanje paketov glede na njihov izvor

Protokoli večsmernega oddajanja usmerjajo podatkovne pakete proti sprejemnikom glede na naslov skupine večsmernega oddajanja. Če imamo več poti od oddajnika do sprejemnika, lahko pride do posredovalnih zank. Posredovalne zanke povzročijo, da se promet, ki se znajde v njih, neprestano posreduje naprej in lahko hitro pride do zasičenja omrežja ter njegovega razpada.

Slika 19: Posredovalna zanka.



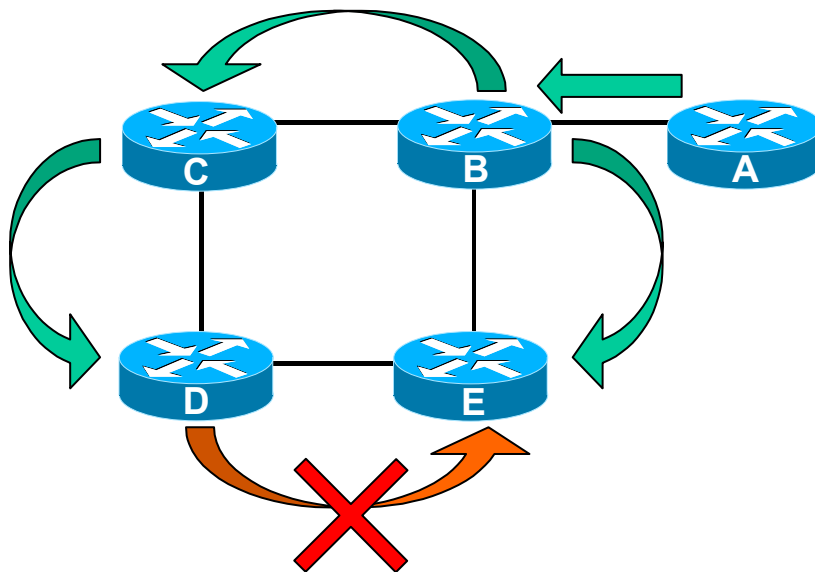
Vir: avtorjevo lastno delo.

Kot primer vzemimo omrežje, kjer imamo priključen oddajnik na omrežje za usmerjevalnikom A in sprejemnike na vseh omrežjih, priključenih na usmerjevalnike B, C, D in E (slika 19). Ko oddajnik pošlje podatkovni paket, naslovljen na skupino sprejemnikov, ga usmerjevalnik A posreduje naprej v omrežje. Usmerjevalnik B paket dobi in ga posreduje sprejemnikom, ki so priključeni na omrežje za njim. Hkrati pa posreduje paket naprej po omrežju do drugih sprejemnikov, torej proti usmerjevalnikoma C in E. Usmerjevalniki C, D in E postopajo enako, vendar promet, ki ga usmerjevalnik E dobi od usmerjevalnika D in ga je usmerjevalnik E že dobil od usmerjevalnika B, povzroči posredovalno zanko. Promet, ki ga je usmerjevalnik B že dobil od usmerjevalnika A in ga je posredoval naprej, sedaj pride do njega iz smeri usmerjevalnika E. Usmerjevalnik

B ga znova posreduje naprej v omrežje. S tem se količina prometa podvaja in povečuje, dokler se komunikacijske povezave v omrežju ne zasitijo.

Pri večsmernem oddajanju lahko v stabilnem omrežju pride do posredovalne zanke. Samo naslov skupine torej ni dovolj za preprečitev posredovalnih zank, ker so sprejemniki lahko kjer koli. Preden usmerjevalnik odpošlje paket večsmernega oddajanja naprej, se mora zato najprej prepričati, da s tem ne bo povzročil posredovalne zanke. To naredi tako, da preveri naslov oddajnika, ki je izvor paketa, in vmesnik, po katerem je usmerjevalnik prejel paket. Če je paket prispel po vmesniku, ki je glede na usmerjevalno tabelo usmerjenega oddajanja najbližji glede na omrežni naslov, iz katerega je paket prispel, potem usmerjevalnik lahko paket posreduje naprej, sicer ga mora zavreči. Ker pakete, ki jih usmerjevalnik ne dobi po optimalni poti, zavrne, problema neskončnih posredovalnih zank več ni (slika 20).

Slika 20: Preprečevanje posredovalnih zank s preverjanjem najkrajše poti do oddajnika.



Vir: avtorjevo lastno delo.

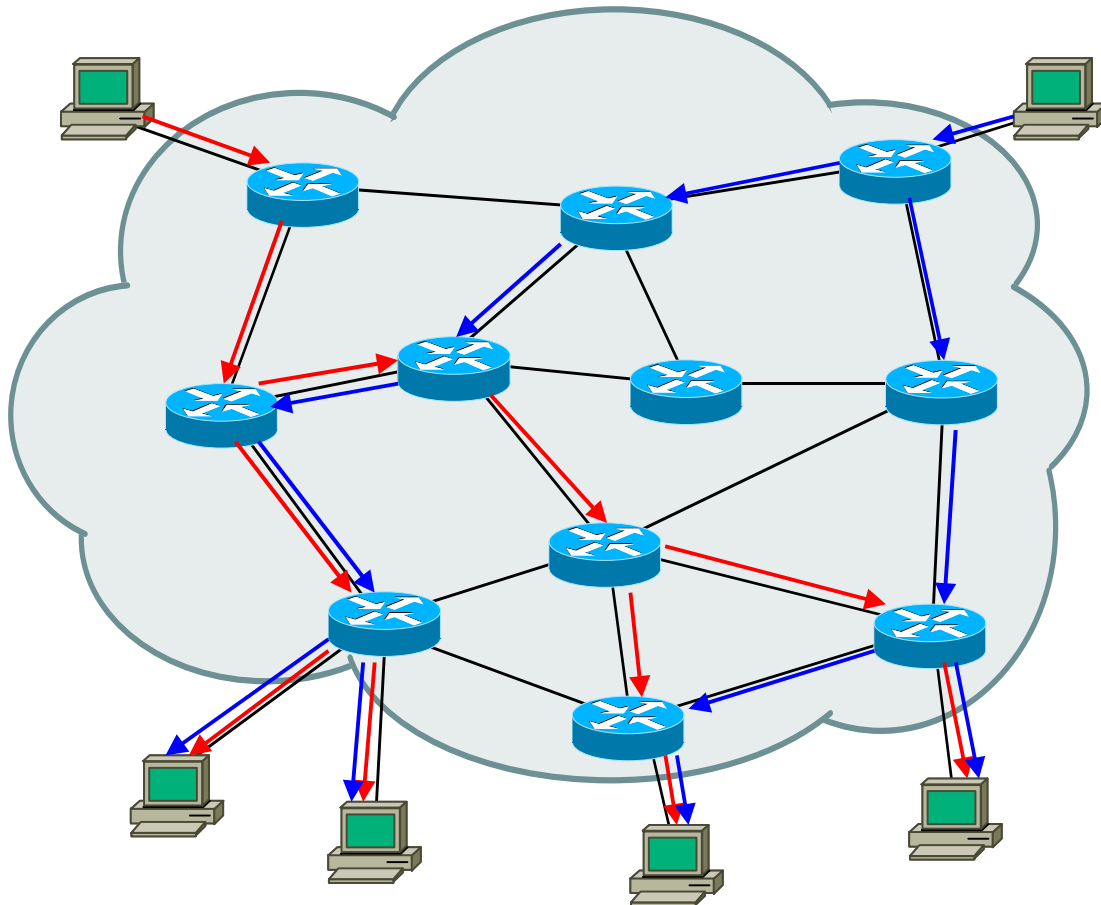
Protokoli za usmerjanje večsmernega oddajanja morajo, preden paket odpošljejo naprej proti sprejemnikom, preveriti, ali je bil paket sprejet po najkrajši poti od oddajnika. Drugi pomembni opravili, ki ju morajo zagotavljati usmerjevalni protokoli za večsmerno oddajanje, sta gradnja in vzdrževanje distribucijskega drevesa med oddajnikom (oddajniki) in sprejemniki.

5.4.2 Gradnja distribucijskega drevesa za večsmerno oddajanje

Glede na način, kako se zgradi distribucijsko drevo za dostavo paketov večsmerne oddajanja, ločimo dve skupini distribucijskih dreves:

1. oddajniška drevesa (angl. source distribution tree) in
2. deljena drevesa (angl. shared distribution tree).

Slika 21: Prikaz poteka prometa večsmerne oddajanja v oddajniškem drevesu.



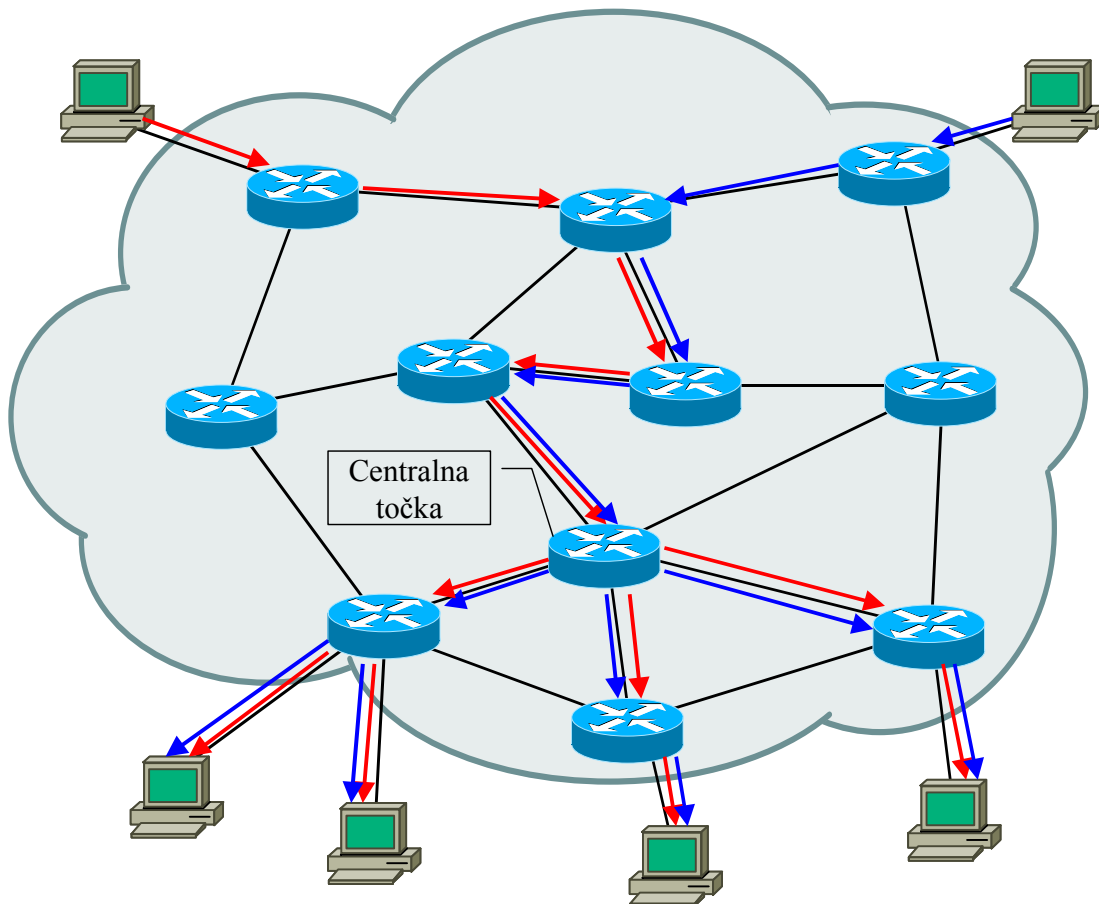
Vir: avtorjevo lastno delo.

V oddajniških drevesih (slika 21) je koren drevesa vedno oddajnik večsmerne oddajanja. Drevo se gradi od oddajnika proti sprejemnikom po najbližji poti, takoj ko začne oddajnik oddajati pakete. Prednost oddajniških dreves je optimalna dostava prometa. Promet potuje od oddajnika do sprejemnikov po najkrajši poti (najhitreje). Slaba stran oddajniških dreves je večja poraba pomnilniškega prostora na usmerjevalnikih, ki morajo hraniti usmerjevalno informacijo za vsak sprejemnik in za vsako skupino večsmerne oddajanja. Če imamo deset oddajnikov, ki oddajajo desetim različnim skupinam, imamo na vsakem usmerjevalniku med oddajnikom in sprejemniki 100 usmerjevalnih stanj. Zaradi funkcij, ki jih usmerjevalniki opravljajo (usmerjanje podatkovnih paketov), nimajo veliko

prostega pomnilniškega prostora in uporaba oddajniških dreves lahko posredno povzroči njihov izpad.

V deljenih drevesih (slika 22) je koren drevesa prestavljen v določeno centralno točko v omrežju. Centralna točka zbira prijave za dostavo večsmernega prometa od vseh usmerjevalnikov, katerim prijave pošljejo sprejemniki. Ko oddajnik začne oddajati promet, ga usmerjevalniki najprej usmerijo proti centralni točki, ki potem porazdeli promet naprej proti prijavljenim sprejemnikom. Prednosti deljenih dreves sta manjše število stanj in s tem manjša poraba pomnilnika, saj imamo samo eno usmerjevalno stanje na dejavno skupino večsmernega oddajanja. Zaradi dodatne poti, ki jo morajo paketi prepotovati v omrežju (najprej jih moramo posredovati centralni točki), se dodatno zakasni, kar je slaba stran uporabe deljenih dreves.

Slika 22: Prikaz poteka prometa večsmernega oddajanja v deljenem drevesu.



Vir: avtorjevo lastno delo.

5.5 Omrežni protokoli za usmerjanje podatkovnih paketov večsmernega oddajanja

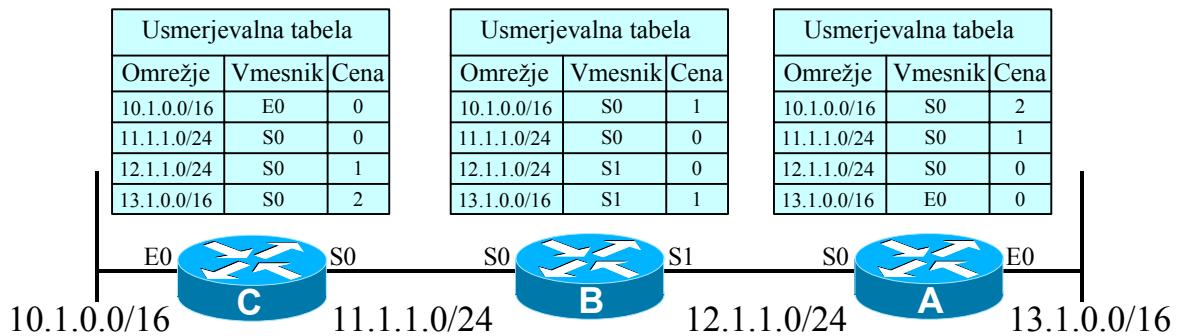
Omrežni protokoli za usmerjanje podatkovnih paketov pri večsmernem oddajanju so se razvili precej pozneje kot protokoli za usmerjanje podatkovnih paketov pri usmerjenem oddajanju. Tako je protokol DVMRP (Distance Vector Multicast Routing Protocol), ki je kot prvi nastal za usmerjanje podatkovnih paketov pri večsmernem oddajanju, nastal po predlogi protokola za usmerjeno oddajanje RIP (Routing Information Protocol) (Malkin, 1999). Tudi MOSPF (Multicast extension to OSPF) je nastal kot nadgradnja protokola OSPF (Open Shortest Path First), ki je protokol za usmerjanje podatkovnih paketov usmerjenega oddajanja. Kasneje so bili razviti še protokoli CBT (Core Based Trees), PIM (Protocol Independent Multicast), ki obstaja v več izvedbah, in MGBP (Multi-protocol extensions to BGP), ki je nadgradnja protokola BGP (Border Gateway Protocol) za učinkovito usmerjanje tudi drugih protokolov poleg IPv4 (Internet Protocol version 4), saj trenutno podpira še večsmerno oddajanje in različico 6 protokola IP (Internet Protocol) (Wittmann, 2000, str. 90).

5.5.1 Protokol DVMRP

Protokol DVMRP zgradi usmerjevalno tabelo tako, da najprej vanjo zapiše vsa lokalno dostopna omrežja. Lokalno omrežje je za usmerjevalnik vedno dosegljivo s ceno 0. To informacijo izmenja s sosednjimi usmerjevalniki, ki tudi uporabljajo protokol DVMRP. Ko dobi informacijo o dosegljivih omrežjih od sosedu, za vsako tako omrežje poveča dobljeno ceno za 1 in informacijo shrani v svojo usmerjevalno tabelo. Označi si tudi, da je najkrajša pot do oddaljenega omrežja skozi vmesnik, ki vodi do sosedu, od katerega je dobil informacijo o omrežju. Če se v tabeli že nahaja informacija o dosegljivosti določenega omrežja, potem shrani samo informacijo z nižjo ceno in ustrezno popravi informacijo o najkrajši poti (Waitzman, 1988, str. 13-22).

Iz slike 23 je razvidno, da so usmerjevalniki že izmenjali informacijo o dosegljivih omrežjih in si ustvarili celotno sliko omrežja. Tako je cena za dosego omrežja 13.1.1.0/16, gledano z usmerjevalnika C, enaka 2 enoti. Protokol DVMRP pri ceni ne upošteva hitrosti komunikacijske povezave ali zakasnitve na njej. Kot ceno vzame le oddaljenost v skokih (koliko vmesnih usmerjevalnikov mora prepotovati paket) do ciljnega omrežja.

Slika 23: Usmerjevalna tabela protokola DVMRP.



Vir: avtorjevo lastno delo.

Ko usmerjevalnik dobi paket večsmernega oddajanja, najprej preveri izvorni naslov in vmesnik, iz katerega je paket prispel. Usmerjevalnik nato preveri usmerjevalno tabelo, da ugotovi, katera pot do omrežja, iz katerega je paket prispel, je najcenejša. Če je tista, po kateri je paket prispel, potem paket posreduje naprej, sicer ga zavrže. Paket se posreduje naprej vsem sosednjim usmerjevalnikom, na katerih teče protokol DVMRP. Če kakšen od usmerjevalnikov ugotovi, da je promet določene skupine večsmernega oddajanja nepotreben (ni nobenega sprejemnika), pošlje posebno sporočilo, katerega prejme usmerjevalnik, ki je prvi na poti proti oddajniku, in preneha pošiljati promet za večsmerno skupino navzdol po distribucijskem drevesu.

Distribucijsko drevo se ustvari šele, ko oddajnik začne oddajati promet. Promet se razširi po celotnem omrežju, ki podpira protokol DVMRP. Če določene veje distribucijskega drevesa prometa ne potrebujejo, usmerjevalniki pošljejo posebna sporočila in promet po teh vejah se ustavi.

Prednosti protokola DVMRP so v njegovi enostavnosti in razširjenosti. Slabosti so omejitve v ceni, saj je maksimalna cena 32 enot, kar protokolu ne omogoča učinkovite uporabe v bolj razvejenih omrežjih. Izmenjava celotne tabele ob vsaki spremembi ali periodično na 60 sekund dodatno obremenjuje komunikacijske povezave in način računanja cene poti ne upošteva lastnosti komunikacijskih povezav med dosegljivimi omrežji. Zaradi vseh teh slabosti protokol DVMRP počasi izginja iz sodobnih omrežij, ki podpirajo večsmerno oddajanje.

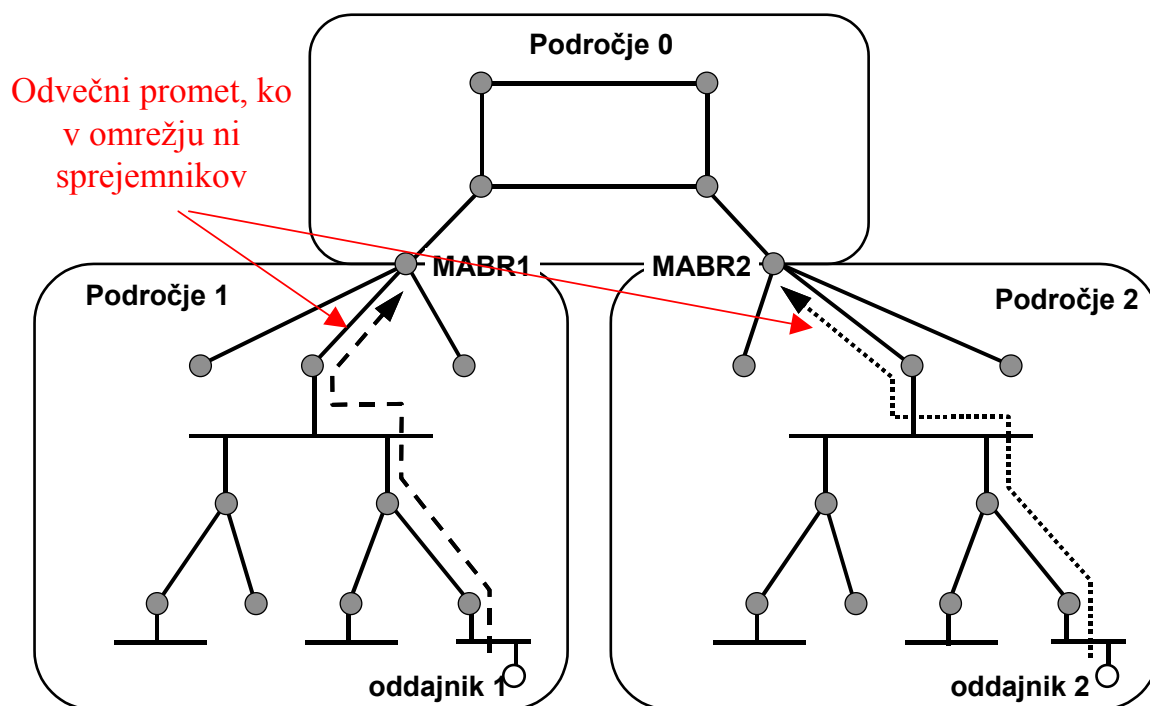
5.5.2 Protokol MOSPF

Protokol MOSPF je nadgradnja usmerjevalnega protokola za usmerjeno oddajanje OSPF. Dodana funkcionalnost omogoča protokolu MOSPF gradnjo distribucijskih dreves za večsmerno oddajanje.

Za razliko od protokola DVMRP, ki pošilja sosedom celotno usmerjevalno tabelo, protokol MOSPF pošilja le podatke o omrežnih segmentih z dejavnimi sprejemniki. Ker se informacija razširi med vsemi usmerjevalniki v omrežju, vsak usmerjevalnik ve, kam usmeriti promet večsmernega oddajanja, ko oddajnik postane dejaven.

Prednost protokola MOSPF je, da za računanje najboljše poti med točkami v omrežju ne uporablja števila skokov, temveč ceno poti, katero izračunamo iz pasovne širine komunikacijske povezave. Protokol MOSPF tako bolj optimalno izbere pot za prenos podatkov v primerjavi s protokolom DVMRP (Moy, 1994, str. 1-2).

Slika 24: Odvečni promet večsmernega oddajanja pri protokolu MOSPF, ko v omrežju ni dejavnih sprejemnikov.



Vir: prirejeno po Williamson, 2000, str. 206.

Ker ima vsak usmerjevalnik vse podatke o komunikacijskih povezavah in njihovih stanjih v omrežju, si sproti, ob vsaki spremembi, lahko izračuna najkrajšo pot med dvema omrežnima segmentoma z uporabo algoritma Dijkstra. Žal je računanje najkrajše poti procesorsko izjemno zahtevno in ga mora usmerjevalnik opraviti ob vsaki spremembi v omrežju. Rešitev je uvedba področij (angl. areas), kjer usmerjevalniki poznajo topologijo samo znotraj področja, v katerem se nahajajo, ter pot, kako priti do centralnega področja, ki povezuje posamezna področja med seboj. S tem lahko bistveno zmanjšamo količino podatkov in preračunavanje topologije ob spremembah, saj smo omejeni na manjše področje (Thomas, 2003).

Vendar zaradi rešitve, ki je sicer zelo primerna za usmerjanje prometa usmerjenega oddajanja, naletimo na težave s prometom večsmernega oddajanja. Ker morajo usmerjevalniki, ki se nahajajo na mejah med področji (angl. Multicast Area Border Router - MABR), vedeti, da imamo v določenem področju dejaven oddajnik, mora mejni usmerjevalnik takega področja biti hkrati tudi sprejemnik za ves promet večsmernega oddajanja (slika 24). Ker se ta promet pojavlja tudi, kadar v omrežju ni niti enega dejavnega sprejemnika (razen MABR), je odvečni promet, ki se v tem primeru pretaka, bistvena pomanjkljivost protokola MOSPF.

Predvsem zaradi pomanjkljivosti protokola MOSPF, ko morajo mejni usmerjevalniki sprejemati ves promet oddajnikov, tudi kadar v omrežju ni nobenega dejavnega sprejemnika, ni dobil velike podpore v produkcijskih omrežjih kljub prednostim, ki jih ponuja v primerjavi s protokolom DVMRP.

5.5.3 Protokol CBT

Protokol CBT se je pojavil kasneje, za protokoloma DVMRP in MOSPF. Vpeljal je novost, ki bistveno poenostavi usmerjevalni protokol za večsmerno oddajanje.

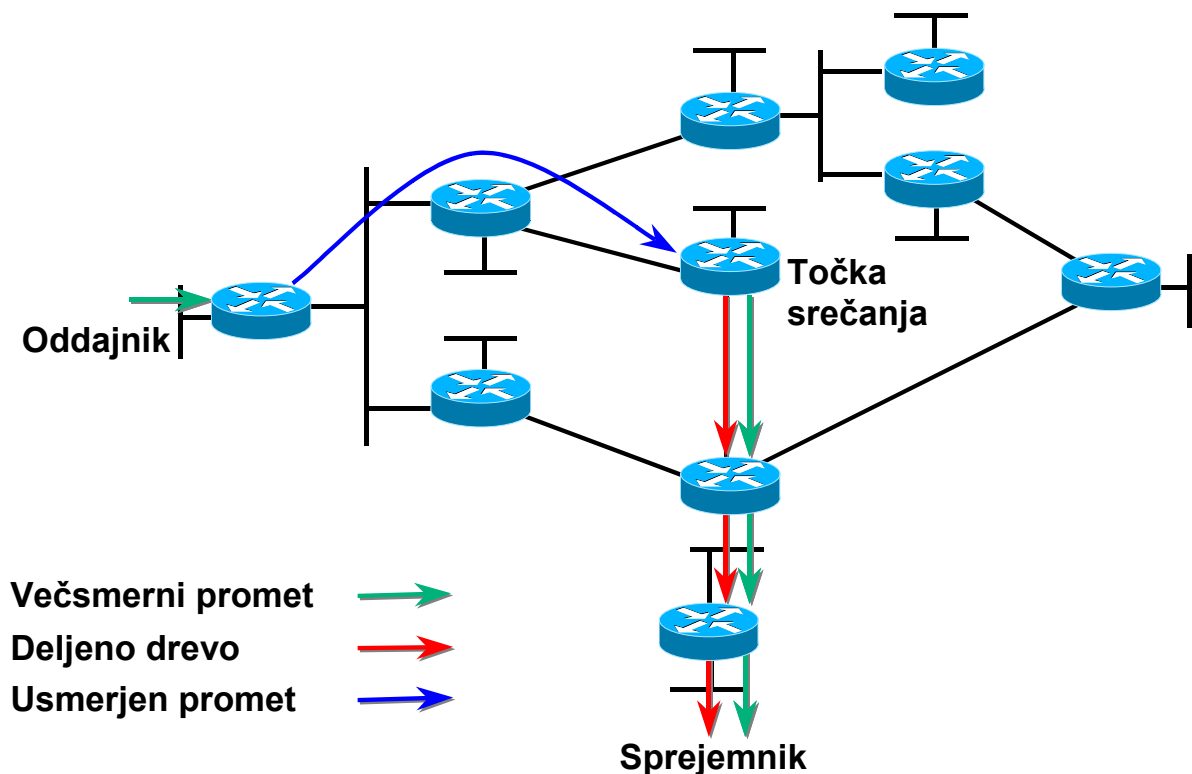
Ker usmerjevalniki že izmenjujejo informacije o omrežjih usmerjenega oddajanja s pomočjo protokolov za usmerjanje usmerjenega oddajanja, ni potrebe, da bi protokoli za usmerjanje večsmernega oddajanja gradili svojo tabelo. Podatke o omrežjih si lahko enostavno "sposodijo" iz tabel usmerjenega oddajanja.

Distribucijsko drevo, ki ga ustvari protokol CBT, je vedno deljeno. To pomeni, da se promet vseh oddajnikov najprej usmeri proti določeni centralni točki v omrežju. Točka se v tem protokolu imenuje jedro (angl. core). Od tod tudi izvira ime protokola, drevesa z jedrom ali krajše CBT (Core Based Trees). Od te točke naprej promet potuje po deljenem drevesu do sprejemnikov (Ballardie, 1997, str. 3-6).

Gradnja deljenega drevesa je pomembna predvsem za sprejemnike. Ko oddajnik še ni dejaven, omrežni elementi nimajo informacije, od kod bo promet prišel, in tako ne morejo zgraditi distribucijskega drevesa. V korenu drevesa bi moral biti oddajnik in če ta še ni dejaven, potem protokol CBT vzame za koren drevesa točko, ki je jedro. Vsi usmerjevalniki imajo informacijo, kje je jedro, zato lahko zgradijo deljeno drevo do le-tega. Ko imamo zgrajeno drevo do jedra za dejavne sprejemnike, je delo končano. Čakamo le še na promet od oddajnika. Ko oddajnik postane dejaven, usmerjevalniki usmerijo promet proti jedru. Od jedra naprej promet steče po distribucijskem drevesu in tako steče celoten promet med oddajnikom in sprejemniki.

Ko začne oddajnik oddajati večsmerni promet, tega prestreže prvi usmerjevalnik na poti proti točki srečanja. Med jedrom in prvim usmerjevalnikom ni zgrajenega distribucijskega drevesa, zato prvi usmerjevalnik vse pakete večsmernega oddajanja vstavi v pakete usmerjenega oddajanja in jih pošlje neposredno proti jedru, ki pakete večsmernega oddajanja izlušči iz paketov usmerjenega oddajanja, in od tod naprej potujejo proti sprejemniku po deljenem drevesu (slika 25).

Slika 25: Potek prometa pri protokolu CBT.



Vir: avtorjevo lastno delo.

Novost "sposojanja" informacij iz usmerjevalne tabele usmerjenega oddajanja, ki jo je vpeljal ravno protokol CBT, je vplivala na razvoj vseh protokolov večsmernega oddajanja, ki so se začeli pojavljati kasneje. Žal sam protokol CBT ni dobil podpore izdelovalcev usmerjevalnikov in se tako danes ne uporablja v produkcijskih omrežjih.

5.5.4 Protokol PIM

Protokol PIM, po ideji iz protokola CBT, jemlje informacijo o dosegljivih omrežjih in cenah poti do njih iz usmerjevalne tabele protokolov za usmerjeno oddajanje. S tem se poenostavi izbira poti in omrežnega vmesnika, ki je najbližje izvoru

prometa. Ta informacija se potem uporabi za preprečevanje usmerjevalnih zank prometa večsmernega oddajanja. Druga prednost protokola PIM je, da se ne omejuje samo na usmerjanje večsmernega prometa za protokol IP, temveč ga lahko uporabimo tudi za protokol IP različice 6, protokol IPX, protokol DECNet in druge.

Protokol PIM je v bistvu družina protokolov, ki jo sestavljajo kar štiri različice:

1. zgoščen način (angl. Dense Mode - PIM-DM),
2. razpršen način (angl. Sparse Mode - PIM-SM),
3. dvosmerni način (angl. Bi-directional PIM) in
4. oddajniško usmerjen način (angl. Source Specific Multicast - PIM-SSM).

5.5.4.1 Zgoščen način delovanja protokola PIM

Pri zgoščenem načinu delovanja protokola PIM izhajamo iz predpostavke, da so sprejemniki na gosto porazdeljeni po omrežju. Na večini omrežnih segmentov naj bi imeli večino časa vsaj en sprejemnik.

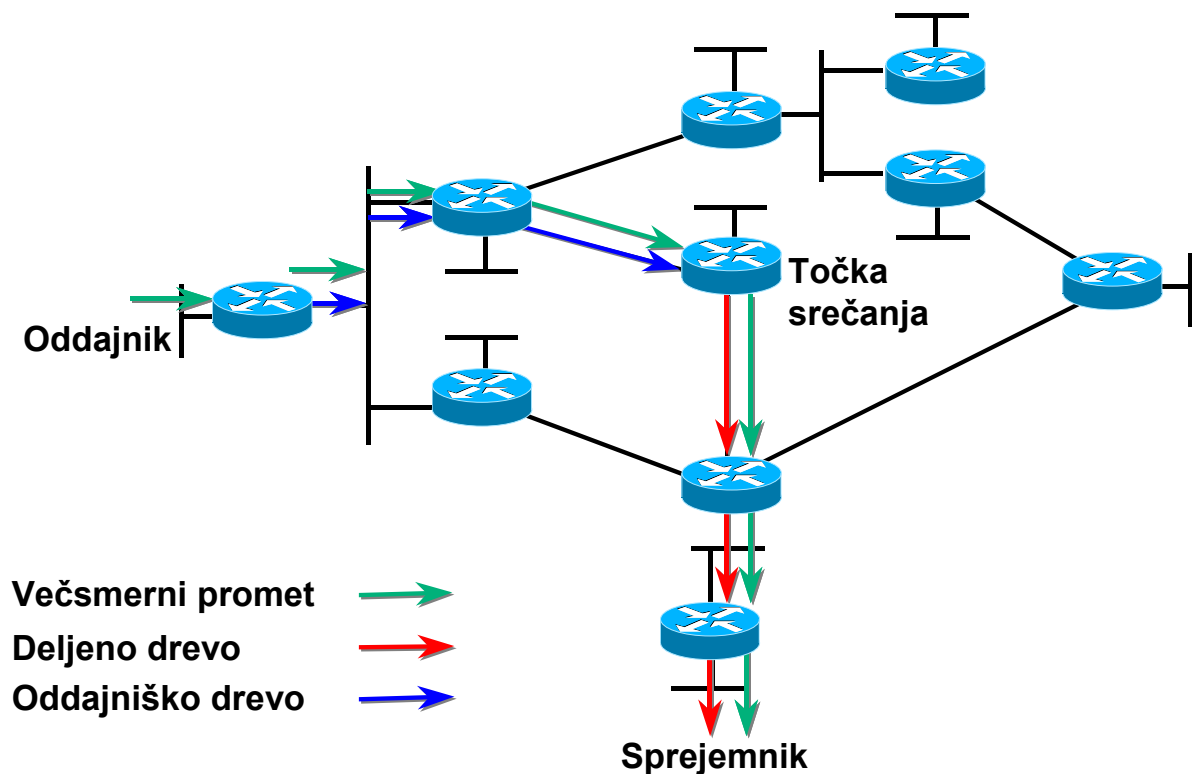
Večsmerni promet, ki ga začne oddajati oddajnik, razpošljemo po vseh segmentih omrežja. Promet tako potuje po najkrajši poti od oddajnika proti sprejemnikom. Če na določenem segmentu ni sprejemnikov, potem usmerjevalnik, priključen na tak segment, pošlje posebno sporočilo nazaj in povzroči, da se na veji, na kateri se nahaja, promet ustavi (Adams Brian, 2004, str. 128).

Na sliki 26 vidimo z zelenimi puščicami označen večsmerni promet oddajnika, ki ga usmerjevalniki posredujejo vsem omrežnim segmentom. Z rdečimi puščicami so označena posebna sporočila za ustavitev posredovanja prometa proti tistim segmentom, kjer ni dejavnih sprejemnikov.

Zgoščen način protokola PIM se izkaže kot zelo enostaven za implementacijo v omrežju, vendar zaradi ne najbolj optimalnega izkoristka komunikacijskih povezav ni priporočljiv za omrežja, kjer imamo le malo omrežnih segmentov s sprejemniki. Ker promet razpošlje po celotnem omrežju, ne glede na to, kje so dejavni sprejemniki, zgoščenega načina ne najdemo pogosto v produkcijskih omrežjih.

Edina mogoča slaba lastnost razpršenega načina delovanja protokola PIM je obstoj točke srečanja. Ker se v tej točki združi promet oddajnikov na poti proti sprejemniku, lahko tu pride do zasičenja omrežja ali do zasičenja same točke srečanja. Zato imamo možnost, da potem ko začne promet prihajati do sprejemnika, zgradimo oddajniško drevo med oddajnikom in sprejemnikom. S tem se izognemo točki srečanja in promet ima manj zakasnitve, saj potuje po najkrajši možni poti. Pozorni moramo biti samo na dodatno količino usmerjevalnih stanj, ki jih s tem pridobimo na usmerjevalnikih.

Slika 27: Delovanje razpršenega načina protokola PIM.



Vir: avtorjevo lastno delo.

Prav zaradi izjemno dobrega obnašanja protokola je razpršen način protokola PIM najpogostejši protokol za usmerjanje prometa večsmerne oddajanja. Možnost povezovanja omrežij različnih ponudnikov je druga dobra lastnost, ki jo s pridom uporabljajo ponudniki medmrežnih storitev za povezovanje medsebojnih omrežij večsmerne oddajanja.

5.5.4.3 Dvosmerni način delovanja protokola PIM

Dvosmerni način delovanja protokola PIM je optimizacija razpršenega načina, ki pride prav, če imamo večsmerno komunikacijo. To pomeni, da ne gre samo za

pošiljanje podatkov oddajnika, temveč lahko tudi sprejemniki začnejo oddajati promet. Pravimo, da so sprejemniki hkrati tudi oddajniki in nasprotno.

Zaradi gradnje oddajniškega drevesa za vsak oddajnik se na usmerjevalnikih pojavlja veliko stanj za usmerjanje prometa večsmernega oddajanja. Z uporabo dvosmernega načina delovanja protokola se vsa ta stanja združijo v eno. Proti točki srečanja promet posredujejo izbrani posredniki (angl. designated forwarders). Izbrani posrednik je vedno tisti, ki ima najcenejšo pot do točke srečanja. S tem je zagotovljeno, da se promet posreduje optimalno, po najkrajši poti, do točke srečanja.

Če natančneje pogledamo oddajniško drevo, opazimo, da promet teče po njem v dve smeri:

1. od oddajnika proti točki srečanja teče promet tega oddajnika in
2. od točke srečanja proti oddajniku, ki je sedaj sprejemnik za promet drugih oddajnikov.

Zato tej različici pravimo dvosmerni način delovanja protokola PIM (Bidirectional PIM, 2003).

Pri dvosmernem načinu delovanja protokola PIM ni preklopa z deljenega drevesa na oddajniško drevo. Ves promet gre vedno skozi skupno točko. Seveda imamo lahko za vsako skupino naslovov večsmernega oddajanja svojo skupno točko, paziti pa moramo, da količina prometa ne preseže zmogljivosti komunikacijskih povezav v njeni okolici.

Dvosmerni način delovanja protokola PIM je izredno podoben delovanju protokola CBT. Oba protokola uporabljata dvosmerno distribucijsko drevo, le da protokol CBT vedno uporablja usmerjeno oddajanje podatkovnih paketov točki srečanja, medtem ko protokol PIM uporablja večsmerno oddajanje tudi za promet med oddajnikom in točko srečanja.

5.5.4.4 Oddajniško usmerjen način delovanja protokola PIM

V opisanih različicah nismo imeli informacije o oddajniku. Zato je bilo treba poslati oddajniški večsmerni promet po celotnem omrežju ali zgraditi deljeno drevo do točke srečanja. Če sprejemnik pozna omrežni naslov oddajnika, lahko takoj zgradimo oddajniško drevo za prenos podatkov med oddajnikom in sprejemnikom. Točka srečanja tako ni več potrebna za prenos začetnega prometa. Protokol se poenostavi in težave, ki so povezane s koncentracijo prometa okoli točke srečanja, odpadejo.

Za oddajniško usmerjen način delovanja protokola PIM moramo, poleg informacije o skupini večsmernega oddajanja, ki ji bo oddajnik pošiljal večsmerni promet, zagotoviti tudi omrežni naslov oddajnika (Bhattacharyya, 2003, str. 4).

5.5.5 Protokol MBGP

Protokoli DVMRP, MOSPF, CBT in PIM so namenjeni za usmerjanje prometa večsmernega oddajanja znotraj domene podjetja, protokol MBGP pa izvira iz protokola, ki je namenjen usmerjanju prometa med domenami predvsem ponudnikov omrežnih storitev. Ker pri svojem delovanju ne izbira poti samo glede na ceno, ampak tudi glede na dodatne parametre, ki jih določi administrator, lahko s tem še dodatno vplivamo na izbiro poti, po katerih bo promet tekkel. Zaradi tega je veliko bolj prilagodljiv, s tem pa tudi zahteven za uporabo (Halabi, 2000, str. 461-470).

Vsaka domena, povezana s protokolom MBGP, mora imeti svojo točko srečanja. Če sprejemniki čakajo na promet oddajnika v eni domeni, oddajnik pa postane dejaven v drugi domeni, promet med njimi ne steče. Mehanizem protokola MSDP (Multicast Source Discovery Protocol) nam omogoča, da se informacija o sprejemnikih in oddajnikih razširi med točkami srečanja. Ker točke srečanja periodično izmenjujejo informacijo o dejavnih oddajnikih, lahko druge točke srečanja zgradijo distribucijsko drevo za svoje sprejemnike (Adams Andrew, 2002, str. 37-91).

Čeprav se podjetja najpogosteje odločajo za uporabo protokolov za usmerjanje prometa večsmernega oddajanja znotraj lastne domene, obstajajo primeri, ko je učinkoviteje uporabiti kombinacijo protokola PIM s protokolom MBGP.

5.6 Zagotavljanje zanesljivosti prenosa podatkov pri večsmernem oddajanju

V četrtem poglavju smo spoznali, da ima protokol UDP samo osnovno transportno funkcijo omrežja:

1. pošiljanje paketov brez vzpostavitve seje,
2. multipleksiranje povezav višjega sloja in
3. ugotavljanje napak.

Ko odpošiljamo pakete UDP, je od omrežja odvisno, ali bodo prispeli do prejemnika. Če v omrežju pride do zasičenosti komunikacijskih povezav, lahko

začnejo omrežni elementi presežne pakete izločevati. Ker protokol UDP med udeležencema komunikacije ne vzpostavi seje, prejemnik ne more ugotoviti, ali je med prenosom prišlo do izgube paketov. Pri vzpostavljeni seji, kot jo uporablja protokol TCP, prejemnik potrди vsak paket pošiljalca oziroma pošiljalca ponavlja pošiljanje paketa, dokler prejemnik ne potrди prejema. Tudi če bi protokol UDP znal ugotoviti, da paketi manjkajo, ne vsebuje mehanizma, s katerim bi lahko znova zahteval odpošiljanje izgubljenih paketov. Prejemnik na višjem (npr. aplikacijskem) sloju lahko ugotovi, da nekaj manjka, vendar mora vedeti, kateri paketi manjkajo, da bi jih lahko vnovič zahteval od pošiljalca.

Zaradi različne strategije posredovanja paketov in zaradi različnih poti, ki jih v omrežju lahko uberejo paketi, ne moremo zagotoviti, da bodo do prejemnika prišli v istem zaporedju, kot so bili oddani. Če jih prejemnik upošteva v istem zaporedju, kot so prišli, lahko sestavi napačno zaporedje. Vedeti moramo, da protokol UDP ni bil namenjen pošiljanju večje količine podatkov (zanesljivemu prenosu večje količine podatkov je namenjen protokol TCP).

Pomanjkljivosti protokola UDP onemogočajo njegovo uporabo za zanesljiv prenos podatkov, ki jih transakcijski sistemi zahtevajo. Zato se za medsebojno komunikacijo porazdeljenih podatkovnih baz uporablja usmerjeno oddajanje z vzpostavitvijo sej protokola TCP. Vzpostavljanje sej v porazdeljenih sistemih, kjer imamo veliko števil strežnikov, lahko že samo po sebi povzroči preobremenitev strežnikov, zaradi obilice informacij, ki se prenašajo med vsemi sistemi, pa lahko povzroči zasičenje komunikacijskih povezav. Možnost za povečanje števila porazdeljenih sistemov nam daje le protokol UDP, ki omogoča pošiljanje podatkov hkrati več sprejemnikom.

Večsmerno oddajanje naslovi podatkovni paket s skupinskim naslovom. Ta opisuje skupino prejemnikov, ki so zahtevali prejemanje podatkov. Če bi za prenos uporabili seje protokola TCP, bi moral oddajnik vzpostaviti sejo z vsakim sprejemnikom. Za vsakega sprejemnika posebej bi moral tudi pošiljati podatkovne pakete. Seveda bi vsak tak paket, razposlan vsakemu sprejemniku posebej, nosil enake podatke. Zato moramo za učinkovit prenos podatkov uporabiti večsmerno oddajanje in s tem protokol UDP. Sicer ne pride do vzpostavitve seje, lahko pa podatkovni paket naslovimo na vse sprejemnike hkrati.

Ker se večsmerno oddajanje uporablja predvsem za avdio- in videokonference, problem izgube paketa ni tako velik. Človeško uho ne razpozna izgube enega ali nekaj paketov, kjer vsak nosi 20 milisekund podatkov. Pri transakcijah pa sta zaporedje in celovitost prenesenih podatkov zelo pomembna. Zato moramo pri komunikaciji med porazdeljenimi podatkovnimi bazami zagotoviti vsaj (Miller, 1998, str. 153):

1. ugotavljanje in ponovno pošiljanje izgubljenih paketov in
2. označevanje vrstnega reda paketov, da lahko sprejemniki sestavijo pravi vrstni red.

Prednosti večsmernega oddajanja izkorišča več različnih aplikacij. Zahteve po zanesljivosti so pri vsaki drugačne. Aplikacije, ki prenašajo govor ali sliko, zahtevajo samo možnost sestave pravilnega vrstnega reda podatkovnih paketov na strani sprejemnika. Aplikacije za prenos datotek zahtevajo poleg oznake vrstnega reda še dostavo vseh podatkovnih paketov, ker v nasprotnem primeru ne morejo sestaviti celotne datoteke. Najzahtevnejše aplikacije pa so finančne in transakcijske. Te ne zahtevajo samo dostave vseh podatkovnih paketov in sestave njihovega pravilnega vrstnega reda, ampak tudi pravočasnost podatkov.

Protokoli, ki se umestijo med aplikacijski sloj in omrežni sloj, poskrbijo za informacijo o zaporedju podatkovnih paketov, njihovo časovno označevanje, sinhronizacijo in označevanje tipa podatkov, ki jih vsebujejo.

Najpogosteje uporabljeni protokol za prenos podatkov avdio in video aplikacij v realnem času je Real-time Transfer Protocol/Real-time Transfer Control Protocol (RTP/RTCP) (Schulzrinne, 2003, str. 5-8). Žal je za transakcijske sisteme le delno uporaben, ker ne zna ob ugotoviti izgube podatkovnih paketov zahtevati njihovega ponovnega pošiljanja. Transakcijski sistemi zahtevajo dostavo vseh podatkov, večinoma pa zahtevajo tudi dostavo podatkov v pravilnem vrstnem redu.

Najbolj znana izmed protokolov, ki omogočajo dostavo vseh paketov večsmernega oddajanja, sta:

1. protokol RMTP (Reliable Multicast Transport Protocol) in
2. protokol PGM (Pragmatic General Multicast).

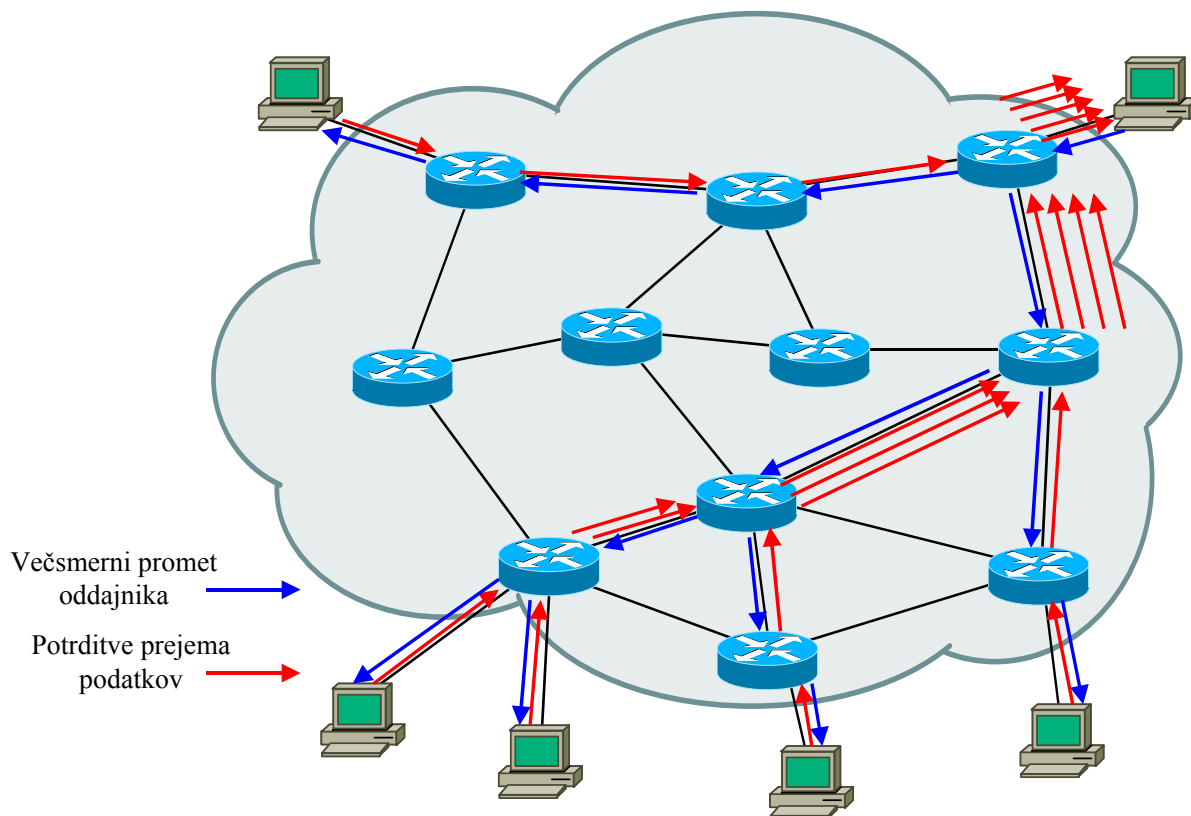
Drugi protokoli, ki zagotavljajo zanesljivo dostavo paketov večsmernega oddajanja, vsebujejo večino značilnosti prvega ali drugega in niso posebej zanimivi za natančnejšo predstavitev.

5.6.1 Protokol RMTP

Protokol RMTP zagotavlja zanesljivo dostavo paketov protokola UDP s potrjevanjem. Sprejemniki potrjujejo vsak prispeli paket s pošiljanjem posebnega potrditvenega paketa. Da se izognemo ogromni količini potrditvenih paketov pri oddajniku (ko vsak sprejemnik pošlje enega), se na vsakem segmentu s sprejemniki izbere en sprejemnik, ki je zadolžen za pošiljanje potrditvenih paketov oddajniku (Sanjoy, 1997). Drugi sprejemniki na segmentu pošiljajo svoje

potrditvene pakete izbranemu sprejemniku. Ta jih zbira in pošilja oddajniku. Oddajnik iz prejetih potrditvenih paketov razbere, ali je treba paket poslati znova ali pa lahko nadaljuje s pošiljanjem naslednjih paketov. S tem sta zagotovljena tudi pravi vrstni red prejetih paketov in ponovno pošiljanje izgubljenih podatkov (slika 28).

Slika 28: Pretok podatkov in potrditev prejema pri uporabi protokola RMTP.



Vir: avtorjevo lastno delo.

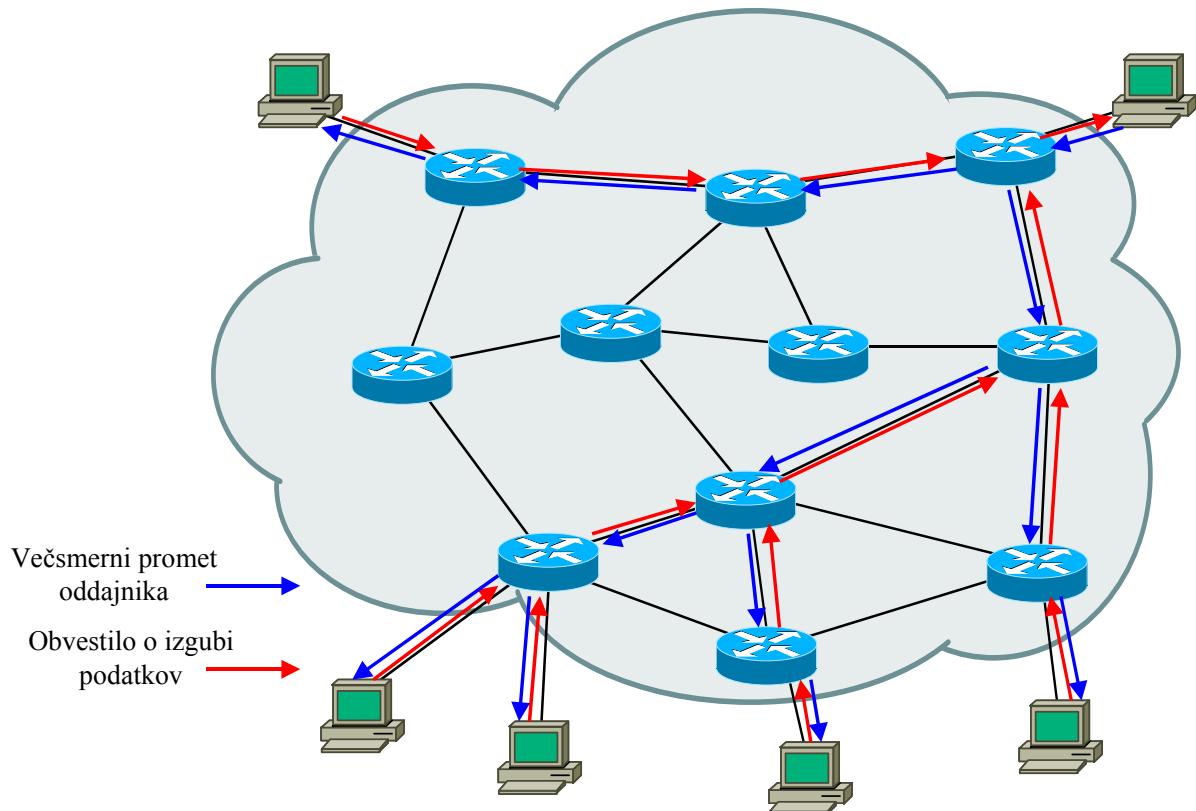
Ker protokol RMTP zahteva pošiljanje potrditvenih paketov, ni najučinkovitejši. Z naraščanjem števila sprejemnikov na različnih segmentih narašča število potrditvenih paketov. Velika količina potrditvenih paketov lahko hitro povzroči zasičenje podatkovnih povezav. Druga pomanjkljivost protokola RMTP je čakanje na potrditev pred pošiljanjem naslednjega paketa, ki negativno vpliva na hitrost komunikacije na velike razdalje.

5.6.2 Protokol PGM

Protokol PGM spada v skupino protokolov, ki zagotavljajo zanesljivo dostavo paketov protokola UDP s potrjevanjem (Speakman, 2001, str. 9-12). Med protokoloma RMTP in PGM obstajata dve bistveni razliki:

1. protokol PGM ne zahteva potrjevanja prejema vsakega paketa, temveč se pošlje le zahteva za ponovno pošiljanje izgubljenih paketov,
2. protokol PGM podpirajo tudi usmerjevalniki, ki prestrežejo zahteve za ponovno pošiljanje izgubljenih paketov, si zapomnijo, kateri podatek je bil zahtevan, in naprej proti oddajniku pošljejo samo eno zahtevo za ta podatek.

Slika 29: Pretok podatkov in obvestil o njihovi izgubi pri uporabi protokola PGM.



Vir: avtorjevo lastno delo.

Če primerjamo oba protokola s stališča omrežja, opazimo, da protokol PGM veliko bolj racionalno izkorišča komunikacijske povezave. Protokol PGM izkorišča sodelovanje omrežnih elementov za učinkovito rešitev problema implozije zahtev po ponovnem pošiljanju podatkov. Namesto da bi oddajnik zasuli z zahtevami po ponovnem pošiljanju izgubljenih paketov, oddajnik dobi kvečjemu eno tako zahtevo (slika 29). V omrežjih pričakujemo več uspešno prenesenih paketov, zato z zahtevo za ponovno pošiljanje izgubljenega paketa manj obremenimo komunikacijske povezave.

5.7 Zagotavljanje varnosti prenosa podatkov pri večsmernem oddajanju

Pri prenosu podatkov prek javnega omrežja obstaja nevarnost, da lahko nekdo med začetno in končno točko komunikacije pregleduje vsebino paketov in jo tudi spremeni. Pri večsmernem prometu je to še lažje, saj je dovolj, da se prijavi kot sprejemnik določene skupine večsmernega prometa in takoj začne prejemati vse podatke. Zato moramo pri prenosu podatkov prek javnih povezav uporabljati varnostne mehanizme za zagotavljanje zaupnosti, neponovljivosti in celovitosti podatkov ter overjanje izvora podatkov.

Varnostni mehanizmi so izvedeni kot (Varnostni mehanizmi, 2002):

- šifriranje (angl. encryption), s katerim zagotavljamo zaupnost podatkov,
- elektronsko podpisovanje (angl. digital signing), s katerim zagotavljamo celovitost transakcij in overjanje izvora,
- zgoščevanje (angl. hashing), ki služi kot podlaga zagotavljanja celovitosti podatkov, in
- številčenje podatkov (angl. sequencing), s katerim zagotavljamo neponovljivost transakcij.

Varnostne mehanizme danes večinoma izvajamo na enem ali več slojih omrežja, najpogosteje pa na:

- sloju podatkovne zveze, kjer namenske naprave na komunikacijskem mediju ščitijo okvire podatkov; ta način je vezan na medij in v mnogih primerih občutljiv na napade "vmesnega človeka" (angl. man-in-the-middle), kjer napadalec, ki vdre v neko komunikacijsko napravo, med dvema medijema lahko prestreže promet, saj v tej točki ni zaščiten,
- omrežnem sloju, kjer usmerjevalniki ali namenske naprave ščitijo pakete podatkov neodvisno od medija, topologije in aplikacije,
- sejnem sloju, kjer v aplikacijo vgrajeni mehanizmi ščitijo podatke tik nad transportnim slojem, in
- aplikacijskem sloju, kjer so vsi mehanizmi vgrajeni v aplikacijo in vezani na posamezno aplikacijo.

Zaradi specifičnosti večsmerne oddaje se najpogosteje uporablja zaščita na aplikacijskem sloju. Oddajnik odpošlje podatkovni paket, vendar pa je vsebina zaščiten in dostopna samo tistim sprejemnikom, ki imajo ustrezen ključ. V porazdeljenem sistemu skupina strežnikov, ki sodeluje pri komunikaciji, ni dinamična, ampak vnaprej znana. Zato lahko uporabimo simetrične algoritme za šifriranje podatkov. Pri simetričnih algoritmih je ključ, ki podatke šifrira, enak ključu, ki podatke odšifrira. Tako je distribucija ključev enostavna in ne

potrebujemo posebnih mehanizmov za dinamično izmenjavo ključev. Poskrbeti moramo, da so ključi dovolj dolgi in da se redno izvaja njihova zamenjava. Poleg tega so algoritmi za šifriranje podatkov z uporabo simetričnih ključev dovolj hitri, da ne zasedejo veliko zmogljivosti na strežniku.

6 ANALIZA MEHANIZMOV ZA UČINKOVITO ZAGOTAVLJANJE PODATKOV V PORAZDELJENIH PODATKOVNIH BAZAH

Na podlagi preučevanja metod za zagotavljanje podatkov v porazdeljenih podatkovnih bazah in protokolov za usmerjanje večsmernega prometa bom v tem poglavju analiziral probleme, ki se pojavljajo v porazdeljenih sistemih brez podvajanja podatkov in s podvajanjem podatkov, ter predlagal učinkovite rešitve z vpeljavo protokolov za večsmerno oddajanje.

6.1 Porazdeljeni sistemi brez podvajanja podatkov

V porazdeljenih sistemih brez podvajanja podatkov imamo več podatkovnih baz porazdeljenih po strežnikih, ki so povezani v omrežje. Vsak podatek se nahaja le v eni izmed podatkovnih baz. S tem prihranimo na pomnilniškem prostoru, potrebnem za hranjenje podatkov. Zaradi hkratnih dostopov do podatkov moramo uporabiti mehanizem za zagotavljanje podatkov. Na podlagi ugotovitev v tretjem poglavju menim, da je v tem primeru najučinkovitejši način za zagotavljanje podatkov metoda zaseganja podatkov v dveh fazah. Lokalni sistemi za upravljanje s podatkovno bazo zagotavlja zaporedje dostopov samo za lokalno bazo, globalni sistem pa poskrbi za porazdelitev operacij transakcij ustreznemu lokalnemu sistemu. Ko se transakcija izvede, globalni sistem pošlje ukaz "zaključiti" vsem sistemom, ki so sodelovali pri izvedbi transakcije.

Ker lokalni sistem za upravljanje s podatkovno bazo obsega vse potrebne podatke, da lahko ugotovi, ali lahko transakcijo izvede na lokalni ravni brez konfliktov, se mu ni treba zanašati na komunikacijo z drugimi porazdeljenimi sistemi. Tako se izognemo težavam, ki se pojavljajo zaradi zakasnitve ali prekinitve komunikacije med porazdeljenimi podatkovnimi bazami.

Večji problem je sprostitev zaseženih podatkov. Pri zaseganju podatkov v dveh fazah sem pokazal, da lokalni sistem v določenih primerih ne more vedeti, kdaj lahko sprostijo zasežene podatke. Porazdeljena transakcija lahko po zadnjem zaseženem podatku na sistemu S_1 še vedno zasega podatke na drugih sistemih. Sistem S_1 v tem primeru ne more sprostiti podatkov za druge transakcije, saj bi s

tem kršil temeljno pravilo zaseganja podatkov v dveh fazah (transakcija po sprostitvi podatka ne sme več zahtevati novih zaseganj).

Glede na moje ugotovitve dosledno zaseganje v dveh fazah reši ta problem, saj mora v tem primeru transakcija ob prvi sprostitvi zaseženega podatka hkrati sprostiti še vse druge zasežene podatke. Problem mrtvih zank rešujemo na globalni ravni. Kateri način bo uporabljen, določata število in lokacija distribuiranih sistemov. Ko imamo malo strežnikov, lahko uporabimo metodo centralnega odkrivanja zank, ko pa število sistemom naraste, sta primernejši metodi hierarhičnega ali porazdeljenega odkrivanja mrtvih zank.

Primer učinkovite poslovne uporabe porazdeljenih sistemov brez podvajanja podatkov so predvsem finančne in bančne aplikacije. Menim, da je pri takih aplikacijah zelo pomembno, da vedno zagotovimo prave vrednosti podatkov. Manjša napaka ali izpad sistema lahko povzročita veliko škodo. To je tudi razlog, da je v takih okoljih lažje zagotoviti sredstva za izvedbo dragih in hitrih komunikacijskih povezav med posameznimi lokacijami. Ker se podmnožica porazdeljenih podatkov nahaja samo v eni podatkovni bazi, se lahko za zagotovitev večje razpoložljivosti podatkov uporabi lokalno podvajanje podatkov z replikacijo. Hitri lokalni segmenti omogočajo hitro izmenjavo podatkov med lokalnimi bazami in tako odpade večina problemov replikacije prek prostranega omrežja.

6.1.1 Komunikacija med porazdeljenimi sistemi brez podvajanja podatkov

Glede na izkušnje, ki jih imam, so porazdeljeni sistemi večinoma urejeni kot gruča strežnikov. Pri tem se vsi nahajajo na istem omrežnem segmentu. Kapaciteta omrežnega segmenta mora biti ustrezno dimenzionirana, ker pričakujemo veliko količino izmenjanih podatkov. Velika količina prometa na omrežnem segmentu je rezultat izmenjave podatkov med samimi porazdeljenimi strežniki zaradi prenosa informacij o transakcijah in zaseganju podatkov ter reševanja konfliktov. Prav tako pa se promet ustvarja zaradi velike količine transakcijskih zahtev uporabnikov. Primerne tehnologije za izvedbo hitrih lokalnih omrežij so:

- tehnologija FastEthernet, ki omogoča hitrosti do 100Mbps (megabitov na sekundo),
- tehnologija GigabitEthernet, ki omogoča hitrosti do 1Gbps (gigabitov na sekundo),
- tehnologija ATM (Asynchronous Transfer Mode), ki omogoča hitrosti do 10Gbps.

Katero bomo izbrali, pa je odvisno od potrebnih zmogljivosti (Saunders, 1996, str. 85-161).

Ker se porazdeljeni sistemi nahajajo na istem omrežnem segmentu, lahko za medsebojno komunikacijo uporabljajo tehnologijo razpršenega oddajanja. Zaradi hitrosti omrežnega segmenta se podatkovni paketi prenašajo zelo hitro in prihajajo v enakem vrstnem redu, kot so bili oddani (Kemme, 2003, str 3-4).

Vendar ima postavitve vseh porazdeljenih sistemov na en segment resne pomanjkljivosti. V primeru izpada segmenta so nedosegljivi vsi podatki. Dostop do omrežnega segmenta je hiter samo za lokalne uporabnike. Uporabniki, ki se nahajajo na oddaljenih omrežnih segmentih, dostopajo do podatkov počasneje (odvisno od hitrosti in zasičenosti komunikacijskih povezav).

Moj predlog za rešitev teh težav je porazdelitev sistemov s postavitvijo strežnikov s podatkovnimi bazami na tiste segmente v omrežju, kjer pričakujemo največ uporabnikov. Zagotoviti moramo, da so tudi podatki, ki so shranjeni na teh strežnikih, tisti, do katerih lokalni uporabniki najpogosteje dostopajo. Zaradi medsebojne komunikacije porazdeljenih strežnikov moramo zagotoviti hitre komunikacijske povezave ali celo posebne povezave, namenjene izključno komunikaciji med porazdeljenimi sistemi. S tem se izognemo težavam, ko zasičenje uporabniških povezav onemogoči kritično komunikacijo med strežniki.

Z večanjem števila porazdeljenih strežnikov lahko že sama komunikacija med njimi povzroči zasičenje komunikacijskih povezav. Zato je pomembno, da je komunikacija realizirana z večsmernim oddajanjem. Za prenos večsmernega oddajanja po omrežju predlagam uporabo dvosmerne različice protokola PIM, saj so vsi porazdeljeni strežniki hkrati sprejemniki podatkov in tudi sami oddajajo podatke o transakcijah. Podatki, ki se prenašajo med porazdeljenimi sistemi, niso podatki samih transakcij, pač pa podatki o zaseženih podatkih. Teh podatkov je precej manj kot podatkov, ki jih moramo prenesti zaradi izvajanja transakcij, zato količina prometa ni tako velika, da bi lahko povzročila zasičenje komunikacijskih povezav v točki srečanja.

6.2 Porazdeljeni sistemi s podvajanjem podatkov

Ker je porazdelitev strežnikov po oddaljenih lokacijah zaradi omejitev pri komunikaciji izjemno draga rešitev in ker je težko zagotoviti lokalnost dostopov do podatkov, se veliko pogosteje uporablja rešitev z replikacijo podatkov. Replicirane podatkovne baze so porazdeljene podatkovne baze, ki vsebujejo kopije podatkov, shranjenih na več lokacijah. Glavni razlog za replikacijo podatkov je omogočiti njihovo dostopnost v primeru izpada ene ali več podatkovnih baz. Druga razloga sta izboljšanje učinkovitosti in odzivnosti sistema s postavitvijo (vseh) podatkov blizu uporabnikom ter razbremenitev omrežja.

Glede na pogostost replikacije podatkov ločimo dve strategiji, ki zagotavljata podatke v repliciranih bazah (Grey, 1996, str. 1):

1. pohlepna replikacija (angl. eager replication) in
2. zakasnjena replikacija (angl. lazy replication).

6.2.1 Pohlepna replikacija

Pohlepna replikacija poskuša ves čas zagotoviti celovitost vseh repliciranih podatkovnih baz. Takoj ko transakcija spremeni določen podatek, se transakcija dostavi v izvajanje drugim bazam. Transakcija se pri tem zakasni zaradi dveh posrednih učinkov:

1. transakcija mora počakati, da se sprememba izvede na vseh repliciranih podatkovnih bazah, in
2. tudi druge replicirane podatkovne baze izvajajo lokalne transakcije in jih dostavljajo v obdelavo drugim bazam ter s tem povzročajo zasedenost tudi lokalne baze.

Preden transakcija nadaljuje z delom, se mora prenesti in izvesti na vseh drugih sistemih. V porazdeljenih sistemih je to huda omejitev. Zakasnitev pri komunikaciji v omrežju povzročajo zakasnitve izvajanja transakcij, še večje težave pa se pojavijo ob izpadu enega ali več sistemov. Z naraščanjem števila porazdeljenih sistemov narašča tudi verjetnost mrtvih zank, in sicer s tretjo potenco, z naraščanjem transakcij pa s peto potenco (Holliday, 1999a, str. 1).

Ravno zaradi teh pomanjkljivosti se pohlepna replikacija v osnovni obliki zelo redko uporablja. Pogosteje pohlepno replikacijo uporabljamo v kombinaciji z metodo večine, kjer je dovolj, da se transakcija prenese samo na določeno (večinsko) podmnožico vseh sistemov, ki sodelujejo v replikaciji. S sistemom

večine potem zagotovimo, da se dostop do spremenjenih podatkov dovoli samo, če pri tem ne bo prišlo do konfliktna situacije.

Pohlepno replikacijo uporabljamo pri časovno občutljivih poslovnih procesih, kjer je nujno, da so vsi podatki razpoložljivi in so vrednosti v vseh porazdeljenih podatkovnih bazah večinoma enake. Ker vsi uporabniki dostopajo do podatkov v lokalni porazdeljeni podatkovni bazi, so dostopni časi do teh podatkov zelo kratki. Čeprav je ta rešitev cenejša in zanesljivejša od sistemov, ki ne uporabljajo podvajanja podatkov, pa lahko pride v primeru izpada komunikacijskih povezav med porazdeljenimi podatkovnimi bazami do nekonsistence podatkov. Zato moramo z ustreznim načrtovanjem omrežja (vzpostavitev vzporednih poti) zagotoviti, da je verjetnost popolnega izpada kar se da majhna. To pa v končni fazi podraži rešitev.

6.2.2 Zakasnjena replikacija

V nasprotju s pohlepno replikacijo, ki poskuša delne rezultate transakcije takoj izvesti na vseh repliciranih sistemih, zakasnjena replikacija najprej izvede transakcijo in shrani vse spremembe podatkov v lokalno podatkovno bazo ter šele nato transakcijo prenese in izvede na drugih repliciranih sistemih (slika 30).

Slika 30: Primerjava izvedbe transakcije v centralizirani podatkovni bazi ter pri pohlepni in zakasneni replikaciji.

Centralizirana podatkovna baza	Pohlepna replikacija	Zakasnjena replikacija
	Baza1 Baza2 Baza3	Baza1 Baza2 Baza3
zapiši A	zapiši A	zapiši A
zapiši B	zapiši B	zapiši B
zapiši C	zapiši C	zapiši C
zaključí	zaključí	zaključí

Vir: Gray, 1996, str. 2.

Zakasnjena replikacija omili stroške, povezane z implementacijo pohlepne replikacije. Čeprav ni primerna za časovno občutljive poslovne procese, ponuja veliko prednosti zaradi enostavnejše in predvsem veliko cenejše izvedbe. Ne zahteva velike zmogljivosti komunikacijskih povezav, ker v primeru zasičenja lahko prestavi izvedbo transakcije na termin, ko je obremenitev manjša. Primer učinkovite poslovne uporabe porazdeljenih sistemov z zakasnjeno replikacijo so različne imeniške aplikacije. Pri tem gre lahko za sezname poslovnih partnerjev, ki vsebujejo tudi podatke, pomembne za poslovne procese (vrednost posla, popusti, količina prodanega ali kupljenega materiala), ali kakšne druge podatkovne baze, katerih podatki morajo biti dostopni vsem lokacijam.

Pri zakasnjeni replikaciji poznamo dva načina, ki določata, kako lahko transakcije izvajajo spremembe podatkov (Gray, 1996a, str. 3-4):

1. zakasnjena skupinska replikacija (angl. lazy group replication) in
2. zakasnjena centralizirana replikacija (angl. lazy master replication).

6.2.2.1 Zakasnjena skupinska replikacija

Zakasnjena skupinska replikacija dovoljuje spremembo podatka v kateri koli replicirani podatkovni bazi. Ko se transakcija zaključi, se pošlje v izvedbo še na druge replicirane sisteme. Težave se pojavijo, ko imamo veliko izpadov komunikacije med repliciranimi podatkovnimi bazami za daljši čas. Ker ne moremo ločiti med izpadom omrežja in dejanskim izpadom podatkovne baze, lahko pride do sprememb podatkov na obeh sistemih, razdeljenih z izpadom komunikacijskih povezav.

Vendar zakasnjena skupinska replikacija ni nujno nerešljivo slaba rešitev. V sistemih, kjer imamo veliko mobilnih sistemov, je ta način replikacije zelo uporaben. Mobilni sistemi imajo dostop do omrežja relativno malo časa. Ker ne moremo zahtevati, da so med spremembami podatkov povezani v omrežje, podatkov ne bi mogli spreminjati. Tu pride v poštev samo zakasnjena skupinska replikacija, ki dovoljuje spremembe tudi takrat, ko sistemi niso povezani. Transakcijske spremembe podatkov se prevedejo na operacije nad podatki in se izmenjujejo z repliciranimi podatkovnimi bazami takrat, ko se povežejo v omrežje. Žal pa je ta rešitev primerna samo za relativno majhno število sistemov, ker verjetnost konflikta narašča s kvadratom njihovega števila.

6.2.2.2 Zakasnjena centralizirana replikacija

Pri zakasnjeni centralizirani replikaciji se poskušamo izogniti težavam z lokalnimi spremembami podatkov, medtem ko sistemi ne morejo komunicirati med seboj, na način lastništva podatkov. Samo lastnik podatka lahko shrani njegovo spremembo vrednosti, ker ve, katera je trenutno prava vrednost podatka. Spremembe mora izvesti najprej lastnik in nato se lahko spremembe pošljejo drugim repliciranim podatkovnim bazam.

Ko transakcija zahteva spremembo določenega podatka, se pošlje zahteva za spremembo lastniku podatka. S tem je zagotovljena celovitost spremembe. Sistem, ki izvaja transakcijo, mora komunicirati s sistemom, ki je lastnik podatka, sicer izvedba transakcije ni mogoča. Zagotavljanje podatkov je najprimerneje izvedeno na način zaklepanja v dveh fazah.

Zakasnjena centralizirana replikacija ni najprimernejša za mobilne sisteme, ki so samo omejeno kratko količino časa priklopljeni na omrežje, zato pa zagotavlja celovite podatke in ustvari manj prometa v omrežju med izvajanjem transakcij.

6.2.3 Dvonivojska replikacija

Dvonivojska replikacija poskuša združiti dobre lastnosti pohlepne in zakasnjene replikacije. Zato podpira obe vrsti sistemov, mobilne in centralne. Ker so mobilni sistemi lahko dlje časa brez povezave in posodobitve podatkov, je na njih lahko izvesti samo začasne spremembe. Zavedati se morajo, da so začasne spremembe lahko sprejete, lahko pa so tudi zavrnjene zaradi konflikta z drugimi transakcijami. Začasne spremembe postanejo trajne, ko se mobilni sistem priključi na omrežje in izmenja podatke z drugimi sistemi.

Mobilni sistem je lahko tudi lastnik določene manjše podmnožice podatkov. Spremembe teh podatkov so, ne glede na to, ali je mobilni sistem priključen na omrežje ali ne, vedno trajne. Dovolj natančna izbira podmnožice lastnih podatkov mobilnega sistema lahko bistveno zmanjša število zavrnjenih transakcij.

Centralni sistemi, ki torej niso mobilni, se obnašajo enako kot pri zakasnjeni centralni replikaciji. Določeni podatki so jim lastni in nad njimi lahko izvajajo trajne spremembe. Do drugih podatkov dostopajo prek omrežja. Če je oddaljeni sistem dosegljiv, potem lahko izvedejo trajno transakcijo, sicer izvedejo začasno spremembo lokalnih podatkov. Tu spremembe počakajo, da se vzpostavi

komunikacijska povezava in poskuša izvesti transakcija na oddaljenem sistemu, ki je lastnik podatkov.

Pravilno delovanje mobilnih sistemov zagotavljajo pravila, ki določajo, kdaj in kako se izvede posodobljanje podatkov ob replikaciji. Ko se mobilni sistem poveže s katerim od centralnih sistemov, potem mobilni sistem (Gray, 1996b, str. 5):

1. najprej zavrže različice začasnih podatkov, ker bo ob replikaciji dobil prave podatke s centralnih sistemov,
2. pošlje v izvedbo transakcije, ki so spreminjale podatke, katerih lastnik je mobilni sistem,
3. pošlje vse začasne transakcije v izvedbo sistemom, ki so lastniki podatkov, v enakem zaporedju, kot so prispele v obdelavo na mobilnem sistemu,
4. sprejme vse spremembe podatkov s centralnih sistemov in
5. sprejme podatke o uspeh in neuspeh lastnih transakcijah, ki jih je v koraku 3 poslal centralnim sistemom v izvedbo.

Ko se mobilni sistem priključi na centralnega, centralni sistem izvede naslednje korake:

1. pošlje zakasnjene transakcije v izvedbo mobilnemu sistemu,
2. sprejme v izvedbo zakasnjene transakcije, ki so spreminjale tiste podatke, katerih lastnik je mobilni sistem,
3. sprejme v izvedbo začasne transakcije v istem vrstnem redu, kot so prispele v obdelavo na mobilnem sistemu,
4. ko centralni sistem izvede transakcije, jih pošlje na način zakasnjene replikacije drugim porazdeljenim sistemom,
5. ko so obdelane vse prejete transakcije, je stanje mobilnega sistema spet celovito.

Transakcije, ki se prenašajo, so samo tiste, ki spreminjajo podatke. Zaželeno je, da so transakcije sestavljene na tak način, da so komutativne. To pomeni, da lahko poljubno spremenimo vrstni red transakcij, vendar bodo rezultati izvedbe transakcij v poljubnem vrstnem redu še vedno enaki.

Večja pomanjkljivost modela dvonivojske replikacije so zastareli podatki na mobilnih sistemih. Ker mobilni sistemi izvajajo replikacijo bolj poredko, je mogoče, da so podatki, do katerih dostopamo, že zastareli. Tega se moramo zavedati pri načrtovanju programske opreme in podatkov, ki morajo biti sposobni odpraviti težave, do katerih lahko pride v tem primeru.

Vzemimo primer, da trgovski potniki replicirajo lokalno porazdeljeno podatkovno bazo enkrat na dan. Vsi posli, ki so sklenjeni čez dan, so začasne transakcije. Čeprav je bilo zjutraj v skladišču 100 kosov blaga, ni nujno, da je v trenutku

prodaje sredi dne blago še vedno na zalogi. Kolikšna je dejanska zaloga, izvemo šele ob koncu dneva, ko vsi trgovski potniki sprožijo replikacijo. Problem, ko je prodanih več izdelkov, kot jih je trenutno na zalogi, rešujemo zunaj domene podatkovnih baz, s podaljšanimi dobavnimi roki ali uporabo rezerv.

6.2.4 Komunikacija med porazdeljenimi sistemi s podvajanjem podatkov

Pri vseh metodah replikacije gre za prenose velikih količin podatkov med strežniki. Optimalen način realizacije prenosa podatkov med njimi je s pomočjo večsmernega oddajanja. Pri uporabi večsmernega oddajanja se podatki ne izmenjujejo samo med dvema lokacijama naenkrat, ampak lahko en strežnik prenese spremembo podatka hkrati na vse porazdeljene strežnike. Namesto N prenesenih sporočil se prenese samo eno, kar je velik prihranek v porazdeljenih sistemih z velikim številom sistemov.

Za varnost komunikacije, še posebej, če ta poteka po javnih povezavah, je smiselno poskrbeti s šifriranjem podatkov. Na podlagi izkušenj menim, da lahko šifriranje najbolj optimalno realiziramo z eno izmed metod, ki uporablja simetrične ključe, ker se komunikacija izvaja med znanimi strežniki. Poleg varnosti moramo poskrbeti tudi za zanesljivost komunikacije večsmernega oddajanja. Pri replikaciji je ponavadi dovolj, če zagotovimo samo zanesljivo dostavo podatkov. Ko je treba zagotoviti tudi pravilen vrstni red oddanih podatkov, lahko uporabimo mehanizme za zagotavljanje pravilnega vrstnega reda podatkov.

Globalni konflikti se rešujejo avtomatsko, kjer ponavadi prevlada časovno prva ali zadnja sprememba, ali ročno kar zahteva poseg administratorja. Poleg tehnike časovnega označevanja lahko za zagotavljanje podatkov uporabimo tudi globalno zaklepanje v dveh fazah.

Kateri protokol za usmerjanje večsmernega oddajanja bomo uporabili, je odvisno od tega, ali gre za replikacijo, kjer dovolimo spremembo podatkov na kateri koli replicirani podatkovni bazi (skupinska replikacija) ali pa dovolimo spremembe podatkov na samo določenih repliciranih podatkovnih bazah (centralna replikacija) (Rankins, 2002, str. 77-85).

Ko je dovoljena sprememba podatkov na kateri koli replicirani podatkovni bazi, potem komunikacija poteka dvosmerno med vsemi bazami. Menim, da je dvosmerno delovanje protokola PIM za usmerjanje prometa večsmernega oddajanja najustreznejša rešitev. Dvosmerno distribucijsko drevo se zgradi med vsemi repliciranimi podatkovnimi bazami in točko srečanja. Večsmerni promet tako

potuje od posamezne replicirane podatkovne baze do točke srečanja in od tod do vseh drugih repliciranih podatkovnih baz.

Če količina prometa presega zmogljivosti omrežja v točki srečanja in povečanje zmogljivosti komunikacijskih povezav ni opravičljivo, se po mojih analizah bolje obnese razpršen način delovanja protokola PIM. Enosmerno distribucijsko drevo s korenem v vsaki replicirani podatkovni bazi dostavlja podatke z večsmernim oddajanjem vsem drugim repliciranim podatkovnim bazam. Ker potrebujemo eno distribucijsko drevo na replicirano podatkovno bazo, se ustvari večja količina stanj za usmerjanje večsmerne prometa na vmesnih usmerjevalnikih kot pri dvosmernem distribucijskem drevesu. Prednost rešitve z razpršenim delovanjem protokola PIM je gradnja oddajniških distribucijskih dreves, kar pomeni, da promet poteka po optimalni poti z najmanjšo mogočo zakasnitvijo v omrežju. Slaba stran rešitve pa je povečana poraba pomnilniškega prostora na usmerjevalnikih.

7 SKLEP

Zagotavljanje hitrega in neprekinjenega dostopa do podatkov v sodobnih podjetjih postaja kritična dejavnost. Podjetja vlagajo veliko denarja za zagotovitev nemotenega poslovanja, saj je vsak izpad strežnika ali komunikacijskih povezav povezan z izgubo. Ta se lahko kaže kot izguba človeškega življenja, ko gre za dostop do medicinskih podatkov pri nujni medicinski pomoči, izguba stranke zaradi napačnega naročila, podaljšani dobavni čas zaradi zakasnitve v proizvodnji ipd.

Podatkovne baze so centralni sistem podjetja, ki v veliki meri omogoča poslovanje. Ko so vsi podatki shranjeni na enem mestu, je zagotovljen enostaven dostop do njih. S porazdelitvijo podatkovnih baz in podatkov po različnih oddaljenih lokacijah podjetja dosežemo večjo hitrost dostopa do podatkov in povečamo stopnjo razpoložljivosti podatkov. Podatki, ki se nahajajo v več podatkovnih bazah, so dosegljivi tudi v primeru, ko ena izmed njih postane nedostopna zaradi izpada komunikacijskih povezav ali izpada strežnika.

Žal je število porazdeljenih sistemov močno omejeno navzgor. Z naraščanjem števila porazdeljenih sistemov se hitro povečuje verjetnost mrtvih zank, prav tako pa se hitro povečuje tudi verjetnost zavrnitve transakcij zaradi konfliktov med njimi. Za reševanje problemov razširljivosti sistema se zato ne uporablja tradicionalna replikacija podatkovnih baz, kjer se operacija spremembe podatka izvede na vseh repliciranih sistemih hkrati. Pogosteje uporabljeni so sistemi, ki uporabljajo metodo večine, ali sistemi, ki uporabljajo metodo zakasnjene replikacije. Ti ne zahtevajo stalne prisotnosti vseh sistemov, ki sodelujejo pri replikaciji.

Zagotavljanje podatkov v porazdeljenih sistemih je bistveno težje kot v centraliziranih sistemih. Potrebna je koordinacija transakcij na globalni ravni. Pri tem je zelo pomembna hitra medsebojna komunikacija med porazdeljenimi sistemi. Problem usmerjenega oddajanja je podvajanje podatkov pri komunikaciji vsakega z vsakim, saj mora sistem, na katerem je transakcija opravila spremembo, to operacijo poslati vsem drugim porazdeljenim sistemom v izvedbo. Zato porazdeljeni sistemi izrabljajo prednost večsmernega oddajanja, kjer z enim samim paketom lahko naslovijo celotno skupino sistemov. Zmanjšana količina podatkov, ki se prenašajo po omrežju, prinese boljše odzivne čase, kar je zelo pomembno za hitro izvajanje transakcij.

Večsmerno oddajanje na večje razdalje brez posebnih omrežnih protokolov ni mogoče. Usmerjevalniki ponavadi prepuščajo in usmerjajo samo podatke usmerjenega oddajanja. Za implementacijo večsmernega oddajanja v omrežju je tako treba načrtno omogočiti protokol za usmerjanje podatkov večsmernega oddajanja. Izbira protokolov večsmernega oddajanja je presenetljivo bogata. Najpogosteje se uporablja razpršen način delovanja protokola PIM. Glede na potrebe porazdeljenih podatkovnih baz pa v primeru manjšega števila porazdeljenih sistemov lahko učinkovito uporabimo tudi dvosmerni način delovanja protokola PIM.

Ker večsmerno oddajanje uporablja protokol UDP, ki nudi le osnovni prenos podatkov od oddajnika do sprejemnika, moramo za prenos podatkov o transakcijah uporabiti katerega izmed protokolov, ki zagotavljajo zanesljivost prenesene informacije. Ob izgubi podatka sprejemnik zazna manjkajoče zaporedje paketov in takoj sporoči oddajniku, naj vnovič pošlje izgubljene podatke, s čimer zagotovi celovitost porazdeljenih podatkovnih baz.

Z uporabo javnih omrežij za prenos podatkov (npr. medmrežja) podjetja zmanjšujejo stroške za komunikacijsko infrastrukturo. Prenos podatkov po takih omrežjih je cenejši, vendar izgubimo zaupnost prenesenih podatkov, ki jo nudi lastno zasebno omrežje. Zato moramo zagotoviti ne samo varnosti prenesenih podatkov, ampak tudi overjanje pošiljatelja in številčenje podatkov, da preprečimo zlorabe iz tujih sistemov.

Pri izboru sistema porazdeljenih podatkovnih baz moramo upoštevati specifične zahteve, ki jih zahteva način shranjevanja in dostopa do podatkov. Samo tako lahko izberemo rešitev, ki nudi zanesljivo, razširljivo, robustno, dosegljivo in hitro porazdeljeno podatkovno bazo, ki v vsakem trenutku zagotavlja prave podatke za podporo poslovanja.

8 LITERATURA

1. Adams Andrew, Nicholas Jonathan, Siadak William: Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised), [URL: <ftp://ftp.rfc-editor.org/in-notes/internet-drafts/draft-ietf-pim-dm-new-v2-04.txt>], 2004. 54 str.
2. Adams Brian et al.: Interdomain Multicast Solutions Guide. Indianapolis: Cisco Press, 2002. 304 str.
3. Ballardie A.: Core Based Trees (CBT version 2) Multicast Routing: Protocol Specification, [URL: <ftp://ftp.rfc-editor.org/in-notes/rfc2189.txt>], 1997. 23 str.
4. Bell David, Grimson Jane: Distributed Database Systems. Wokingham: Addison-Wesley Publishing Company, 1992. 410 str.
5. Bernstein Philip A., Hadzilacos Vassos, Goodman Nathan: Concurrency Control and Recovery in Database Systems. Wokingham: Addison-Wesley Publishing Company, 1987. 363 str.
6. Bezjak Mateja: Baza znanja za podporo uporabnikom. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2002. 86 str.
7. Bhattacharyya S. Ed: An Overview of Source-Specific Multicast (SSM), [URL: <ftp://ftp.rfc-editor.org/in-notes/rfc3569.txt>], 2003. 14 str.
8. Bray H. Olin: Distributed Database Management Systems. Lexington: Lexington Books, 1982. 164 str.
9. Cattell Roderic, Geoffrey Galton: Object data management: object-oriented and extended relational database systems. Wokingham: Addison-Wesley Publishing Company, 1991. 318 str.
10. Comer E. Douglas, Stevens L. David: Internetworking With TCP/IP Vol II: Design, Implementation, and Internals. London: Prentice-Hall, 1991. 532 str.
11. Deering S.: Host Extensions for IP Multicasting, [URL: <http://www.ietf.org/rfc/rfc1112.txt?number=1112>], 1989. 17 str.
12. Estrin D. et al.: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, [URL: <ftp://ftp.rfc-editor.org/in-notes/rfc2362.txt>], 1998. 66 str.
13. Garcia-Molina Hector, Ullman D. Jeffrey, Widom Jennifer: Database systems: The complete book. London: Prentice-Hall, 2001. 1100 str.
14. Gray Jim et al.: The Dangers of Replication and a Solution, [URL: <http://citeseer.nj.nec.com/cache/papers/cs/1999/http:zSzzSzwww.research.microsoft.comzSz~GrayzSzreplicas.pdf/gray96danger.pdf>], 1996. 12 str.
15. Halabi Sam, McPherson Danny: Internet Routing Architectures, Second Edition. Indianapolis: Cisco Press, 2000. 498 str.

16. Holliday Joanne, Agrawal Divyakant, Abbadi Amr El: The Performance of Database Replication with Group Multicast, [URL: <http://citeseer.nj.nec.com/cache/papers/cs/12035/http:zSzzSzwww.cs.ucsb.edu:zS~joanne46zSzfcs29.pdf/holliday99performance.pdf>], 1999. 8 str.
17. Inmon H. William: Effective data base design. London: Prentice-Hall, 1981. 228 str.
18. Jensen S. Christian, Lomet B. David: Transaction Timestamping in (Temporal) Databases, [URL: <http://www.vldb.org/conf/2001/P441.pdf>], 2001. 10 str.
19. Kemme Bettina et al.: Using Optimistic Atomic Broadcast in Transaction Processing Systems, [URL: <http://csdl.computer.org/comp/trans/tk/2003/04/k1018abs.htm>], 2003. 26 str.
20. Kodek Dušan: Organizacija in arhitektura računalniških sistemov. Ljubljana: Fakulteta za elektrotehniko in računalništvo, 1988. 316 str.
21. Maier Luke David: The theory of relational databases. Rockville: Computer Science Press, 1983. 637 str.
22. Makofske David B., Almeroth Kevin C.: Multicast Sockets: Practical Guide for Programmers. Morgan Kaufmann, 2002. 175 str.
23. Malkin Gary Scott: RIP: An Intra-Domain Routing Protocol. Wokingham: Addison-Wesley Publishing Company, 1999. 144 str.
24. Martin James: Design and strategy for distributed data processing. London: Prentice-Hall, 1981. 624 str.
25. Miller C. Kenneth: Multicast Networking & Applications. Wokingham: Addison-Wesley Publishing Company, 1998. 282 str.
26. Minoli Daniel: Enterprise Networking: Fractional T1 to Sonet, Frame Relay to BISDN. Boston: Artech House, 1993. 734 str.
27. Mohorič Tomaž: Podatkovne baze. Ljubljana: Fakulteta za elektrotehniko in računalništvo, 1991. 228 str.
28. Moy J.: MOSPF: Analysis and Experience, [URL: <ftp://ftp.rfc-editor.org/in-notes/rfc1585.txt>], 1994. 13 str.
29. Postel Jon: Transmission Control Protocol, [URL: <ftp://ftp.rfc-editor.org/in-notes/std/std7.txt>], 1981. 85 str.
30. Rankins Ray, Jensen Paul, Bertucci Paul: Microsoft SQL Server 2000 Unleashed, 2nd Edition. Sams, 2002. 1560 str.
31. Rob Peter, Coronel Carlos: Database Systems: Design, Implementation, and Management, Fifth Edition. Course Technology, 2001. 838 str.
32. Sanjoy Paul et al: Reliable Multicast Transport Protocol (RMTP), [URL: <http://citeseer.nj.nec.com/cache/papers/cs/883/http:zSzzSzremus.rutgers.edu:zS~psanjoyzSzrmtp.jsac.www.pdf/paul97reliable.pdf>], 1997. 19 str.
33. Saunders Stephen: The McGraw-Hill high-speed LANs handbook. New York: McGraw-Hill, 1996. 460 str.

34. Schulzrinne H., et al.: RTP: A Transport Protocol for Real-Time Applications, [URL: <http://www.ietf.org/rfc/rfc3550.txt?number=3550>], 2003, 104 str.
35. Sloman Morris: Network and Distributed Systems Management. Wokingham: Addison-Wesley Publishing Company, 1994. 666 str.
36. Speakman T. et al: PGM Reliable Transport Protocol Specification, [URL: <http://www.ietf.org/rfc/rfc3208.txt?number=3208>], 2001. 111 str.
37. Thomas Tom, Thomas Thomas M. II: OSPF Network Design Solutions, Second Edition. Indianapolis: Cisco Press, 2003. 768 str.
38. Vegesna Srinivas: IP Quality of Service. Indianapolis: Cisco Press, 2001. 343 str.
39. Waitzman D., Partridge C., Deering S.: Distance Vector Multicast Routing Protocol, [URL: <ftp://ftp.rfc-editor.org/in-notes/rfc1075.txt>], 1988. 24 str.
40. Williamson Beau: Developing IP Multicast Networks, Volume I. Indianapolis: Cisco Press, 2000. 568 str.
41. Wittmann Ralph, Zitterbart Martina, Wittman Raklph: Multicast Communication: Protocols, Programming, and Applications. Morgan Kaufmann, 2000. 349 str.
42. Wright R. Gary, Stevens W. Richard: TCP/IP Illustrated, Volume 2: The Implementation. Wokingham: Addison-Wesley Publishing Company, 1995. 1174 str.

9 VIRI

1. ACID. [URL: <http://www.wikipedia.org/wiki/ACID>], 15.9.2003
2. Bidirectional PIM. [URL: <http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t2/dtbipim.htm>], 15.9.2003
3. Compression FAQ. [URL: <http://www.faqs.org/faqs/compression-faq/>], 28.11.2002
4. Concurrency: Deadlock and Starvation. [URL: <http://www.cs.tcd.ie/David.Eadie/LectureNotes/chap06-1.ppt>], 15.9.2003
5. Criticisms of the ISO OSI Reference Model. [URL: <http://www.scit.wlv.ac.uk/~jphb/comms/osirm.crit.html>], 15.9.2003
6. Database Models. [URL: <http://unixspace.com/context/databases.html>], 15.9.2003
7. IP Multicast, interno predavanje podjetja Cisco Systems Ltd., 1999.
8. Multicast Tech Notes. Cisco Systems. [URL: http://www.cisco.com/en/US/tech/tk828/tk363/tech_tech_notes_list.html], 14.12.2002

9. The NTP FAQ and HOWTO. [URL: <http://www.ntp.org/ntpfaq/NTP-a-faq.htm>], 15.9.2003
10. Varnostni mehanizmi, interno predavanje podjetja NIL d.o.o., 2002.
11. Welzer - Družovec Tatjana: Podatkovne baze II. [URL: <http://dbtlab.uni-mb.si/pb2/pb2a.pdf>], 17.11.2002

PRILOGA

SLOVAR ANGLEŠKIH STROKOVNIH IZRAZOV IN KRATIC

Beseda	Razlaga
area	področje
BGP	Border Gateway Protocol; protokol za medomrežno povezovanje
broadcast	razpršeno oddajanje
CBT	Core Based Trees; protokol za usmerjanje prometa večsmernega oddajanja
core	jedro
deadlock	mrtva zanka
designated forwarders	izbrani posredniki
digital signing	elektronsko podpisovanje
dirty read	napačno branje
DVMRP	Distance Vector Multicast Routing Protocol; protokol za usmerjanje prometa večsmernega oddajanja
eager replication	pohlepna replikacija
encryption	šifriranje
frame	okvir
hashing	zgoščevanje
HTTP	Hypertext Transfer Protocol; protokol za prenos dokumentov hiperteksta
indeks	ključ
Internet	medmrežje, svetovno računalniško omrežje
ISO/OSI	International Organization for Standardization/ Open System Interconnection; mednarodna organizacija za standarde, pod okriljem katere je bil razvit omrežni model za povezovanje
kbps	Kilobits per second; kilobitov na sekundo
lazy replication	zakasnjena replikacija
lazy group replication	zakasnjena skupinska replikacija
lazy master replication	zakasnjena centralizirana replikacija
livelock	živa zanka
locking	zaseganje podatkov

MABR	Multicast Area Border Router; mejni usmerjevalnik MOSPF področja
man-in-the-middle attack	napad na način "vmesnega človeka"; nekdo prestreza podatke in s pomočjo pridobljene informacije poskuša izkoristiti komunikacijo za svoje nameme
MGBP	Multi-protocol extensions to BGP; protokol za usmerjanje prometa večsmernega oddajanja
mirroring	zrcaljenje podatkov med trdimi diski s ciljem zagotavljanja neoviranega delovanja sistema v primeru izpada enega izmed trdih diskov
MOSPF	Multicast extension to OSPF; protokol za usmerjanje prometa večsmernega oddajanja
MSDP	Multicast Source Discovery Protocol; protokol za razširjanje informacije o aktivnih oddajnikih
multicast	večsmerno oddajanje
NTP	Network Time Protocol; omrežni časovni protokol za sinhronizacijo sitemskega časa računalnika
optimistic methods	oprimistične metode
OSPF	Open Shortest Path First; protokol za usmerjanje prometa usmerjenega oddajanja
packet	paket
parent-child	starš-otrok
PGM	Pragmatic General Multicast; protokol za zagotavljanje zanesljivega prenosa podatkov večsmernega oddajanja
phantom deadlock	navidezna mrtva zanka
PIM	Protocol Independant Multicast; protokol za usmerjanje prometa večsmernega oddajanja
PIM bidirectional mode	dvosmerni način delovanja protokola PIM
PIM dense mode	zgoščen način delovanja protokola PIM
PIM sparse mode	razpršen način delovanja protokola PIM
PIM source specific multicast	oddajniško usmerjen način delovanja protokola PIM
port	vrata
quorum consensus	metoda večine
rendezvous point	točka srečanja
RMTP	Reliable Multicast Transport Protocol; protokol za zanesljiv prenos podatkov večsmernega oddajanja
RTP/RTCP	Real-time Transfer Protocol/Real-time Transfer

	Control Protocol; protokol za prenos podatkov večsmernega oddajanja
sequencing	številčenje podatkov
shared distribution tree	deljena drevesa
source distribution tree	oddajniška drevesa
TCP/IP	Transmission Control Protocol/Internet Protocol; skupina najbolj razširjenih omrežnih protokolov, ki se uporablja na medmrežju
TCP	Transmission Control Protocol; protokol za prenos podatkov z vzpostavitvijo seje med končnima točkama
three way handshake	rokovanje v treh korakih
time stamping	časovno označevanje
UDP	User Datagram Protocol; protokol za prenos podatkov brez vzpostavitve seje med končnima točkama
unicast	usmerjeno oddajanje
wait-for graphs	grafi "čaka-na"