



Diplomski studij
Informacijska i
komunikacijska tehnologija:
Telekomunikacije i informatika
Računarstvo:
Programsko inženjerstvo i
informacijski sustavi
Računarska znanost

Ak.god. 2009./2010.

Raspodijeljeni sustavi

7. Imenovanje

Zaštićeno licencom <http://creativecommons.org/licenses/by-nc-sa/2.5/hr/>





■ slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo

■ pod sljedećim uvjetima:

- **imenovanje**. Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno**. Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima**. Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, prerađu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava. Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licencije preuzet je s <http://creativecommons.org/>.

- ◆ Uvod
- ◆ Nestrukturirano imenovanje
 - razašiljanje, pokazivači prema naprijed, domaća lokacija, raspodijeljene *hash* tablice
- ◆ Strukturirano imenovanje
 - URI, prostor imena, razlučivanje imena, mrežni datotečni sustav, DNS
- ◆ Imenovanje temeljeno na atributima
 - LDAP, RDF

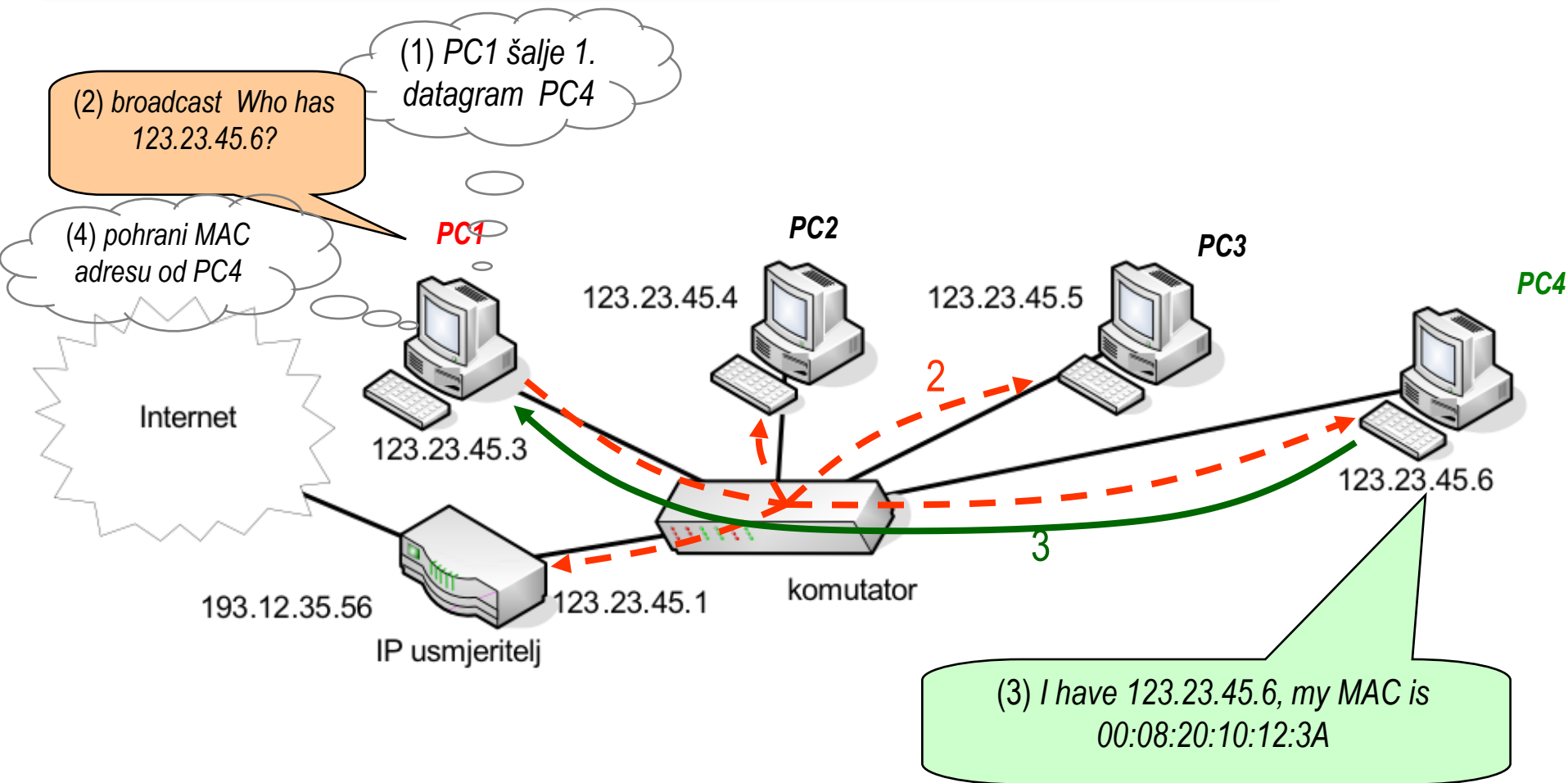
- ◆ imenovanje je vrlo bitno u računalnim sustavima
- ◆ koristi se za:
 - dijeljenje resursa,
 - jedinstveno identificiranje entiteta,
 - označavanje lokacija, itd.
- ◆ u raspodijeljenim sustavima je imenovanje raspodijeljeno
 - učinkovitost i skalabilnost ovise o implementaciji
- ◆ imena možemo podijeliti na:
 - čovjeku prilagođena (*human friendly*) i
 - npr: www.tel.fer.hr
 - čovjeku neprilagođena
 - npr: 161.53.19.59

- ◆ **Ime** (*name*) je niz znakova ili bitova koji predstavljaju entitet.
- ◆ Za pristup entitetu je potrebna **pristupna točka** (*access point*).
- ◆ Ime pristupne točke je **adresa** (*address*).
- ◆ Entitet tijekom vremena može promijeniti pristupnu točku.
 - premještanje računala u drugu mrežu
- ◆ Identifikator (*identifier*) je ime sa sljedećim svojstvima:
 - identifikator predstavlja najviše jedan entitet
 - svaki entitet je predstavljen najviše jednim identifikatorom
 - identifikator uvijek predstavlja isti entitet (nikad nije ponovno iskorišten)

- ◆ u većini slučajeva su slučajan niz bitova
- ◆ ne sadrži informacije kako pronaći pristupnu točku entiteta
- ◆ načini pronalaženja pristupnih točaka:
 - jednostavni načini (samo u lokalnim mrežama)
 - opće razasijanje (*broadcast*)
 - višeodredišno razasijanje (*multicast*)
 - pokazivači prema naprijed (*forwarding pointers*)
 - domaća lokacija (*home location*)
 - raspodijeljene tablice raspršivanja (*distributed hash tables*)
 - hijerarhijski pristup

- ◆ najjednostavniji način je svima poslati upit
- ◆ taj princip se koristi u lokalnim mrežama
 - kada imamo IP adresu, a ne znamo MAC adresu
 - ARP (*Address Resolution Protocol*)
- ◆ postaje neefikasno kada grupa raste
 - iskorištavanje kapaciteta mreže
 - puno zahtjeva na koji jedan čvor ne može odgovoriti

ARP - primjer



- ◆ bolje rješenje od općeg razašiljanja
- ◆ ovaj mehanizam je ugrađen u sklopovlje i usmjeritelje
- ◆ računala se mogu sama uključiti u neku višeodredišnu grupu
- ◆ može se koristiti za lociranje entiteta u mreži npr.
 - imamo skup prijenosnih računala
 - kada se računalo spoji na mrežu dobije IP adresu
 - nakon toga se uključi u višeodredišnu grupu
 - pri lociranju se na grupu pošalje upit “gdje je računalo A?”
 - ako je A prisutan on će odgovoriti sa svojom IP adresom

◆ Princip rada:

- kad se entitet miče s mjesta A na mjesto B onda na mjestu A ostavi pokazivač na novu lokaciju

◆ Prednost: jednostavnost

◆ Nedostatak:

- lokacije u sredini lanca moraju održavati pokazivače
- ranjivost na pokvarene pokazivače
- jako pokretljivi entiteti
 - dugačak lanac - isplativost pronalaženja
 - treba dodati mehanizme za skraćivanje lanca
 - ▶ odgovor od entiteta sadrži i trenutnu lokaciju
 - ▶ odgovor može ići direktno ili kroz obrnuti lanac

◆ Primjena: neki agentski sustavi (*Voyager*), *garbage collection*

- ◆ domaća lokacija prati trenutnu lokaciju entiteta
- ◆ domaća lokacija je obično mjesto gdje je entitet stvoren
- ◆ nakon svakog premještanja entitet se javlja domaćoj lokaciji
- ◆ nedostatak:
 - prvo se mora kontaktirati domaća lokacija (povećano kašnjenje)
 - domaća lokacija je fiksirana i stalno mora biti na istom mjestu kroz duži period
 - nepotrebno ažuriranje trenutne lokacije kada pokretni entitet nitko ne kontaktira
- ◆ primjena: pokretni IP, telekomunikacijska mreža, agentski sustavi (JADE)

- ◆ DHT (*Distributed hash tables*) omogućuju pronalaženje temeljeno na ključu i vrijednosti
- ◆ korisnik može spremiti ili dohvatiti podatak pod određenim ključem
- ◆ koristi se u:
 - P2P sustavima,
 - raspodijeljenim datotečnim sustavima,
 - sustavima raspodijeljenog sadržaja (*content distribution systems*),
 - kooperativnom *cacheiranju* weba (*cooperative web caching*),
 - višeodredišno razašiljanje,
 - DNS,
 - trenutno poručivanje, ...

◆ svojstva:

- decentraliziranost - nema centralnog čvora
- skalabilnost - sustav radi učinkovito i s 1000 i s 1000000 čvorova
- otporan na greške - treba raditi bez obzira na dolazak, odlazak ili kvar pojedinih čvorova

◆ najpoznatiji protokoli i implementacije:

- CAN (*Content Addressable Network*),
- Chord,
- Kademlia,
- Pastry,
- P-Grid i
- Tapestry

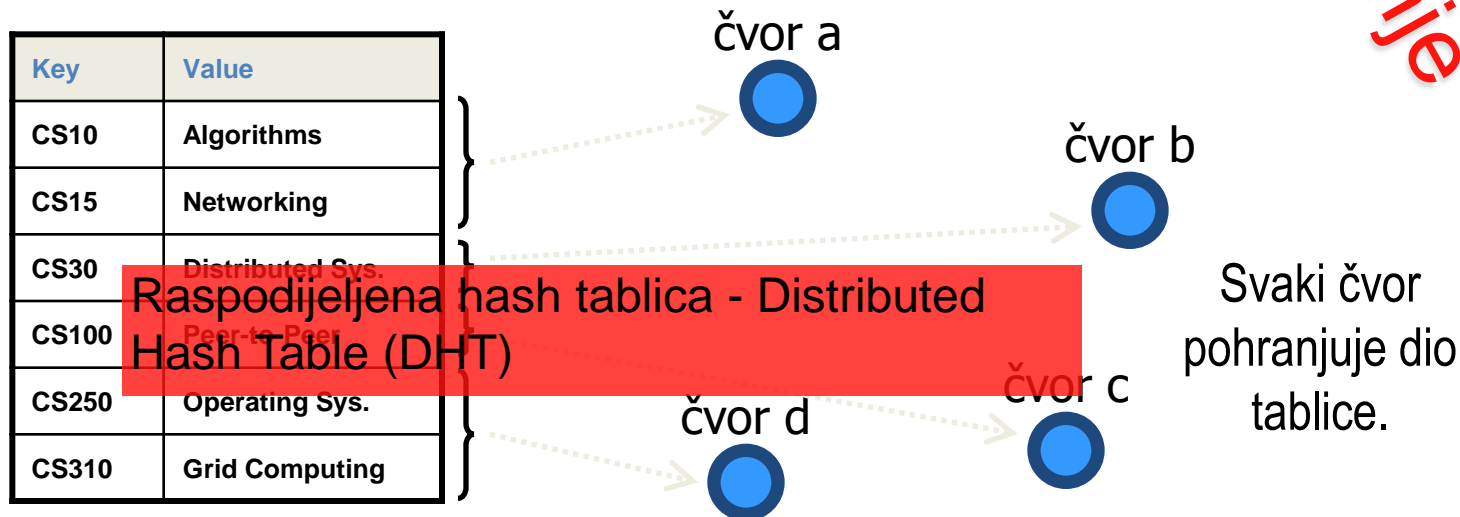
Raspodijeljena *hash* tablica - *Distributed Hash Table* (DHT)



Zavod za komunikacije

Ponavljanje

- ◆ Tipičan primjer strukturiranog sustava P2P
 - *hash* tablica je raspodijeljena na više čvorova.



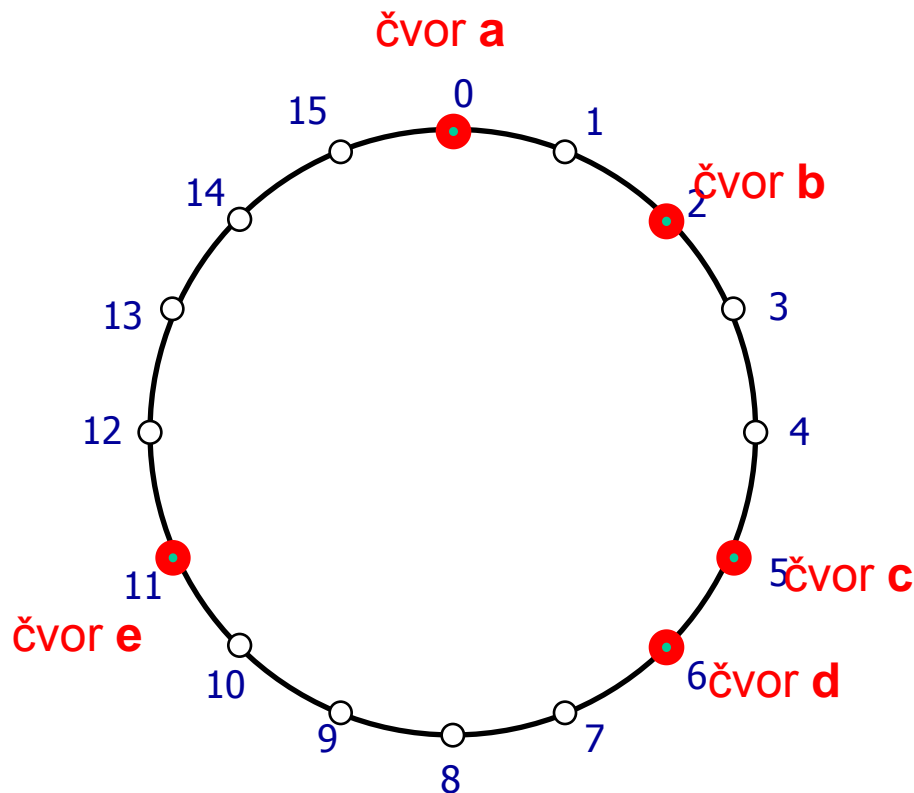
- *lookup* omogućuje svakom čvoru pronalaženje vrijednosti povezane s nekim ključem
- primjer:
lookup("CS30"), odgovor: "*Distributed Sys.*"

Primjer: Kako raspodijeliti *hash* tablicu?



Zavod za komunikacije

ponavljanje



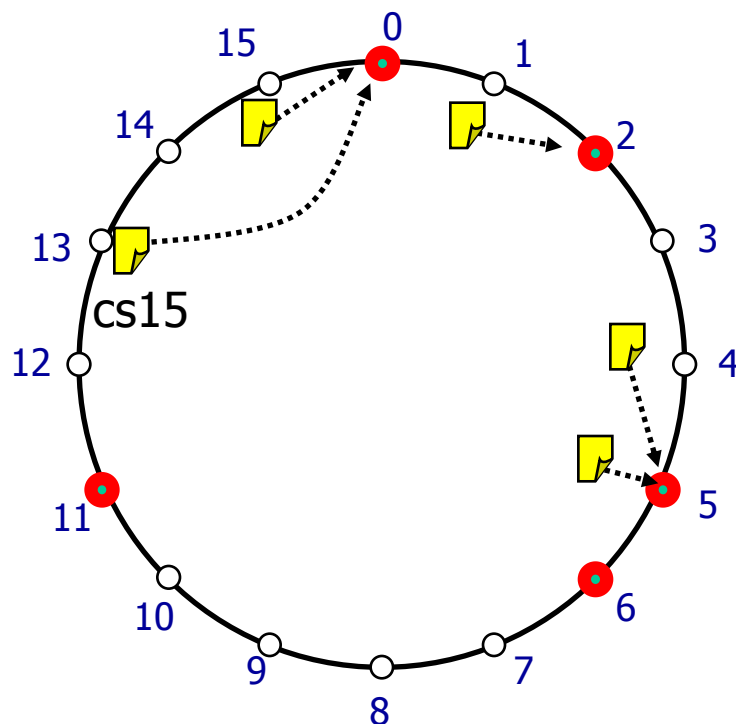
Mogući identifikatori
čvorova:
 $\{0, 1, \dots, 15\}$, $N = 16$

- ◆ Primjer mreže s 5 čvorova **{a,b,c,d,e}**
- ◆ Koristi se prostor identifikatora veličine $N = 16$ (dovoljna su 4 bita za kodiranje)
- ◆ Čvorovima se jednoznačno pridjeljuju identifikatori uz pomoć posebne funkcije H_1 , npr $H_1(a) = 0$.

Kako se podatak dodjeljuje čvoru?



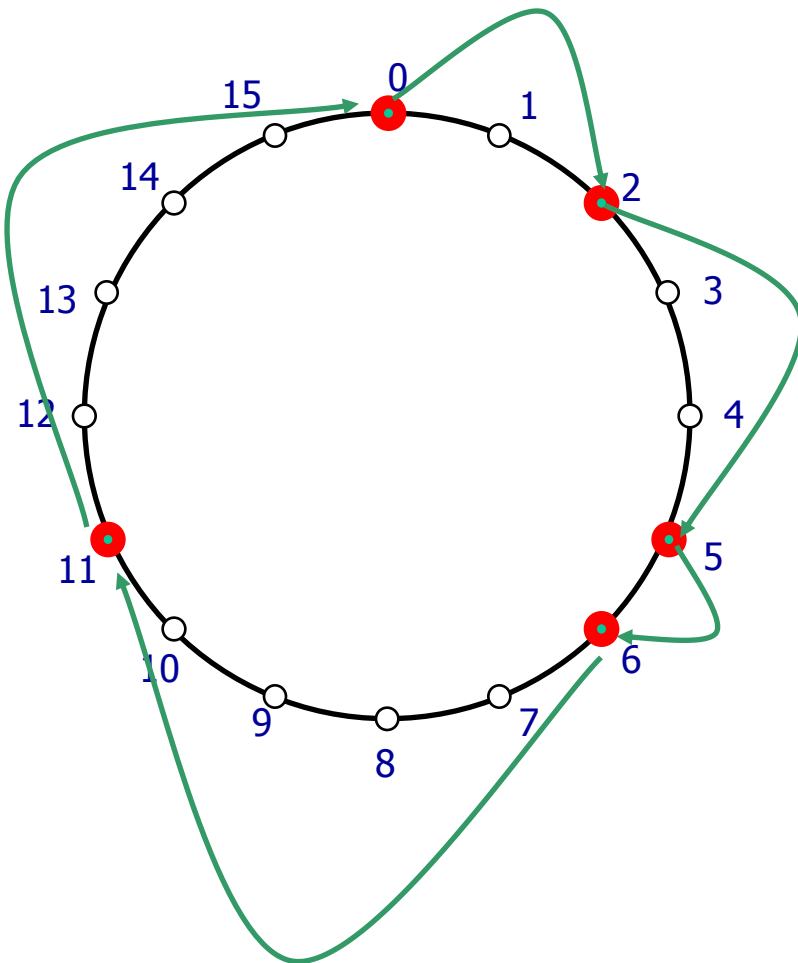
Zavod za telekomunikacije



- ◆ Podacima se pridjeljuju identifikatori iz istog prostora $\{0, 1, \dots, 15\}$ koristeći funkciju H_2
- ◆ Npr. podatak ("**cs15**", "**networking**") se pridjeljuje identifikatoru 13 jer vrijedi $H_2(\text{"cs15"}) = 13$
- ◆ Kako u mreži ne postoji čvor s $\text{id} = 13$, podatak se pohranjuje na prvi sljedeći čvor (to je u ovom slučaju čvor a kojem je $\text{id} = 0$)

Kako ćemo povezati čvorove?

- ◆ Svaki čvor održava jedan pokazivač na sljedbenika, tj. na prvi sljedeći čvor u smjeru kazaljke na satu



sljedbenik čvora 0 → čvor 2
sljedbenik čvora 2 → čvor 5
sljedbenik čvora 5 → čvor 6
sljedbenik čvora 6 → čvor 11
sljedbenik čvora 11 → čvor 0

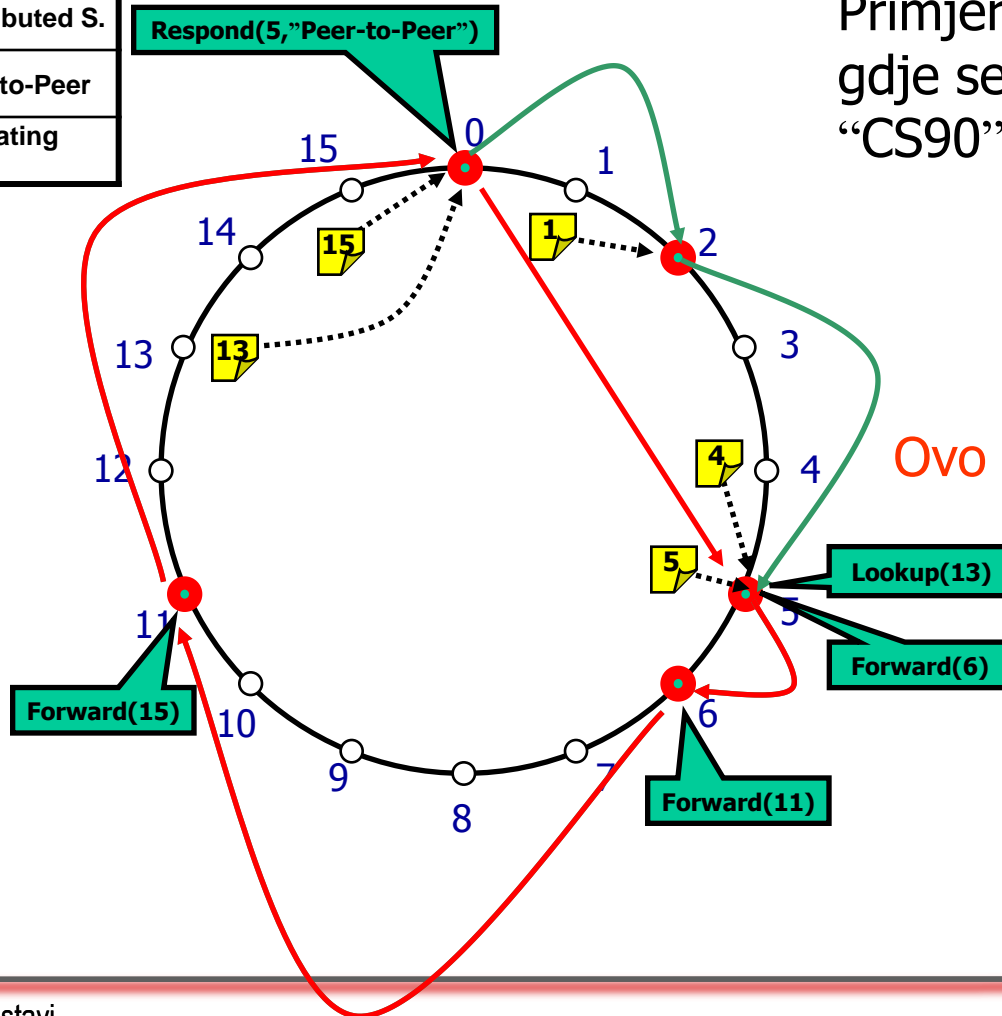
Jednostavno pretraživanje



Zavod za telekomunikacije

Ponavljanje

Key	Value
1 (CS10)	Algorithms
4 (CS15)	Networking
5 (CS30)	Distributed S.
13 (CS90)	Peer-to-Peer
15 (CS95)	Operating Sys.



Primjer: Dolazi upit s čvora 5 gdje se traži vrijednost za ključ “CS90”, $H_2(\text{“CS90”})=13$

Ovo rješenje je vrlo sporo!

Kako ubrzati pretraživanje (1)



Zavod za telekomunikacije

ponavljanje

Primjer: tablica usmjeravanja za $n=16$

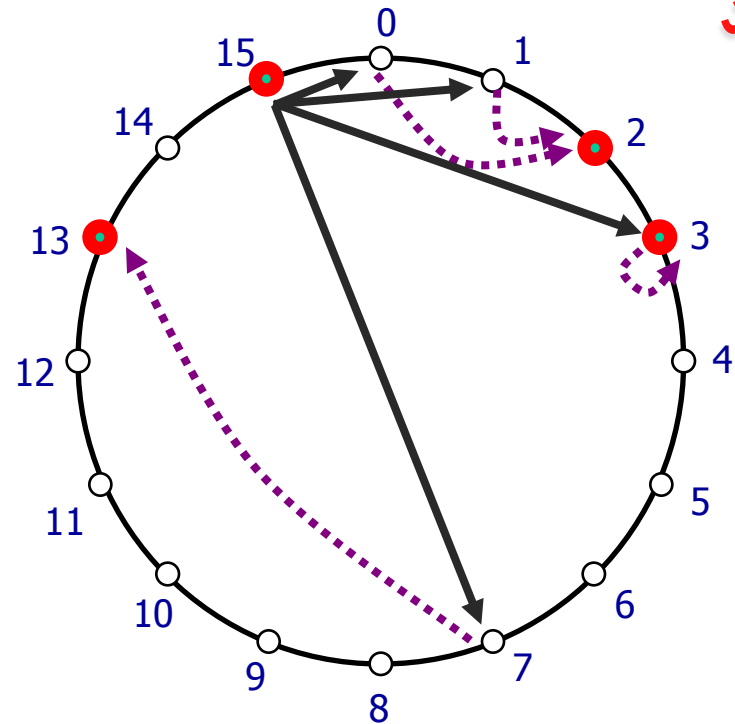
$15+2^0=0 \rightarrow \text{čvor 2}$

$15+2^1=1 \rightarrow \text{čvor 2}$

$15+2^2=3 \rightarrow \text{čvor 3}$

$15+2^3=7 \rightarrow \text{čvor 13}$

Za čvorove koji ne postoje, pokazivač se postavlja na prvog sljedbenika!



Kako ubrzati pretraživanje (2)



Zavod za telekomunikacije

Primjer: na čvoru 3 je postavljen upit za podatkom čiji je identifikator jednak 14, podatak s id=14 je pohranjen na čvoru 15

Routing table (15)

0, 2
1, 2
3, 3
7, 13

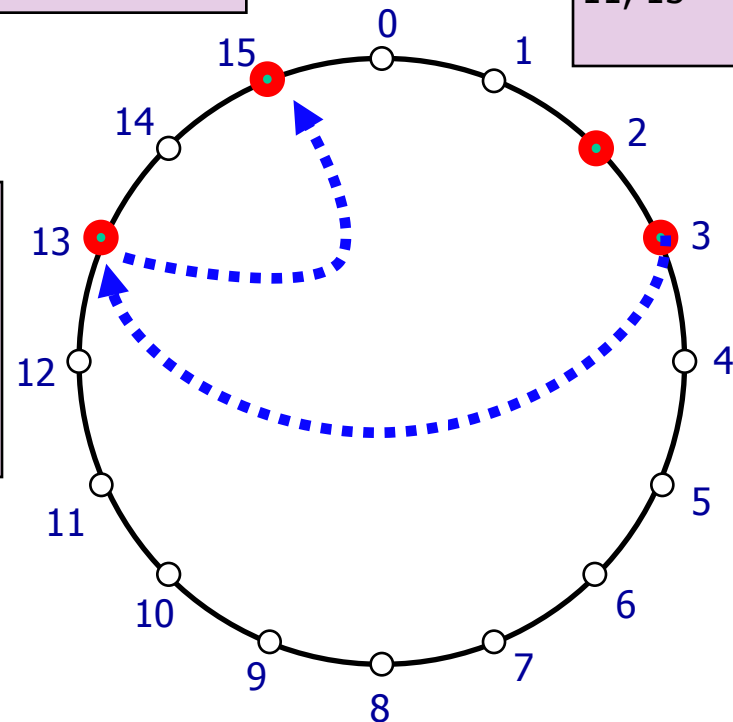
Routing table (3)

4, 13
5, 13
7, 13
11, 13

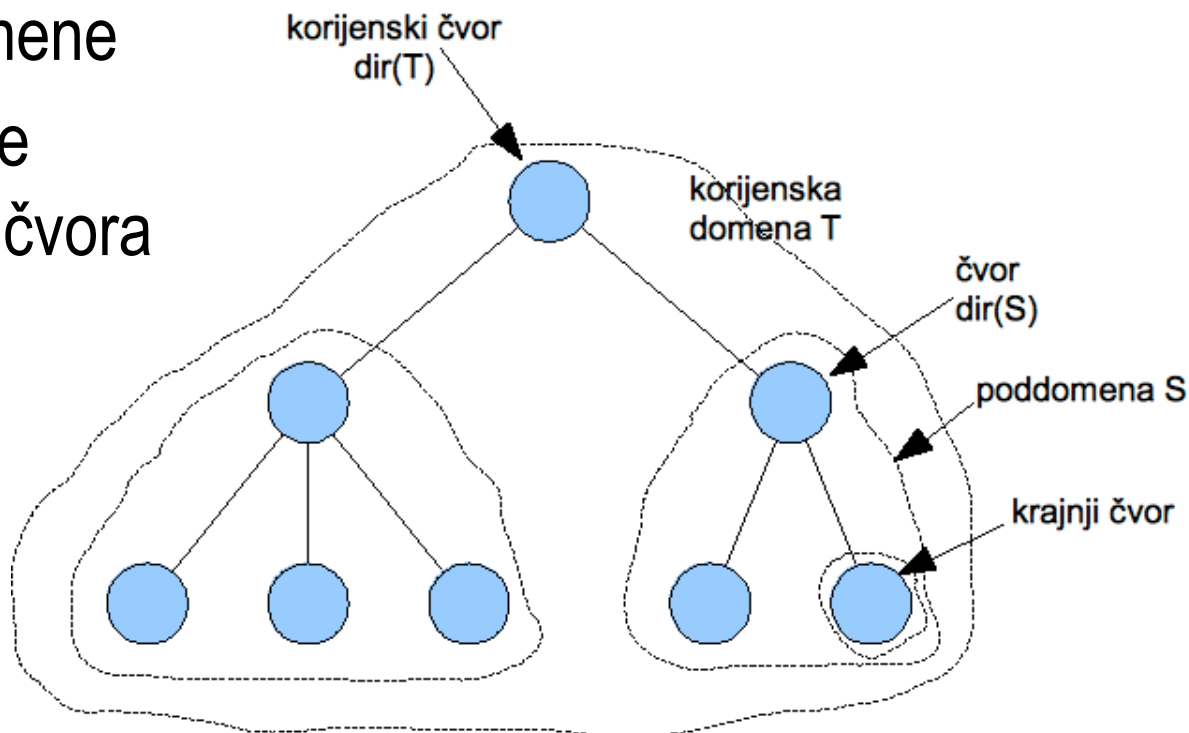
Routing table (13)

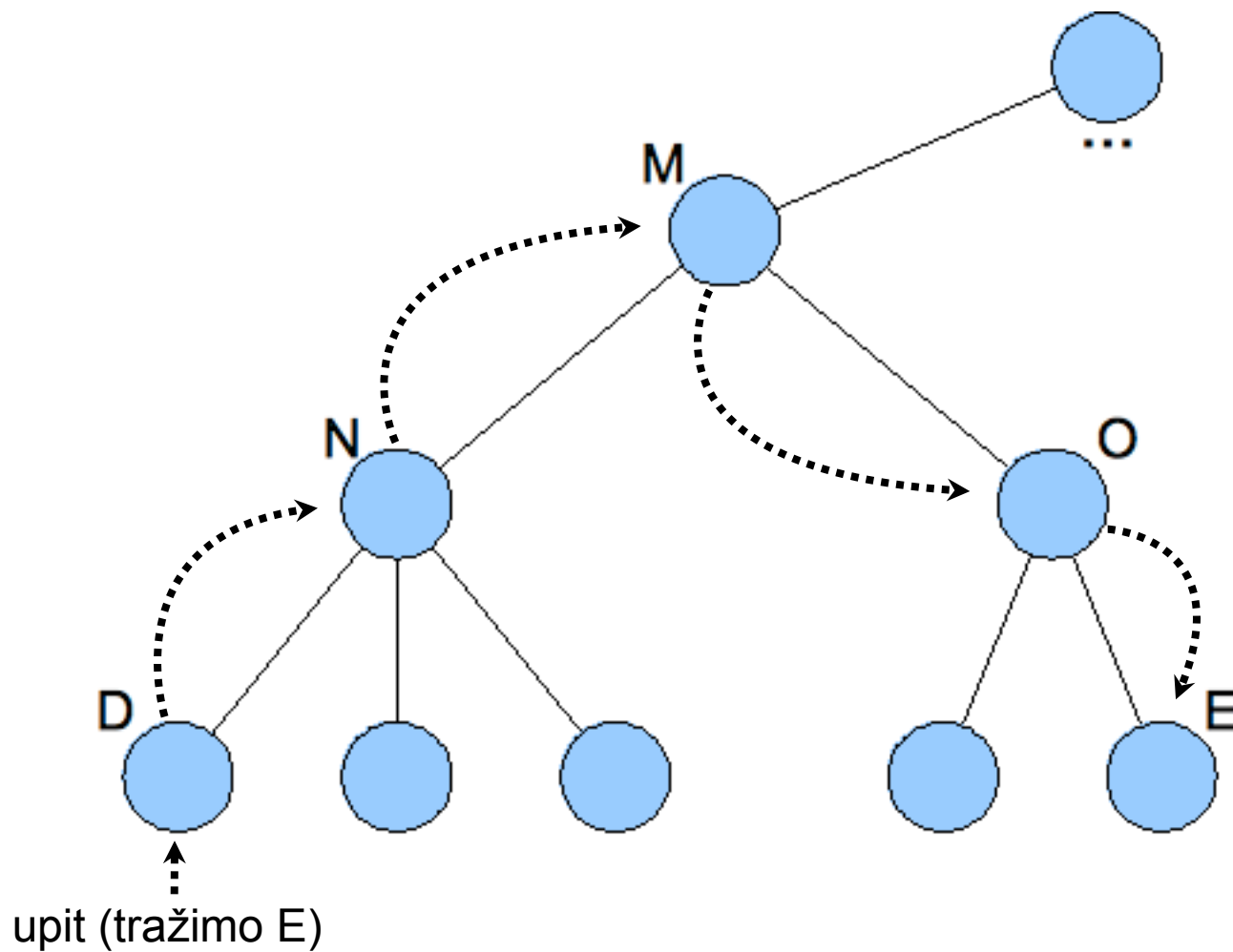
14, 15
15, 15
1, 2
5, 13

čvor 3 usmjerava upit od čvora 13 (id 14 je veći od 11!)
čvor 13 usmjerava upit do čvora 15



- ◆ mreža je podijeljena u domene
- ◆ postoji jedna glavna domena
 - korijenski čvor (direktorij)
- ◆ ostalo su poddomene
- ◆ najniža domena je domena krajnjeg čvora





- ◆ nestrukturirana imena su dobra za računala, ali su loša sa ljude
- ◆ strukturirana imena su dobra za ljude (*human readable*)
 - nazivi računala (imena u DNS-u, ne IP adrese)
 - nazivi direktorija i datoteka
 - URI (*Uniform Resource Identifier*)
 - ...
- ◆ ima definiranu strukturu imena

◆ Hijerarhijski niz komponenata:

- **shema** (engl. *scheme*)
- **autoritet** (engl. *authority*)
- **put** (engl. *path*)
- **upit** (engl. *query*)
- **fragment** (engl. *fragment*).

URI = **scheme** ":" hier-part ["?" **query**] ["#" **fragment**]

hier-part = "/" / **authority** path-abempty
/ path-absolute
/ path-rootless
/ path-empty

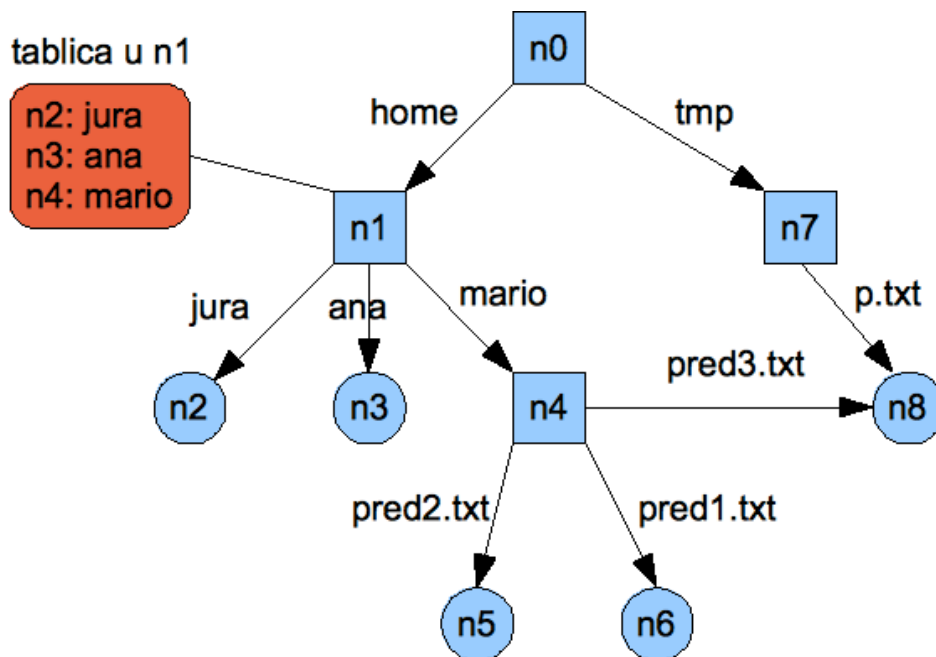
- ♦ imena su organizirana u prostoru imena
- ♦ prostor imena se može prikazati kao označeni usmjereni graf
- ♦ vrste čvorova u grafu:

- krajnji čvorovi

- predstavlja ime entiteta i sadrži informacije o njemu

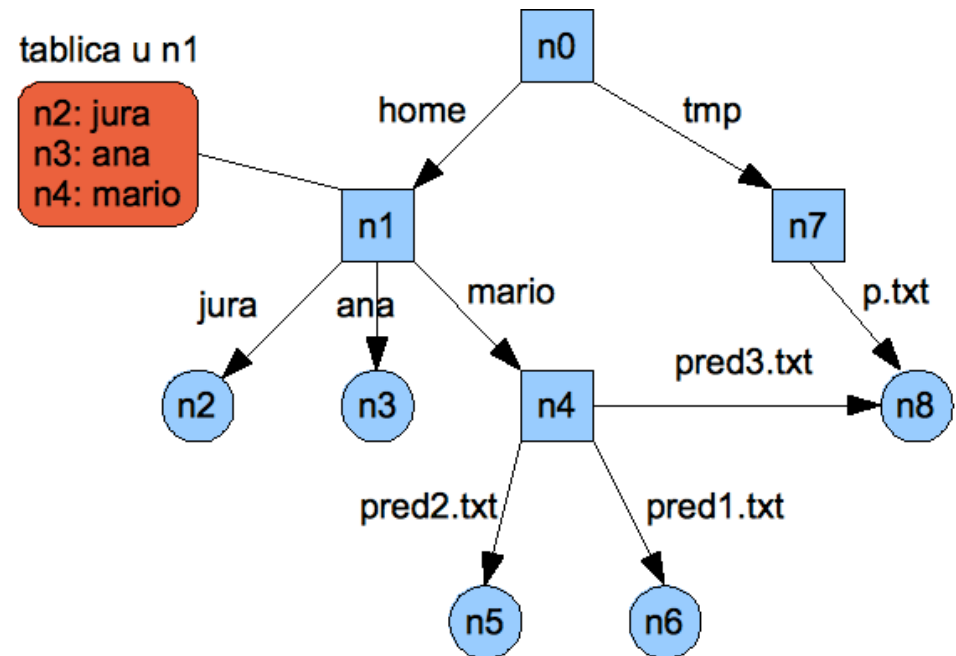
- direktoriji

- ima odlazne veze označene imenom
 - sadrži tablicu oznaka i naziva čvora za svaku odlaznu vezu



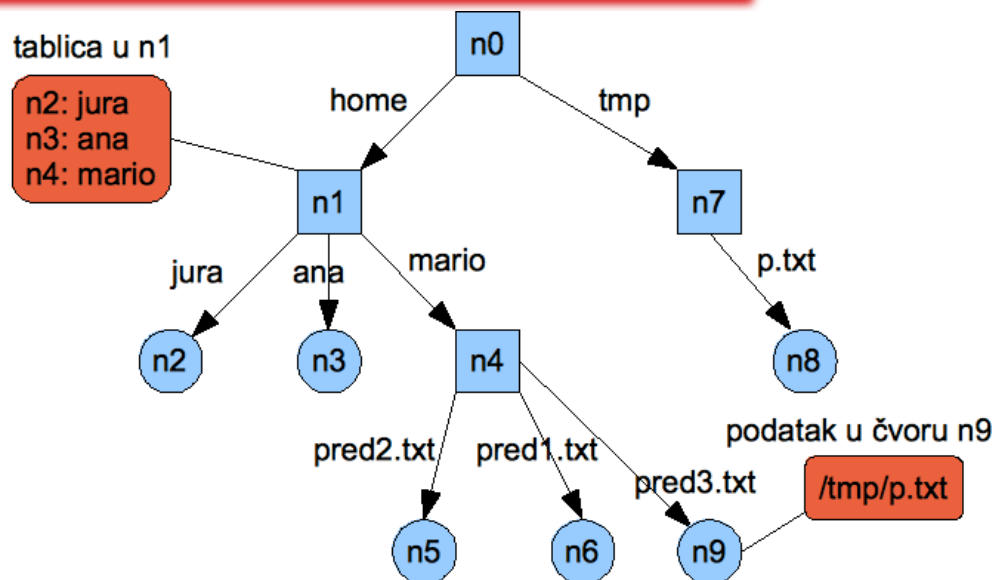
- ◆ svaki put unutar grafa se može zapisati kao niz oznaka na vezama
 - N: <oznaka-1, oznaka-2, ... , oznaka-n>
 - N je čvor od kojeg se kreće
- ◆ niz oznaka je put (*path name*)
- ◆ dvije vrste:
 - apsolutni put - ako počinje s korijenskim čvorom
 - relativni put - ako počinje s nekim drugim čvorom
- ◆ u datotečnom sustavu se put prikazuje nizom znakova
 - oznake su odijeljene specijalnim znakom / ili \
 - ako je početak puta s tim znakom onda je put apsolutan

- ◆ proces pronalaženja čvora iz zadanog puta
- ◆ da bismo mogli naći čvor potreban je početni čvor
 - trenutni direktorij kod datotečnog sustava
- ◆ pogledajmo čvor n8
 - ima dva puta do njega
 - /tmp/p.txt
 - /home/mario/pred3.txt
- ◆ čvor n8 ima drugo ime (*alias*)
- ◆ na slici imamo tvrde veze (*hard link*) prama n8



- ◆ na slici imamo simboličku vezu prema čvoru n8

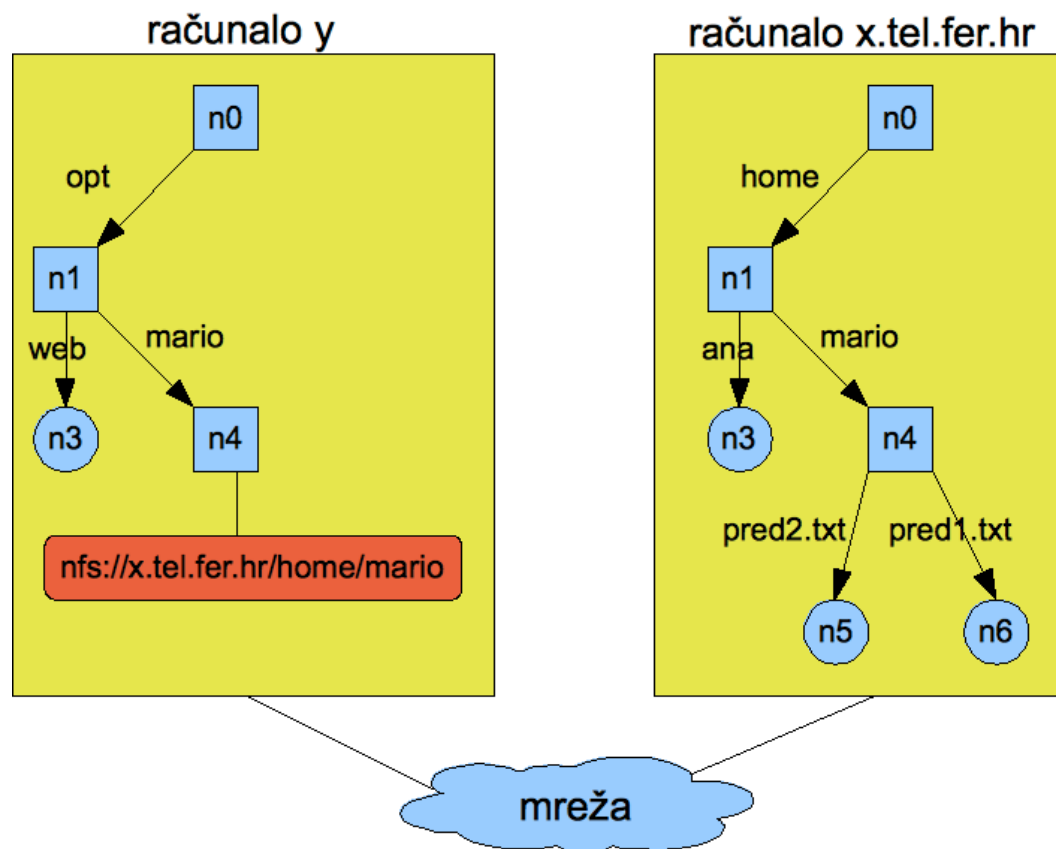
- stvoren je čvor n9 koji ima zapisan put do n8



- ◆ povezivanje više prostora imena u jedan

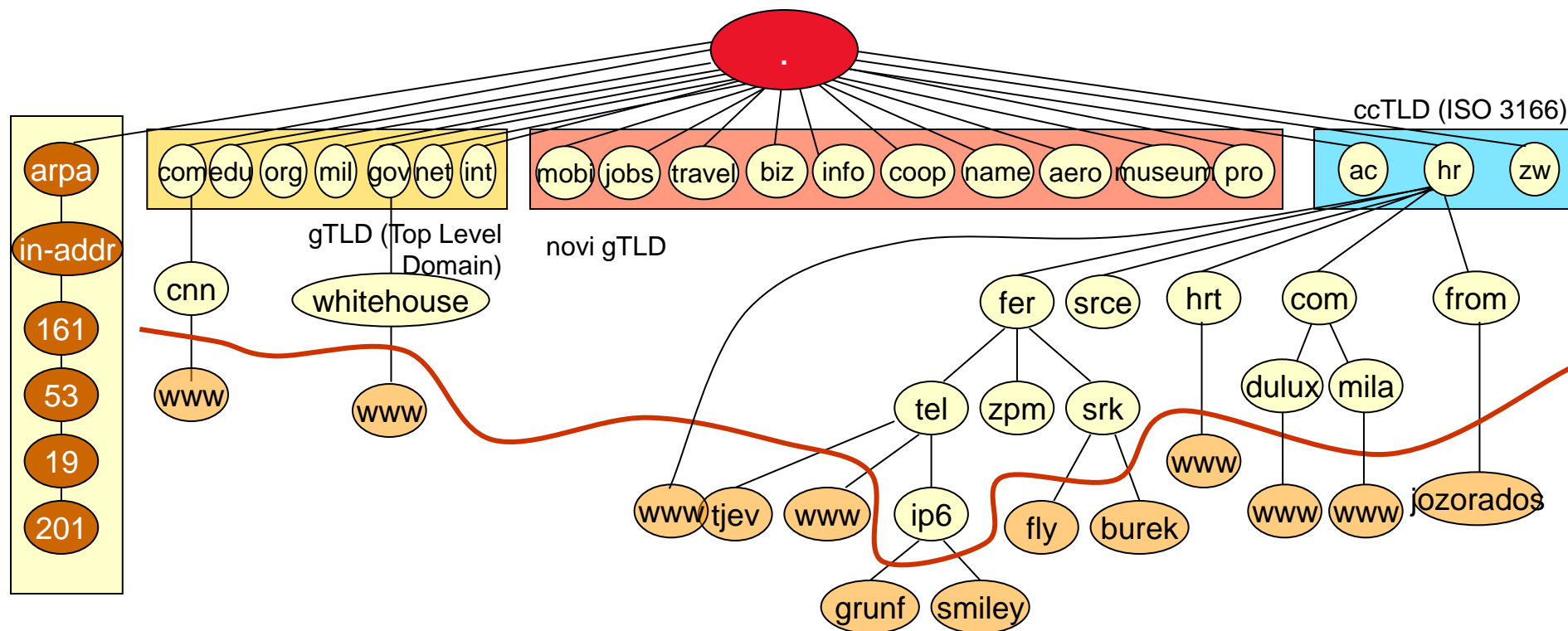
- *mountanje* - kada u jednom grafu na nekom mjestu imamo čvor čiji je identifikator iz drugog grafa (prostora imena)
- čvor koji zadrži vezu na čvor u drugom grafu se zove *mount point*
- čvor u drugom grafu se zove *mounting point*
- obično je *mounting point* korijenski čvor drugog grafa

- ◆ *NFS - Network File System*
- ◆ povezivanje dva grafa na dva različita računala
 - trebamo: pristupni protokol, poslužitelj, ime *mounting pointa*



- ◆ prostori imena u velikim sustavima su raspodijeljeni
- ◆ primjer takvog sustava je DNS (*Domain Name System*)
- ◆ obično je sustav podijeljen na slojeve:
 - globalni sloj
 - u njima se nalaze najviši čvorovi (korijenski)
 - čvorovi su stabilni (rijetko se mijenjaju podaci)
 - moraju biti visoko dostupni (*high availability*)
 - predstavljaju organizacije ili grupe (korijenske domene u DNS-u)
 - administrativni sloj
 - upravljani od strane jedne organizacije
 - relativno stabilni čvorovi (npr. fer.hr)
 - upravljački sloj (*managerial layer*)
 - čvorovi su nestabilni tj. često se mijenjaju podaci (npr. tel.fer.hr)
 - njima upravljaju administratori i krajnji korisnici

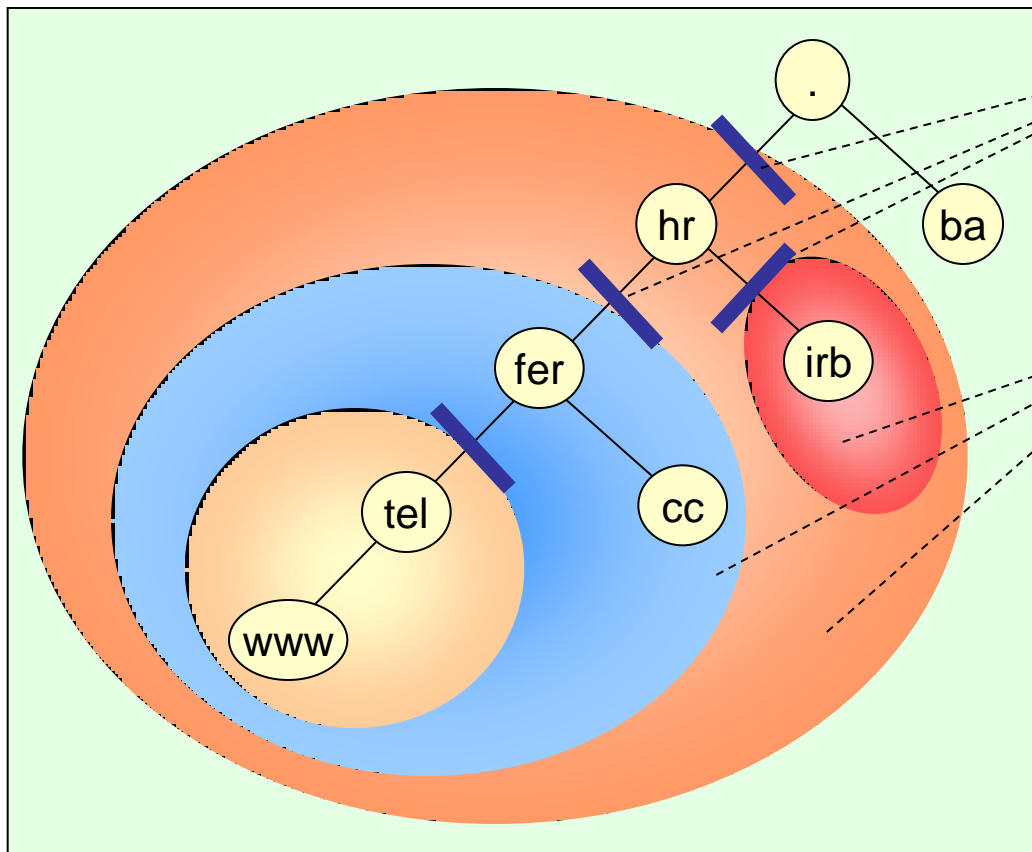
- ◆ Prostor domenskih imena i zapisi o resursima (*resource records, RR*)
 - specifikacija stablaste strukture domenskog prostora gdje svaki čvor ima pridružene podatke (adresu, ime i slično)
- ◆ Poslužitelji imena (DNS poslužitelji)
 - programi koji održavaju informacije o strukturi svog domenskog prostora, a imaju i popis ostalih poslužitelja kojima će proslijediti upite na koje ne znaju odgovor
 - poslužitelji imaju nadležnost nad zonama
- ◆ Razlučitelji (*resolvers*)
 - programi koji u ime klijenta šalju upit DNS poslužitelju i interpretiraju njegove odgovore



postoji 13 vršnih DNS poslužitelja koji imaju pregled cijele mreže

neke zemlje unutar svoje hijerarhije primjenjuju gTLD sustav

- com.au, edu.au, gov.au, net.au, org.au
- ac.uk, co.uk, govt.uk, net.uk, org.uk
- <http://www.root-servers.org/>



Delegacija

dodjeljivanje odgovornosti za podmrežu nekoj drugoj organizaciji

Zona

dio domenskog prostora o kojem domenski poslužitelj ima potpunu informaciju

Nadležnost

poslužitelj je nadležan za jednu ili više zona o kojima ima sve informacije

Razlučivanje

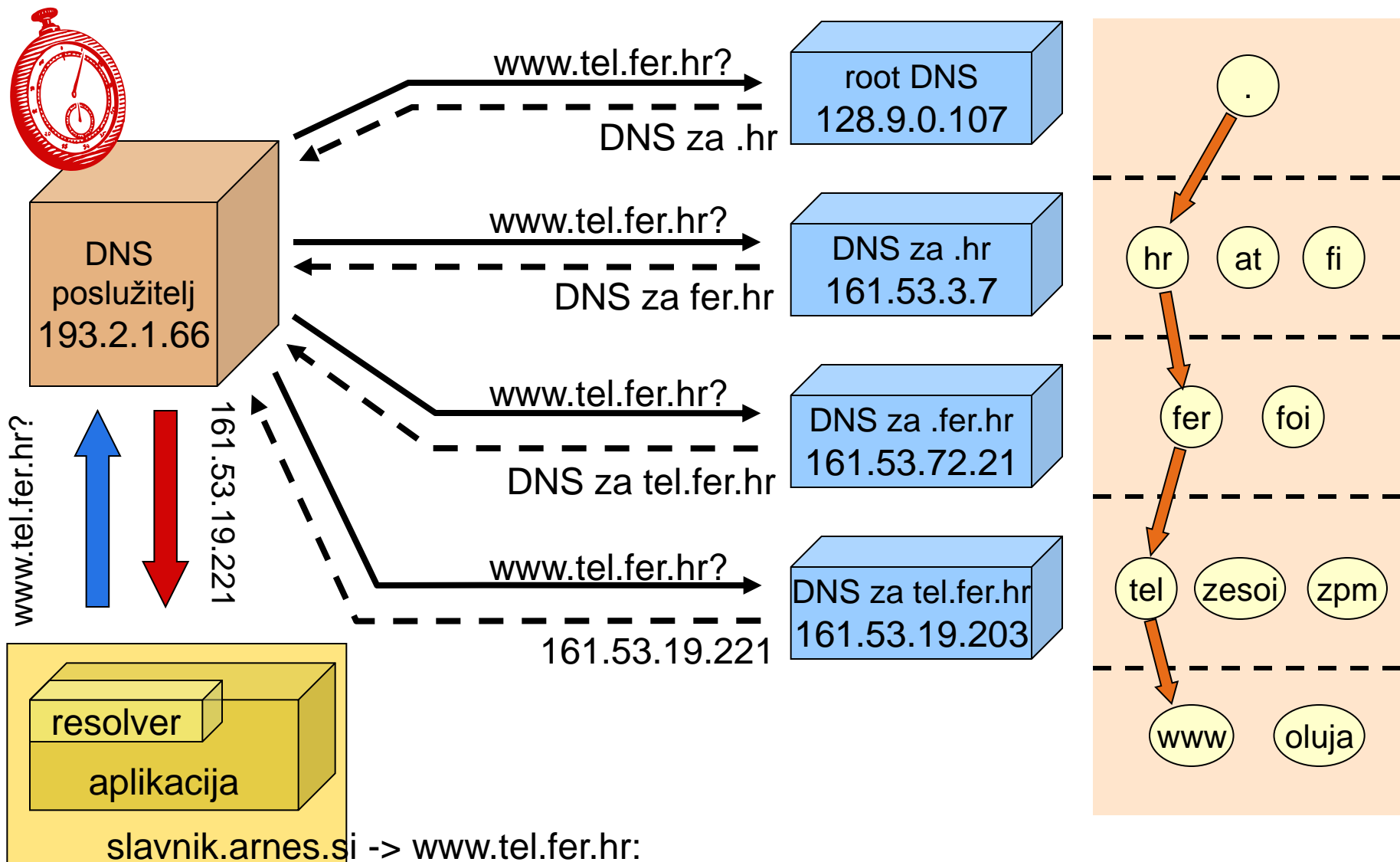
dohvaćanje podataka o imeničkom prostoru

- ◆ osnovna zadaća - odgovoriti na upit klijenta
- ◆ format DNS poruka - RFC 1035
- ◆ načini rada:
 - iterativni (nerekurzivni)
 - poslužitelj odgovara samo na osnovu informacije koju posjeduje
 - vraća grešku, odgovor ili pokazivač na drugi, “bliži” poslužitelj
 - rekurzivni
 - vraća ili grešku ili odgovor, ali nikad ne vraća pokazivače na druge DNS poslužitelje
 - o načinu rada dogovaraju se klijent i poslužitelj
 - ako je poslužitelj spreman raditi u rekurzivnom načinu rada, postavlja odgovarajući bit u zaglavlju

Iterativni način rada



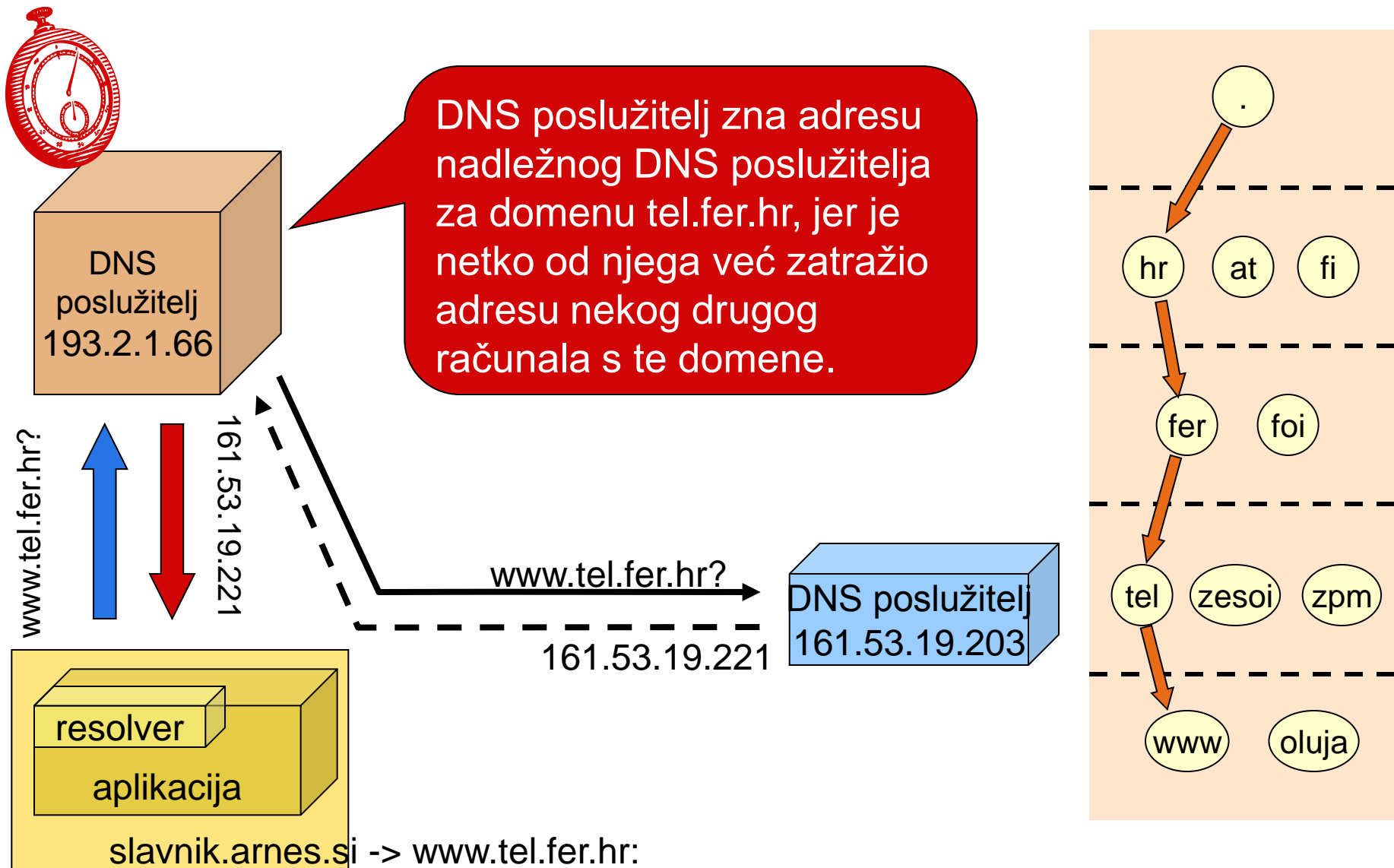
Zavod za telekomunikacije



Caching nadležnog poslužitelja



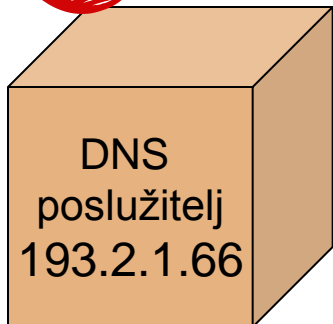
Zavod za telekomunikacije



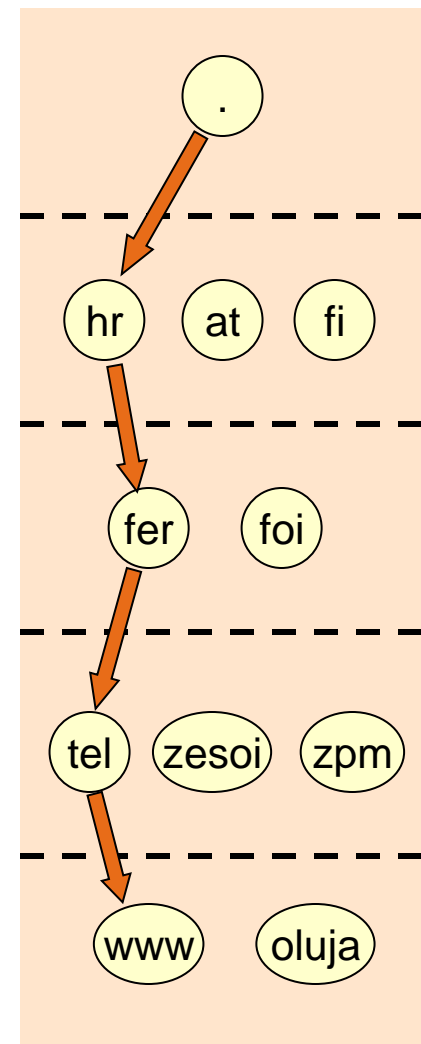
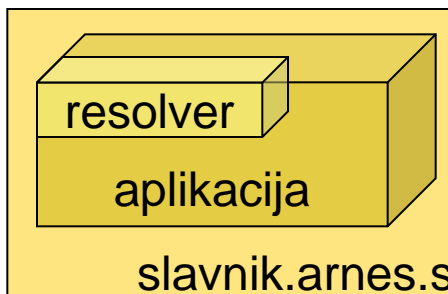
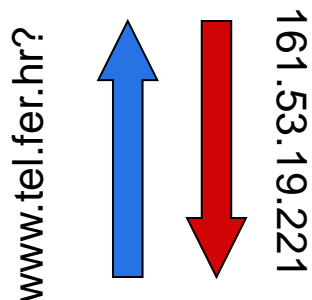
Caching traženog računala



Zavod za telekomunikacije



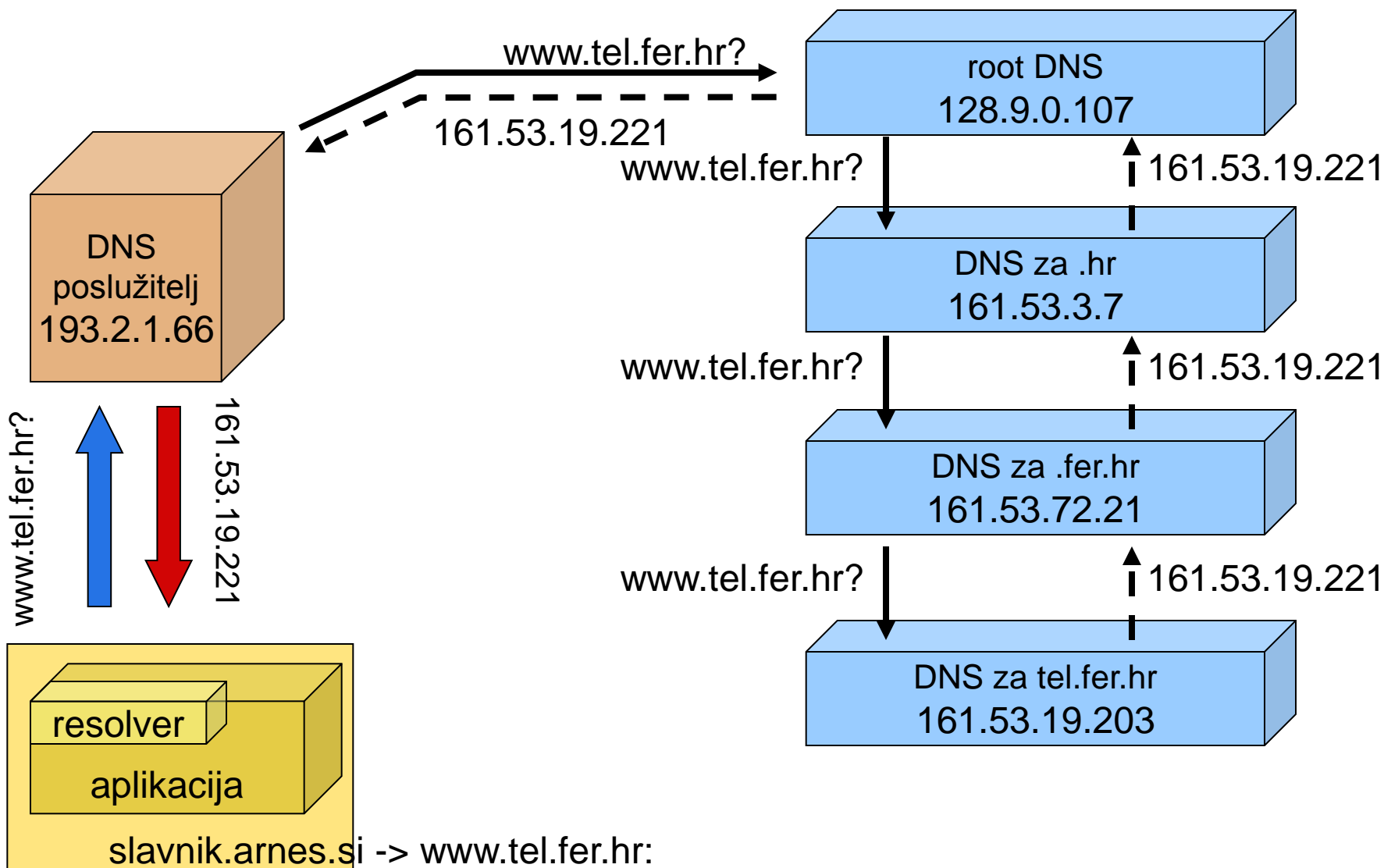
DNS poslužitelj zna adresu računala `www.tel.fer.hr` jer je netko drugi već pristupao tom računalu i tražio njegovu IP adresu.



Rekurzivni način rada



Zavod za telekomunikacije



- ◆ svaki entitet ima skup parova: atribut, vrijednost
- ◆ svaki atribut nešto “govori” o entitetu
- ◆ može se pretraživati entitete pomoću atributa
- ◆ ovi sustavi se još nazivaju i direktorijske usluge
- ◆ tipični primjeri ovakvih sustava su:
 - LDAP (*Lightweight Directory Access Protocol*)
 - kombinira strukturno i atributno imenovanje
 - temelji se na direktorijskim uslugama OSI X.500
 - RDF (*Resource Description Framework*)
 - proširuje vezu na triplete: subjekt, predikat, objekt (npr. Osoba, ime, Krešimir)
 - koristi se u semantičkom webu
 - UDDI (Universal Description Discovery and Integration)
 - za pretraživanje i pronalaženje usluga weba

- ◆ sadrži unose koji se smatraju direktorijskim zapisom
- ◆ direktorijski zapis se može usporediti sa zapisom u DNS-u
- ◆ svaki zapis se sastoji od skupa parova (atribut, vrijednost)
- ◆ svaki atribut ima definiranu vrstu podataka
- ◆ postoje atributi koji mogu imati više vrijednosti
 - koriste se kao polja ili liste
- ◆ shema definira elemente i podatke koji se zapisuju u LDAP
- ◆ svaki zapis ima jedinstveni identifikator: DN (*Distinguished Name*)
- ◆ svaki atribut ima RDN (relativni DN)
- ◆ operacije: dodavanje novog zapisa, brisanje zapisa, promjena zapisa, premještanje zapisa (promjena dn-a)
- ◆ obično se koriste za puno čitanja i malo pisanja

◆ primjena:

- pristup *Microsoft Active Directoryu*
- kod ISP-ova za popis i autorizaciju korisnika (CARNet - AAI@EduHr)
- kao baza podataka za mrežne i računalne resurse
 - npr. popis pisara u nekoj organizacijskoj jedinici
- kao baza podataka o strukturi, organizaciji i djelatnicima u nekoj tvrtci
 - npr. imena i prezimena s telefonskim brojevima i *email* adresama

◆ najpoznatije implementacije:

- OpenLDAP (<http://www.openldap.org>) – otvoreni kod
- IBM, Oracle, Netscape, ... - komercijalno

Primjer jednog LDAP zapisa



Zavod za telekomunikacije

- ◆ zapis je u LDIF (*LDAP Data Interchange Format*) formatu:

dn: cn=Mario Kušek,ou=zzt,dc=fer,dc=hr

cn: Mario Kušek

givenName: Mario

sn: Kušek

telephoneNumber: +385 1 6129 801

telephoneNumber: +1 888 555 1232

mail: mario.kusek@fer.hr

manager: cn=Ignac Lovrek,ou=zzt,dc=fer,dc=hr

Legenda:

dn - Distinguished Name

cn - Common Name

ou - Organizational Unit

dc - Domain Component

sn - Surname