



Preddiplomski studij

Računarstvo

Modul:

Telekomunikacije i  
informatika

# Višemedijske usluge

## Sustavi Peer-to-Peer (P2P)

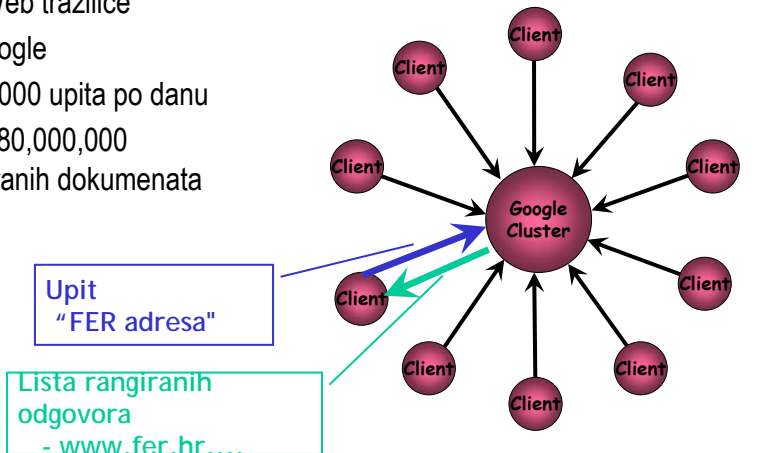
Ak.god. 2007./2008.

05.06.2008.

- ◆ Centralizirani i decentralizirani distribuirani sustavi
- ◆ Definicija sustava P2P
- ◆ Nestrukturirani sustavi P2P
- ◆ Strukturirani sustavi P2P
- ◆ Primjeri aplikacija

## Primjer - Web tražilice

- ◆ npr. Google
- ◆ 91,000,000 upita po danu
- ◆ oko 2,480,000,000 indeksiranih dokumenata



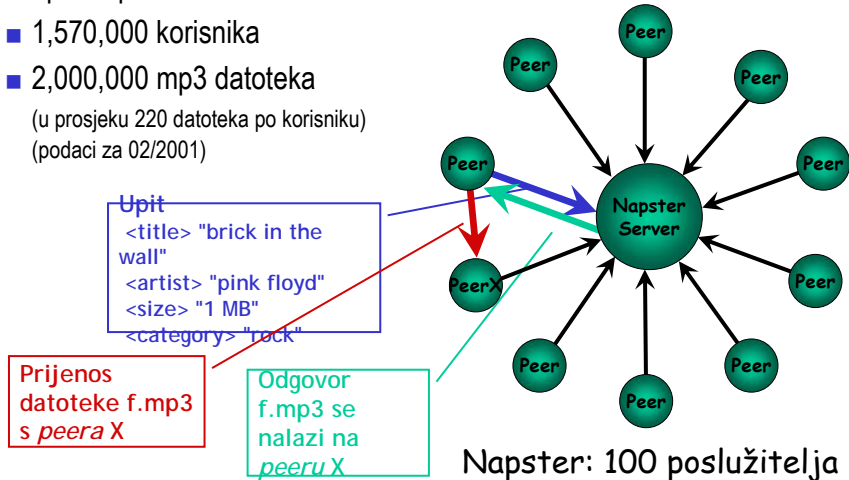
Google Cluster: 15000 poslužitelja  
(podatak iz 2003)

## Model klijent-poslužitelj

- ◆ centralizirani koordinator koji prihvaća sve korisničke upite
- ◆ indeks dokumenata je raspodijeljen, pretraživanje je raspodijeljeno u grozdu računala (*cluster*), no organizacija pretraživanja je centralizirana
- ◆ prednosti
  - efikasnost, kratko vrijeme odgovora
  - globalno rangiranje...
- ◆ nedostaci
  - cijena (infrastruktura, administracija...)

## ◆ Primjer – aplikacija za razmjenu mp3 datoteka

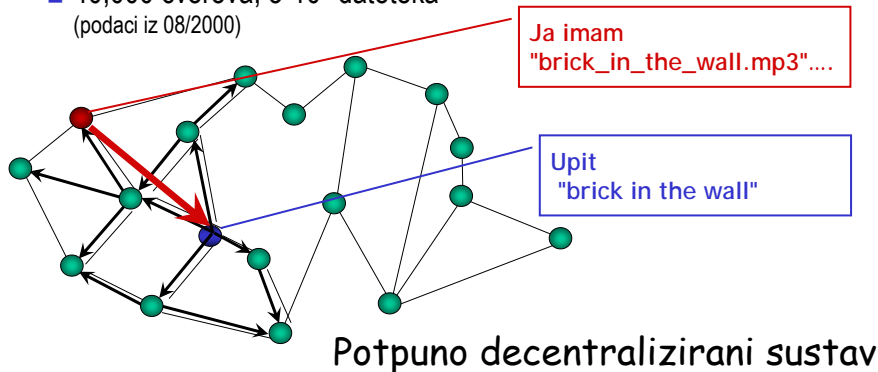
- npr. Napster
- 1,570,000 korisnika
- 2,000,000 mp3 datoteka  
(u prosjeku 220 datoteka po korisniku)  
(podaci za 02/2001)



- ◆ pretraživanje je i dalje centralizirano
  - postoji centralizirani indeks s podacima o lokaciji datoteka
- ◆ pohrana i *download* datoteke je decentraliziran
- ◆ broj potrebnih poslužitelja je znatno manji jer se resursno zahtjevne operacije izvode na decentralizirani način
- ◆ prednosti
  - dijeljenje resursa, svaki čvor (*peer*) “plaća” sudjelovanje u mreži vlastitim resursima (disk, mreža, datoteke)
  - znatno manja cijena infrastrukture i održavanja
- ◆ nedostaci
  - centralizirano pretraživanje i jedinstvena točka ispada

## ◆ Primjer: aplikacija za razmjenu datoteka

- npr. Gnutella
- 40,000 čvorova,  $3 \cdot 10^6$  datoteka  
(podaci iz 08/2000)

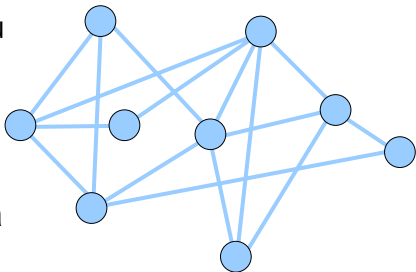


- ◆ Gnutella je primjer potpuno decentraliziranog sustava
  - svi čvorovi sudjeluju u procesu pretraživanja (ne postoji centralizirani indeks)
  - brzo pronalazi datoteke koje su replicirane na velikom broju čvorova
- ◆ prednosti
  - skalabilnost sustava
  - ne postoji posebna infrastruktura niti potreba za održavanjem sustava
  - ne postoji jedinstvena točka ispada
- ◆ nedostaci
  - velika količina generiranog mrežnog prometa
  - ne postoji garancija pronalaska tražene datoteke
  - *free-riding*

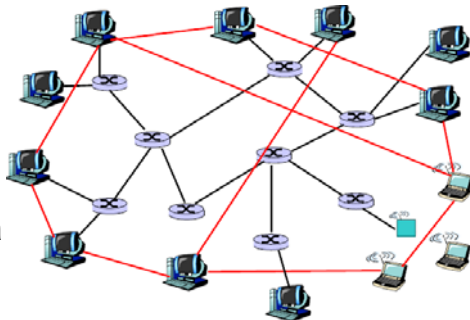


- ◆ Centralizirani i decentralizirani distribuirani sustavi
- ◆ Definicija sustava P2P
- ◆ Nestrukturirani sustavi P2P
- ◆ Strukturirani sustavi P2P
- ◆ Primjeri aplikacija

- ♦ mreža istovrsnih “čvorova” - *peerova*
- ♦ svaki *peer* istovremeno vrši funkciju poslužitelja i klijenta
- ♦ svaki čvor “plaća” sudjelovanje u mreži nudeći dio vlastitih resursa (memorija, CPU) ostalim čvorovima
- ♦ potencijalno sustav P2P nudi neograničene resurse (broj *peerova* nije ograničen)



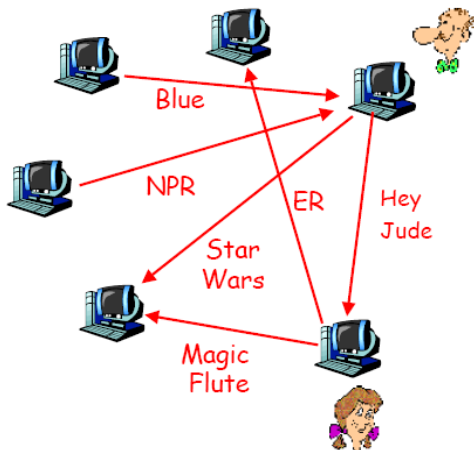
- ♦ “prekrivajuća mreža” (*overlay network*) nad stvarnom mrežnom topologijom
- ♦ *peerovi* su programi koji se izvode na aplikacijskom sloju
- ♦ koristi resurse krajnjih računala koji čine posebnu mrežu neovisnu o mrežnoj topologiji
- ♦ mreža *peerova* se konstantno mijenja (računala se spajaju i odspajaju)



- ◆ Kada su 2 *peera* susjedi?
  - otvorena TCP konekcija
  - virtualne grane među *peerovima*, *peer* zna IP adresu drugog *peera*
- ◆ Kako se održava mreža *peerova*?
  - mreža je izrazito nestabilna
  - npr. *peer* periodički provjerava stanje susjeda (ping porukama)
  - ako je susjed nedostupan, briše se iz liste susjeda
  - potreban je poseban algoritam za otkrivanje novih susjeda
  - poseban algoritam za dodavanje novog *peera* u postojeću mrežu (najčešće poznaje listu *peerova* za inicijalni kontakt)

- ◆ decentralizirani distribuirani sustav
  - nema centralizirane koordinacije među *peerovima*
  - ne postoji jedna točka ispada
- ◆ samoorganizirajuća mreža čvorova
  - *peerovi* su međusobno neovisni
- ◆ skalabilan sustav
  - dodavanje novih čvorova i ispad čvorova je podržano organizacijom P2P mreže i definiranim protokolima
- ◆ globalni informacijski sustav bez velikih ulaganja
  - raspodijeljena instalacija i održavanje

- ◆ Pronalaženje podataka u sustavima P2P!



- ◆ Kako pronaći podatak  $d$  u mreži peerova?
  - $peer$  s adresom  $p$  pohranjuje podatak  $d$  (npr. datoteka) koji se može jedinstveno identificirati ključem  $k$  (npr.  $k = \text{hash}(d)$  )
  - za dani ključ  $k$  pronaći  $peera$   $p$  na kome je pohranjen podatak  $d$
  - dovoljno je znati funkciju  $f: k \rightarrow p$  da bismo pronašli  $d$  jer ta funkcija određuje identifikator  $peera$  na kome je pohranjen  $d$
  - Kako definirati i implementirati funkciju  $f$  u distribuiranoj i decentraliziranoj okolini?

- ◆ nestrukturirani sustavi
  - primjeri: Freenet, Gnutella
  
- ◆ strukturirani sustavi
  - primjeri: CAN, Chord, P-Grid, Pastry

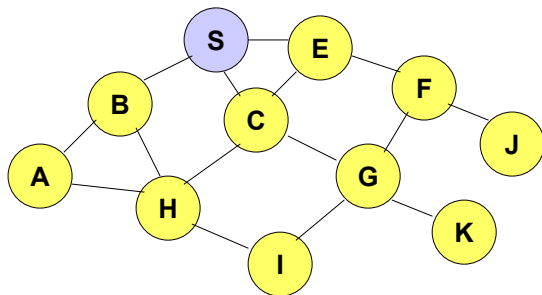


- ♦ mrežna topologija nema definiranu strukturu
- ♦ mrežu *peerova* čini slučajan graf, npr. peer “poznaje” svoja četiri susjeda i preko njih pretražuje cijelu mrežu

- ◆ mrežna topologija je definirana i ima posebnu strukturu
- ◆ podatak  $d$  jedinstveno određuje ključ  $k$  (svaki peer može odrediti  $k$  za  $d$ )
- ◆ podatak  $d$  je pohranjen na *peeru* koji je “zadužen” za ključ  $k$ , a ne na peeru koji ga kreira

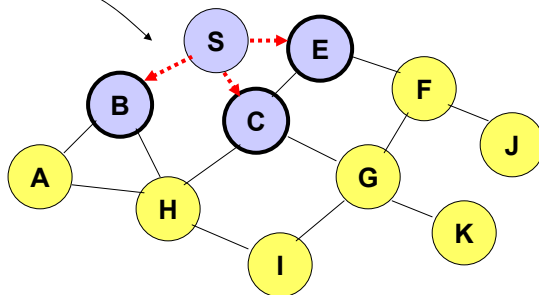
- ◆ Centralizirani i decentralizirani distribuirani sustavi
- ◆ Definicija sustava P2P
- ◆ **Nestrukturirani sustavi P2P**
- ◆ Strukturirani sustavi P2P
- ◆ Primjeri aplikacija

- ◆ podatak (npr. datoteka) je pohranjen na *peeru* koji ga kreira, ne postoji veza između podatka  $d$  i peera  $p$
- ◆ moguće je pohraniti kopiju podatka na peerovima koji ga kopiraju s originalnog peera
- ◆ pretraživanje se izvodi preplavljanjem ili slučajnim izborom (random walk), itd.



Oznaka čvora koji je izvor upita “q”.

preplavlivanje susjednih čvorova



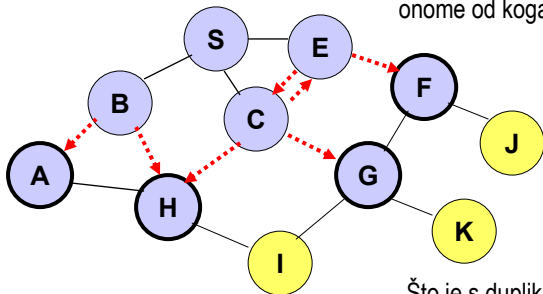
Oznaka čvora koji je primio “q” prvi put.



Prijenos upita “q”

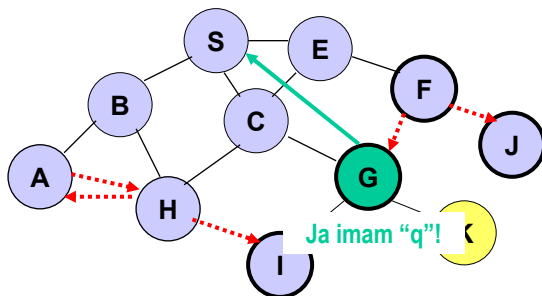
Osnovno načelo:

proslijedi upit svim susjedima osim  
onome od koga si ga primio



Što je s duplikatima?

Npr. H je primio upit od B i C. Ako  
upit ima jedinstveni identifikator, H  
uočava duplikat i ignorira ga.



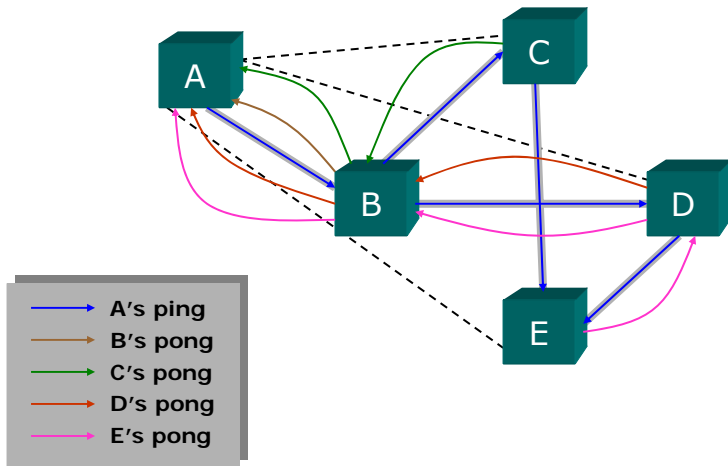
G šalje odgovor direktno do S, no A, F i H ne znaju da je  $q$  pronađen i nastavljaju s preplavljanjem svojih susjeda!



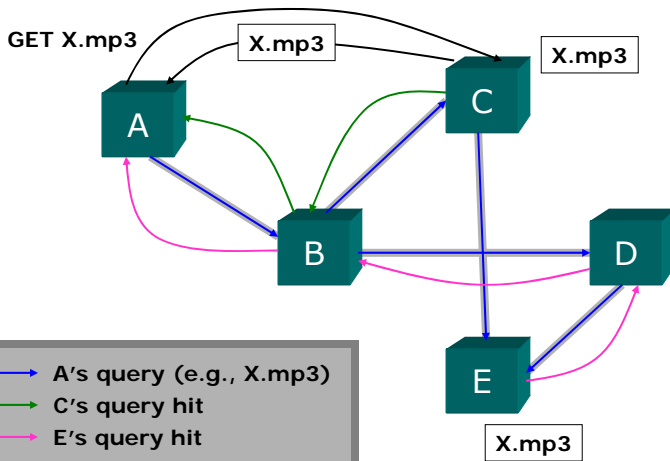
- ◆ Gnutella
- ◆ KaZaA
- ◆ BitTorrent

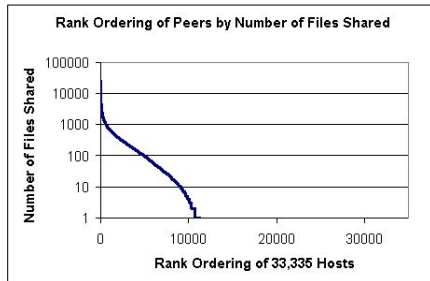
- ◆ aplikacija razvijena u 14 dana za razmjenu recepata (“quick hack” by Nullsoft)
- ◆ koristi ograničeno preplavlivanje pri traženju podataka
  - svaki čvor šalje upit svim svojim susjedima
  - vrijeme valjanosti upita je ograničeno parametrom *time-to-live* (TTL = 7)
  - svaki upit ima jedinstveni identifikator zbog petlji u mreži
- ◆ novi čvor se jednostavno povezuje u sustav tako da se spoji na barem jedan poznati Gnutella čvor

Type	Description	Contained Information
Ping	Announce availability and probe for other servents	None
Pong	Response to a ping	IP address and port# of responding servent; number and total kb of files shared
Query	Search request	Minimum network bandwidth of responding servent; search criteria
QueryHit	Returned by servents that have the requested file	IP address, port# and network bandwidth of responding servent; number of results and result set
Push	File download requests for servents behind a firewall	Servent identifier; index of requested file; IP address and port to send file to

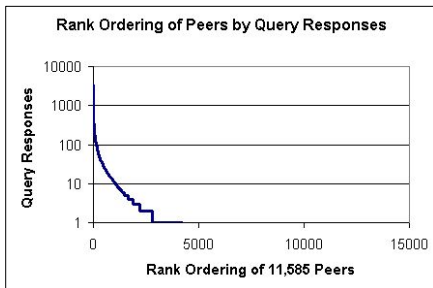


# Gnutella: pretraživanje (Query/QueryHit/GET)





- ◆ Veliki postotak korisnika su “*free riders*”
  - 66% peerova ne nudi vlastite datoteke
  - 73% peerova nude 10 ili manje datoteka
  - 1% peerova nude 37% svih datoteka
  - 10% peerova nude 87% svih datoteka



- ◆ Veliki broj peerova nudi datoteke koje nikoga ne zanimaju
- ◆ od 11,585 peerova koji nude datoteke:
  - 1% peerova odgovara na 47% svih upita
  - 25% peerova odgovara na 98% svih upita
  - 63% peerova nikada ne odgovaraju na upite

- ◆ jednostavnost (jednostavan protokol)
- ◆ robustnost (ne postoji jedna točka ispada)
- ◆ niska cijena objavljivanja novog podatka (podatak ostaje na peeru koji ga objavljuje)
- ◆ velika cijena prilikom pretraživanja, generira se veliki mrežni promet
- ◆ dobro za pronalaženje podataka koji su replicirani na velikom broju *peerova*, ali ne za podatke pohranjene na malome broju *peerova*

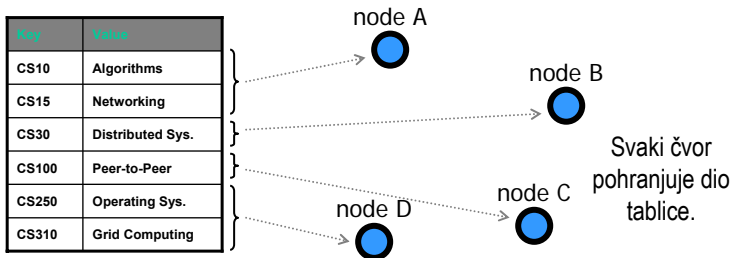


- ◆ Centralizirani i decentralizirani distribuirani sustavi
- ◆ Definicija sustava P2P
- ◆ Nestrukturirani sustavi P2P
- ◆ **Strukturirani sustavi P2P**
- ◆ Primjeri aplikacija

- ♦ podatak  $d$  je pohranjen na *peeru* koji je “zadužen” za ključ  $k$ , a ne na *peeru* koji ga kreira
- ♦ svaki *peer* ima jedinstveni identifikator (adresu)  $p$  te je moguće definiranim distribuiranim algoritmom jednoznačno povezati  $p$  i  $k$ , a stoga i pronaći  $d$
- ♦ sustav P2P implementira metodu  $\text{lookup}(k)$  koja vraća identifikator *peera* za dani ključ  $k$

## Tipičan primjer strukturiranog sustava P2P

- hash tablica je raspodijeljena na više čvorova.

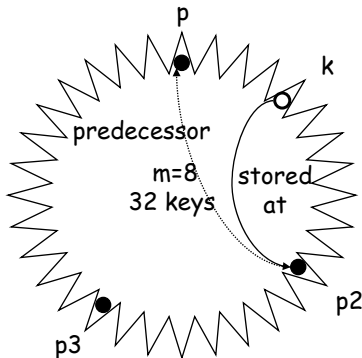


- *lookup* omogućuje svakom čvoru da pronade vrijednost povezanu s nekim ključem
- primjer:  
lookup("CS30"), odgovor: "Distributed Sys."

- ◆ moraju realizirati dvije osnovne funkcije
  - $\text{put}(k, d)$
  - $d = \text{get}(k)$
- ◆ osnova je metoda  $\text{lookup}(k)$  nužna za implementaciju  $\text{put}$  i  $\text{get}$
- ◆ svaki *peer* održava dio globalnog DHT-a, odgovoran je za podskup ključeva  $k$  i njima pridruženih podataka  $d$
- ◆ pretraživanje se provodi u ograničenom broju koraka
  - kada *peer* primi upit, ako ga ne može riješiti jer nije zadužen za  $k$ , *peer* ga prosljeđuje drugome *peeru* koji će s većom vjerojatnošću moći odgovoriti na upit
  - može se garantirati da će podatak biti pronađen

- ◆ Chord
- ◆ CAN
- ◆ P-Grid
- ◆ Pastry

- ♦ koristi *hash* funkciju koja se izvodi nad podatkom i adresom peera i proizvodi binarni ključ duljine **m**
  - npr.  $m=8$
  - podatkovni ključ:  $\text{key}(\text{"jingle-bells.mp3"})=17$
  - ključ peera:  $\text{key}(196.178.0.1)=3$
- ♦ podatak "jingle-bells.mp3" se pohranjuje na *peeru* s prvim **većim** binarnim ključem



## Mogućnosti pretraživanja

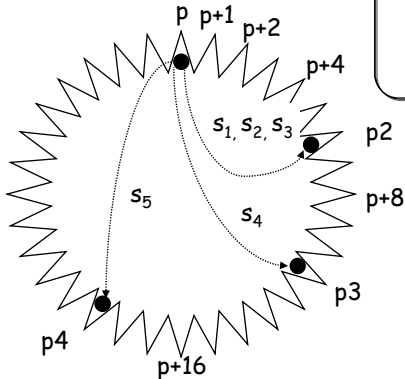
1. svaki peer zna sve ostale peerove u mreži  
 $O(n)$  routing table size
2. svaki peer zna samo svoga prethodnika  
 $O(n)$  search cost

- svaki peer zna  $m$  ostalih peerova s rastućom distancom (razlika između identifikatora originalnog peera i njegovog susjeda, npr. 1, 2, 4, 8, 16)

Tablica usmjeravanja na peeru  $p$

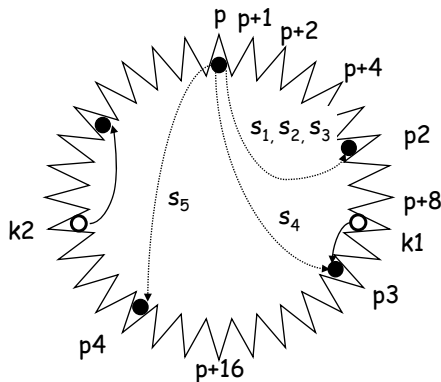
Prvi sljedeći peer čiji je identifikator  $p$  takav da vrijedi

$\rightarrow s_i = \text{successor}(p+2^{i-1})$  for  $i=1, \dots, m$



$s_i$	$p$
$s_1$ (distanca 1)	$p2$
$s_2$ (distanca 2)	$p2$
$s_3$ (distanca 4)	$p2$
$s_4$ (distanca 8)	$p3$
$s_5$ (distanca 16)	$p4$

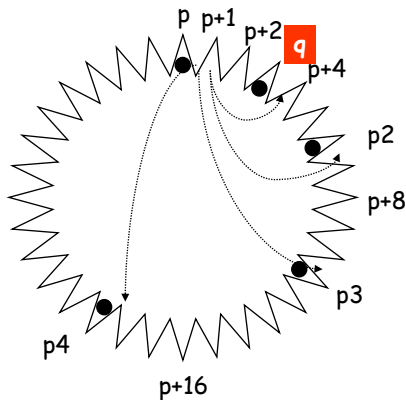
- ◆ kada peer  $p$  primi upit, traži u svojoj tablici usmjeravanja najveći mogući ključ peera koji je **manji** od traženog podatkovnog ključa



Search  
 $O(\log n)$  cijena pretraživanja



- ◆ Novi čvor  $q$  se spaja u mrežu



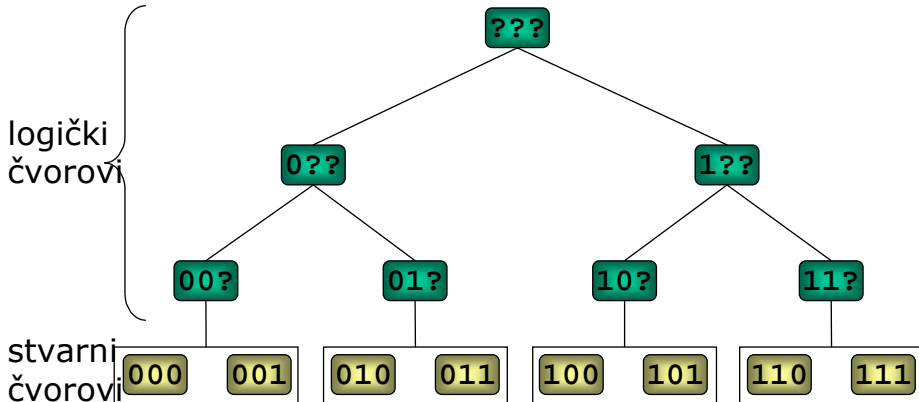
tablica  
usmjeravanja  
peera  $p$

$i$	$p$
$s_1$	$q$
$s_2$	$q$
$s_3$	$p2$
$s_4$	$p3$
$s_5$	$p4$

tablica  
usmjeravanja  
peera  $q$

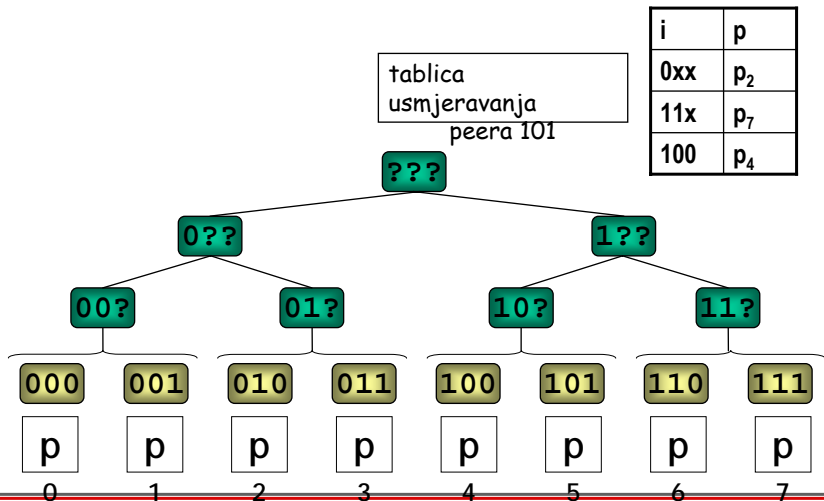
$i$	$p$
$s_1$	$p2$
$s_2$	$p2$
$s_3$	$p3$
$s_4$	$p3$
$s_5$	$p4$

## ♦ binarno stablo

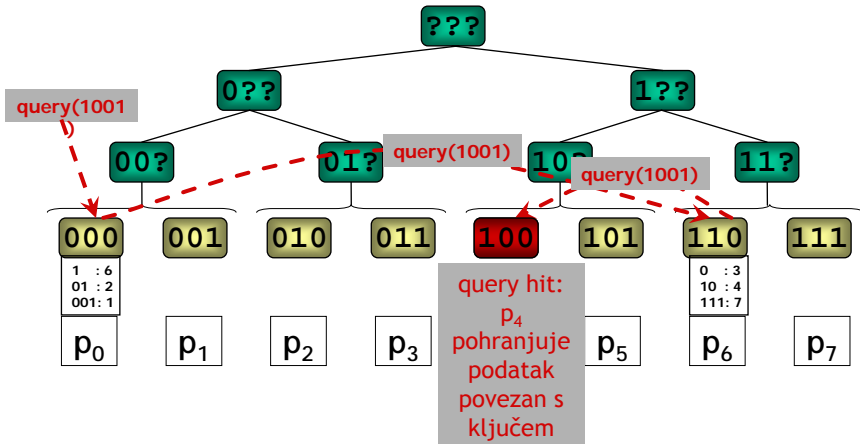


npr. *peer* 101 je zadužen za ključeve  
101xxx

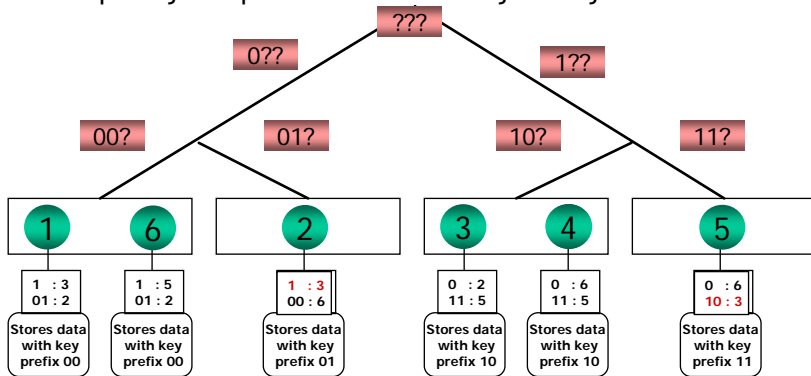
- ♦ za dani primjer svaki peer mora znati 3 peera koji su zaduženi za “drugi dio stabla”



- ◆ kada peer primi upit, proslijeđuje ga čvoru s najbližim ključem iz svoje tablice usmjeravanja

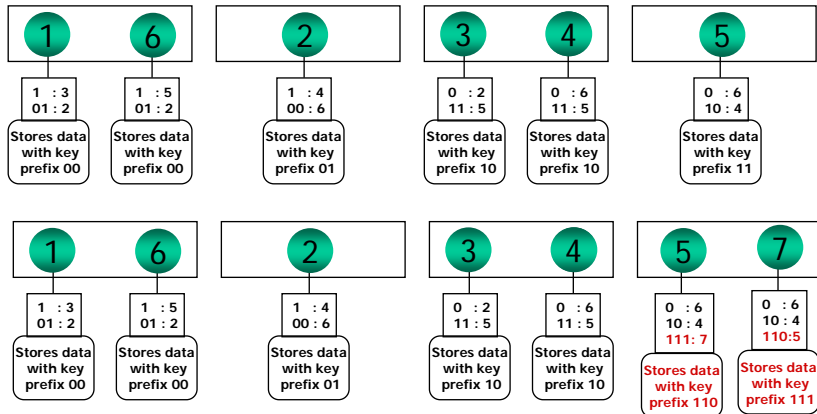


Veći broj peerova može biti zadužen za isti adresni prostor. Npr. peerovi 1 i 6, te 3 i 4. Replikacija se koristi da bi sustav podržao proces kojim peerovi napuštaju mrežu. Potrebno je također promijeniti podatke u tablici usmjeravanja.



# Dodavanje novog čvora u mrežu

Pri dodavanju novog peera u mrežu, kontaktira se peer u postojećoj mreži i u najjednostavnijem slučaju novi peer dijeli adresni prostor s novim peerom.



- ◆ garantira pronalaženje podatka u  $O(\log n)$  koraka ( $n$  je broj *peerova* u mreži)
  - skalabilno rješenje u odnosu na nestrukturirane sustave
- ◆ povećana cijena objavljivanja novog podatka u odnosu na nestrukturirane P2P sustave
  - podatak se pohranjuje na peeru koji je za njega “zadužen”
- ◆ potrebno je održavati dodatne strukture podataka (tablice usmjeravanja) radi umjeravanje upita prema *peerovima* koji pohranjuju tražene podatke

- ◆ Centralizirani i decentralizirani distribuirani sustavi
- ◆ Definicija sustava P2P
- ◆ Nestrukturirani sustavi P2P
- ◆ Strukturirani sustavi P2P
- ◆ Primjeri aplikacija

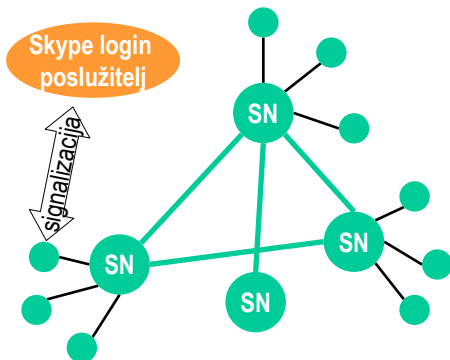


- ◆ Ivica koristi P2P aplikaciju na svome prijenosnom računalu.
- ◆ Povremeno se spaja na internet i pri tome svaki puta njegovo računalo mijenja IP adresu.
- ◆ Ivica registrira datoteke koje ima na svome računalu u sustav i traži “Hey Jude”
- ◆ P2P aplikacija pronalazi ostale peerove koji imaju kopiju “Hey Jude”.
- ◆ Ivica odabire jednoga peera, npr. Maricu od koje kopira datoteku.
- ◆ Dok Ivica kopira datoteku na svoje računalo, ostali korisnici mogu kopirati njegove datoteke.

## Instant messaging (IM)

- ◆ Ivica koristi P2P aplikaciju na svome prijenosnom računalu.
- ◆ Povremeno se spaja na internet i pri tome svaki puta njegovo računalo mijenja IP adresu.
- ◆ Ivica se registrira u IM sustav i pri tome nalazi Maricu koja je online.
- ◆ Za realizaciju komunikacije (slanje kratkih poruka) otvara se direktna TCP konekcija između 2 računala – P2P komunikacija.

- ◆ mreža istovjetnih čvorova koja se koristi za realizaciju usluga VoIP i IM, uključuje prisutnost
- ◆ usluga SkypeOut omogućuje pozive s PC-a prema fiksnoj i pokretnoj mreži
- ◆ hijerarhijska mreža P2P: svaki čvor s javnom IP adresom, zadovoljavajućim RAM-om i brzinom prijenosa može biti odabran za “super čvor” (SN), “samoorganizirajuća mreža”



Mreža sustava Skype