

Рачунарске мреже

Александар Картељ

kartelj@matf.bg.ac.rs

Наставни материјали су преузети од: TANENBAUM, ANDREW S.; WETHERALL, DAVID J., COMPUTER NETWORKS, 5th Edition, © 2011
и прилагођени настави на Математичком факултету, Универзитета у Београду.

Slide material from: TANENBAUM, ANDREW S.; WETHERALL, DAVID J., COMPUTER NETWORKS, 5th Edition, © 2011.

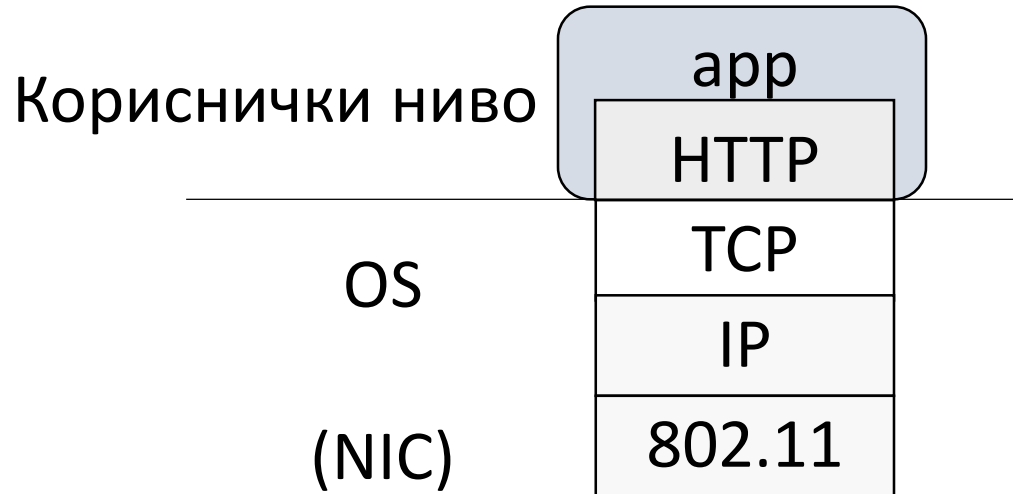
Electronically reproduced by permission of Pearson Education, Inc., Upper Saddle River, New Jersey

Апликативни слој

Преглед

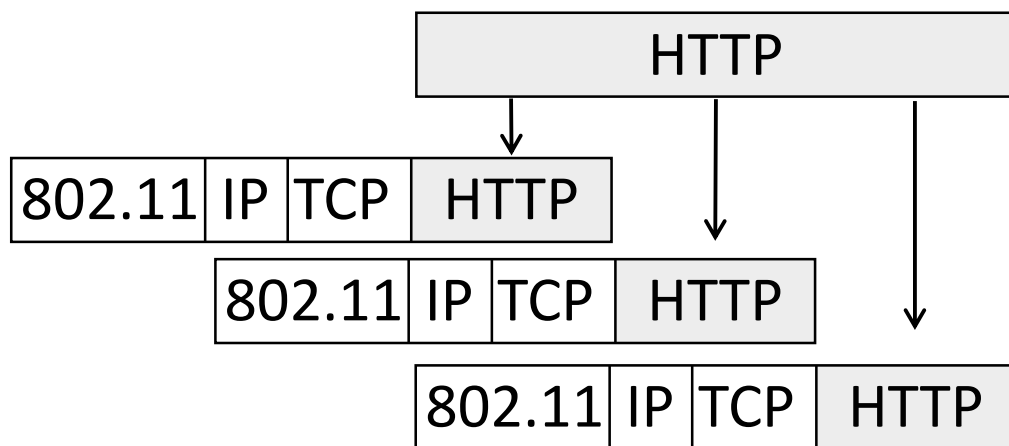
Апликативни слој

- Стигли смо до врха 😊
- Протоколи апликативног слоја су често део апликације (програма)
 - Немају нужно GUI, нпр. DNS који ћемо ускоро изучити



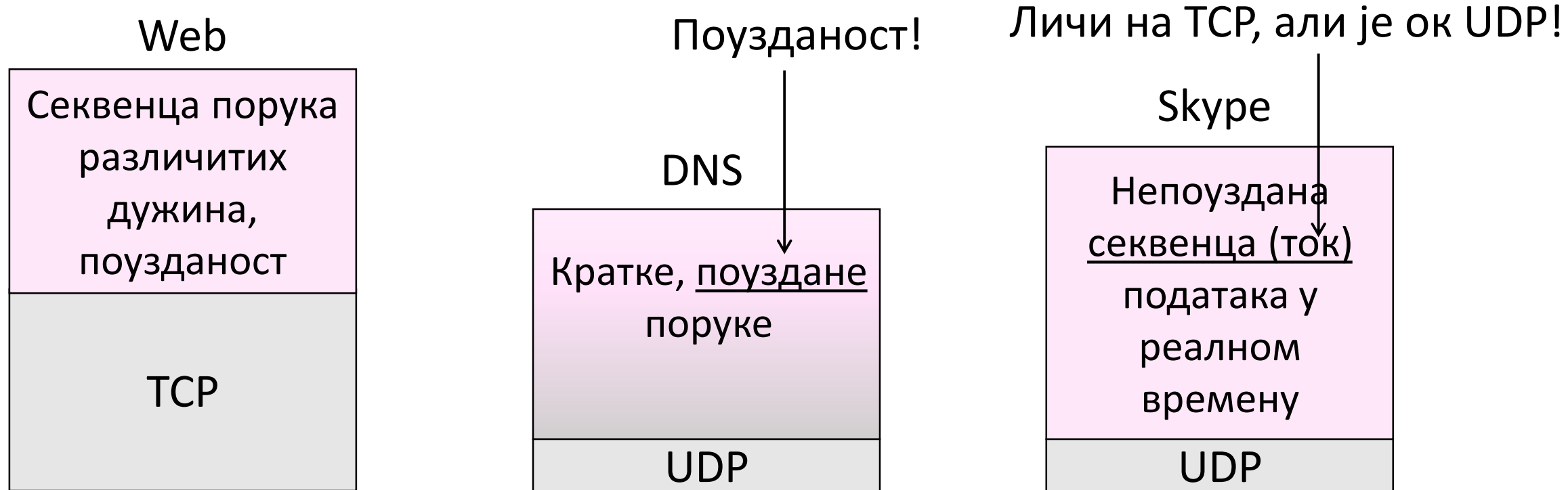
Апликативни слој (2)

- Поруке апликативног слоја се често деле на сегменте, а ови даље обично припадају јединственом пакету
 - Дакле, порука апликативног слоја је у више IP пакета ...

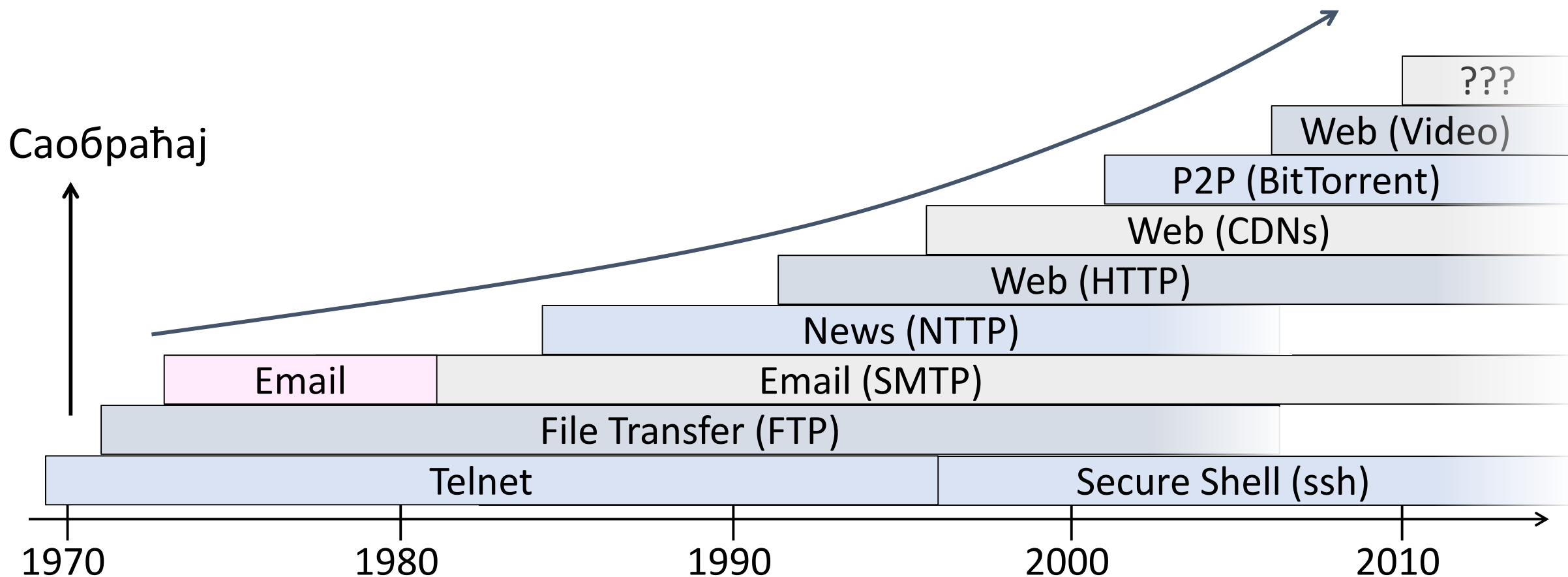


Интеракција са транспортним слојем

- Апликацијама треба „свашта“; нешто од захтевних функционалности и не постоји на транспортном слоју



Еволуција Интернет апликација



Теме за следећи пут

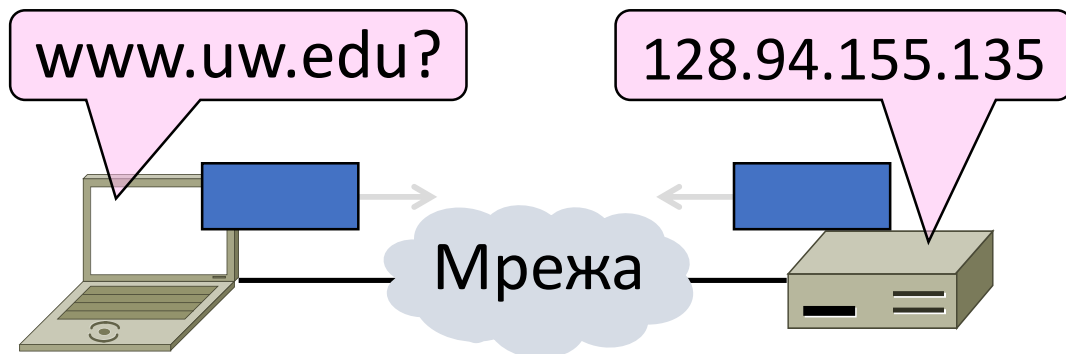
- DNS – пресликавање имена на IP адресе
- HTTP – протокол за WWW
- Веб проксији и кеширање
- CDN – брза испорука и репликација садржаја
- Peer-to-peer (BitTorrent) – систем равноправних чворова

Апликативни слој

DNS - Систем за пресликавање имена на IP адресе
(енг. Domain Name System)

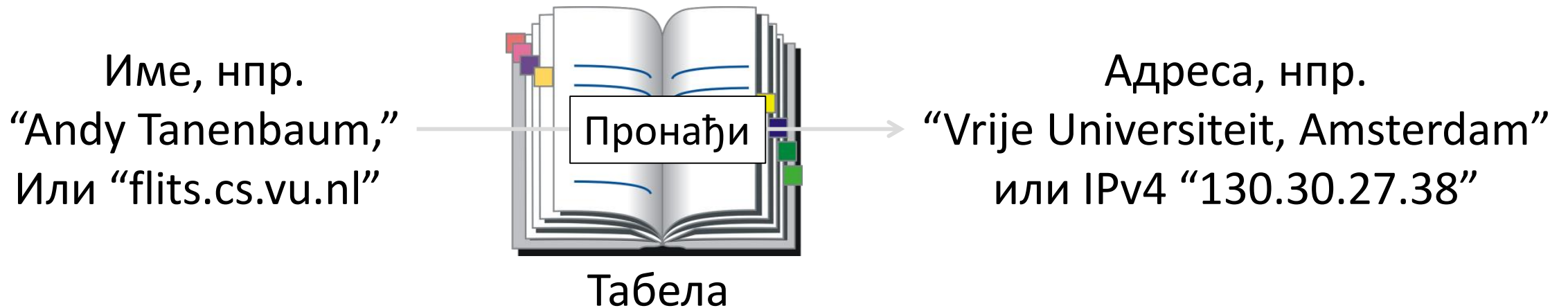
Тема

- Хоћемо да приступимо страници на Интернету путем имена, а не путем IP адресе. Зашто је тако боље?
 - У пракси се уместо име, чешће користи термин домен
1. Организација простора имена
 - Неизводљиво да један рачунар може да садржи све информације...
 2. Процес одређивања адресе за тражено име



Имена и адресе

- Имена су идентификатори ресурса (како се нешто зове?)
- Адресе су локатори ресурса (где се шта налази?)
- Одређивање адресе (енг. name resolution)
је пресликавање имена на адресу



Пре DNS – HOSTS.TXT

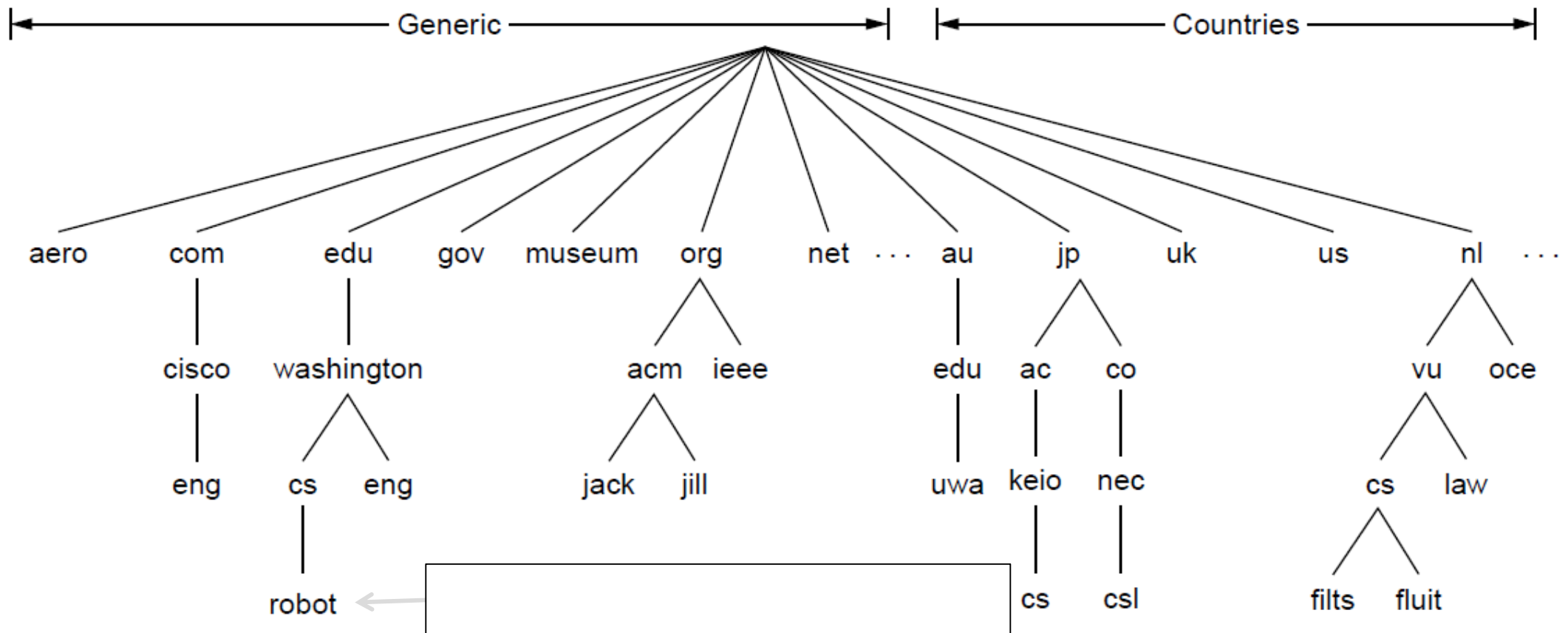
- Табела свих пресликавања на ARPANET-у се налазила у датотеци HOSTS.TXT коју су на дневној основи преузимали сви чворови са једног централног чвора
- Имена су иницијално била неструктурирана, а после су постала хијерархијска, нпр. lcs.mit.edu
 - Тачка раздваја ниво хијерархије
- Тешко за управљање и неефикасно како је ARPANET растао...

DNS

- Систем (апликација) која врши пресликавање имена у IP адресу
 - `www.uwa.edu.au` → `130.95.128.140`
- Циљеви:
 - Лак за управљање када постоји велики број корисника
 - Ефикасан
- Приступ:
 - Наравно, дистрибуирана табела, хијерахијских организована
 - Аутоматски протокол за повезивање делова хијерахије

DNS (2)

- Корени чвор „.“ (тачка) се обично изоставља



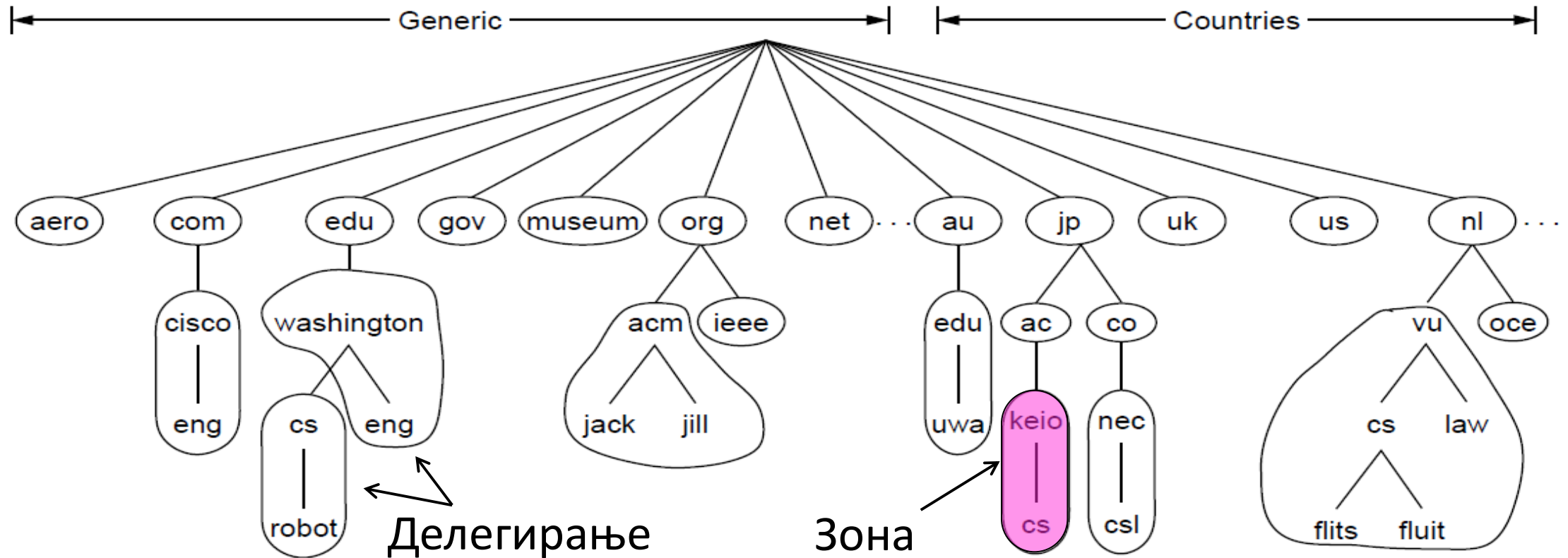
TLD – кровна имена (енг. Top-Level Domains)

- Одржава организација ICANN
(Internet Corp. for Assigned Names and Numbers)
 - Започела са радом 1998. године; финансијки и политички интереси...
- 22+ основна TLD
 - Иницијално 6 TLD: .com, .edu, .gov., .mil, .org, .net
 - Касније додати: .aero, .museum, .xxx, ...
- ~250 националних TLD
 - Двословни, нпр. “.au”
 - Многи „погодни“ су комерцијализовани, нпр .tv (Tuvalu)
 - Разна довијања: instagr.am (Armenia), goo.gl (Greenland), ...

DNS зоне

- Зона је непрекидно парче простора имена

- Делегирање је „подела“ на подзоне ради лакше и ефикасније организације



DNS зоне (2)

- Зоне су основ за даљу дистрибуцију
 - EDU регистар је надлежан за сва имена која се завршавају са .edu
 - UW надлежан за имена са суфиксом washington.edu
 - CS&E надлежан за суфикс cs.washington.edu
- Свака зона има надлежни сервер имена
 - Он зна контакте за делегирање, односно сервере имена за подзоне
 - Нпр. сервер имена за .edu зна за сервер имена зоне washington.edu

DNS слогови (енг. DNS records)

- Свака зона садржи подешавања (слоγοве) који пружају:
 - Информације о именованим рачунарима
 - Информације везане за слање поште
 - Информације о параметрима

Тип слога	Значење
SOA	Разни параметри зоне...
A	IPv4 адресе именованих рачунара
AAAA ("квад А")	IPv6 адресе именованих рачунара
CNAME	На који рачунар се шаље www, ftp итд.
MX	Имена рачунара који се користе за мејлове
NS	Сервер имена за зону

DNS слогови (2)

; Authoritative data for cs.vu.nl

cs.vu.nl.	86400	IN	SOA	star boss (9527,7200,7200,241920,86400)
cs.vu.nl.	86400	IN	MX	1 zephyr
cs.vu.nl.	86400	IN	MX	2 top
cs.vu.nl.	86400	IN	NS	star

← Сервер имена

star	86400	IN	A	130.37.56.205
zephyr	86400	IN	A	130.37.20.10
top	86400	IN	A	130.37.20.11
www	86400	IN	CNAME	star.cs.vu.nl
ftp	86400	IN	CNAME	zephyr.cs.vu.nl

← IP адресе
именованих
рачунара у
зони

flits	86400	IN	A	130.37.16.112
flits	86400	IN	A	192.31.231.165
flits	86400	IN	MX	1 flits
flits	86400	IN	MX	2 zephyr
flits	86400	IN	MX	3 top

rowboat		IN	A	130.37.56.201
		IN	MX	1 rowboat
		IN	MX	2 zephyr

← Мејл сервер

little-sister		IN	A	130.37.62.23
---------------	--	----	---	--------------

laserjet		IN	A	192.31.231.216
----------	--	----	---	----------------

Апликативни слој

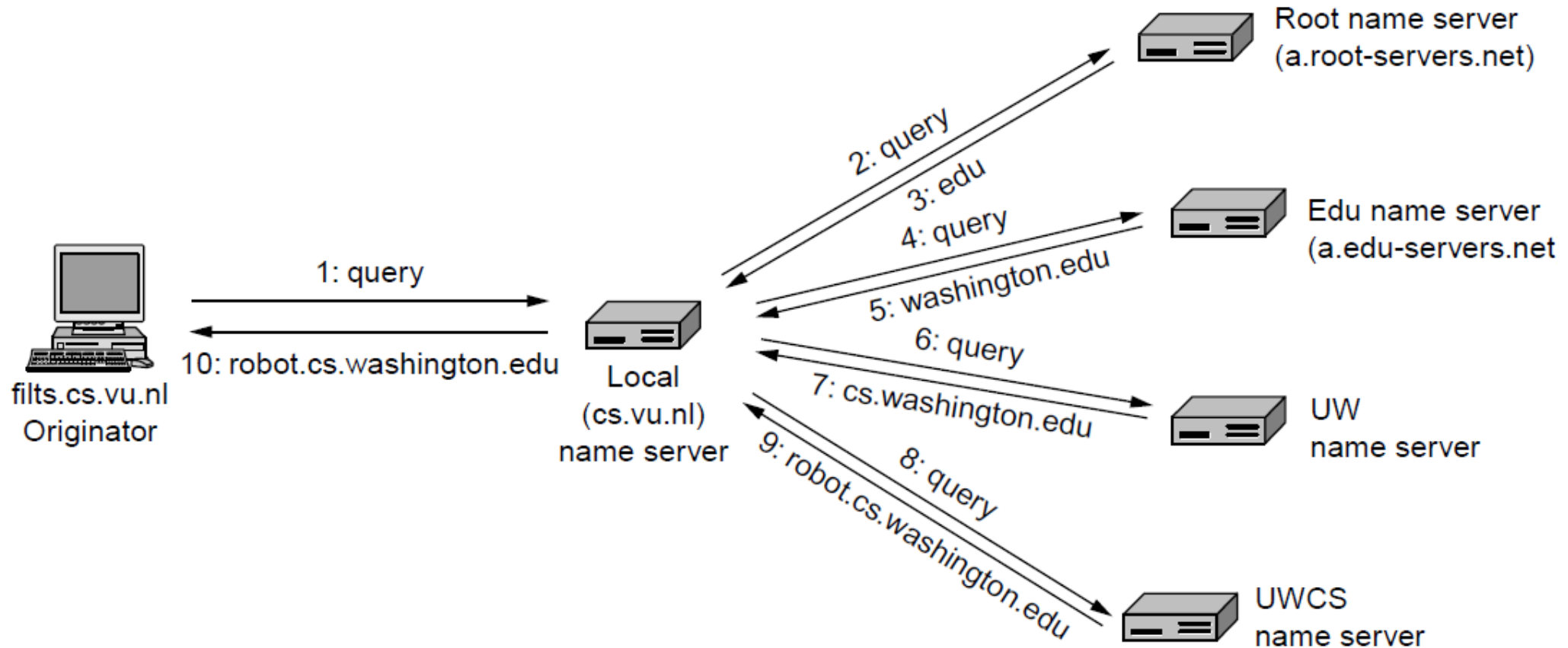
DNS одређивање адреса
(DNS resolution)

DNS одређивање адреса

- Неки рачунар (клијент) контактира DNS како би му овај послао IP адресу неког другог рачунара на Интернету
- Општа идеја је да:
 - Рачунар контактира свој најближи DNS (локални DNS)
Локални DNS, ако не зна тражено име од раније, започиње тражење са корених сервера имена (edu, com...)
 - Па се онда иде ниже кроз хијерахију...

DNS одређивање адреса (2)

- flits.cs.vu.nl захтева адресу за robot.cs.washington.edu



Итеративни и рекурзивни DNS

- Рекурзивни DNS

- Рекурзивни DNS завршава цео посао за клијента, тј. он налази IP адресу и враћа је клијенту
- У претходном примеру, то значи да локални DNS испоручује клијенту тражену адресу

- Итеративни DNS

- Ако не зна за име, итеративни само враћа референце на друге DNS-ове који знају одговор (и ти референтни могу бити итеративни или рекурзивни)
- У претходном примеру је нпр. локални био рекурзивни, корени су увек итеративни, и неко је на крају одговорио.

Итеративни и рекурзивни DNS (2)

- Рекурзивни
 - Смањује оптерећење ономе ко захтева информацију
 - Омогућава памћење (кеширање) за претходне упите, што може побољшати перформансе на страни онога ко пита
- Итеративни
 - Не захтева велику меморију, само памћење директних потомака (сервера имена нижег нивоа)
 - Једини избор уколико се DNS „бомбардује“ захтевима, као нпр. корени DNS сервери

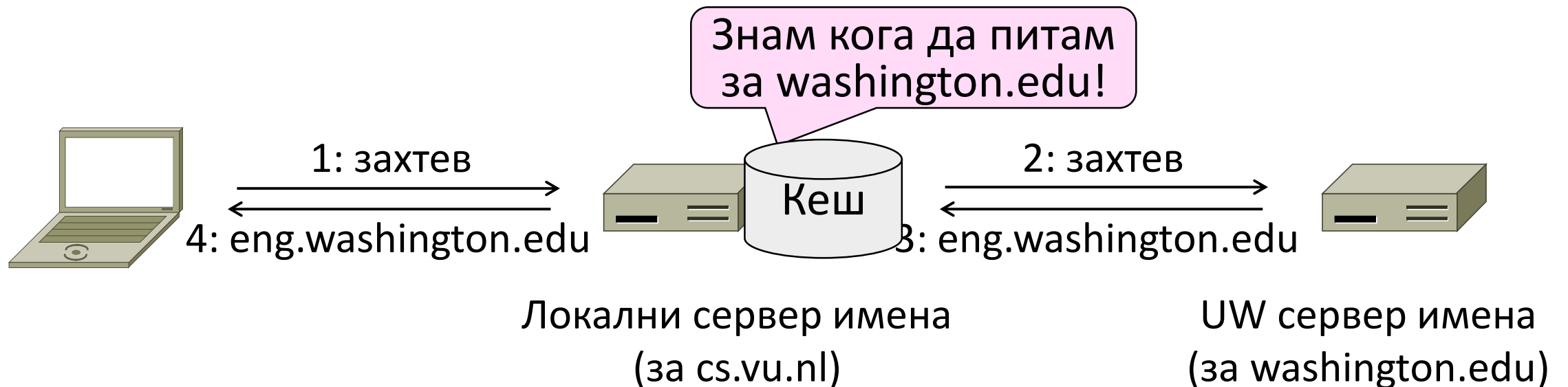
Кеширање

- Кашњење у одређивању имена треба да буде мало
 - Како не би предуго чекали за нпр. учитавање сајта
- Ако је име кеширано у DNS серверу, он одговара моментално, без питања других DNS
 - Информације о делу имена се такође могу искористити
 - Одговори од DNS у себи садрже TTL за кеширање



Кеширање (2)

- Ако flits.cs.vu.nl сада тражи eng.washington.edu
 - Локални DNS (ако је рекурзиван) ће знати добар део пута



Локални DNS сервер

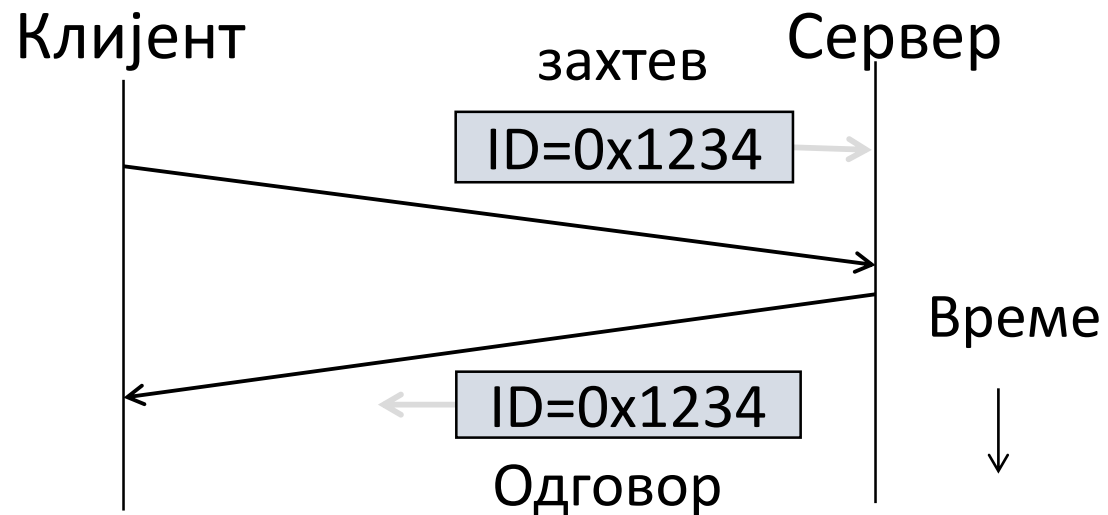
- Обично у власништву фирме или ISP
 - Али могу бити и на клијенту или AP (приступној тачки)
 - Постоје и јавно доступни нпр. Google јавни DNS
- Клијенти морају да знају ко им је локални DNS
 - Ово се обично подешава путем DHCP о истом трошку када се рачунару у мрежи шаље сопствена IP адреса

Корени DNS сервери

- Корени „.“ DNS чини заправо 13 сервера
 - a.root-servers.net до m.root-servers.net
 - Њихове IP адресе су фиксиране у подешавањима свих других DNS сервера
- Заправо има >250 реплицираних корених DNS сервера
 - Нпр. за a.root-servers.net може бити двадесетак различитих физичких машина које се мапирају на ово име (у зависности од близине онога ко захтева – anycast)
 - Врло поуздани и ефикасни

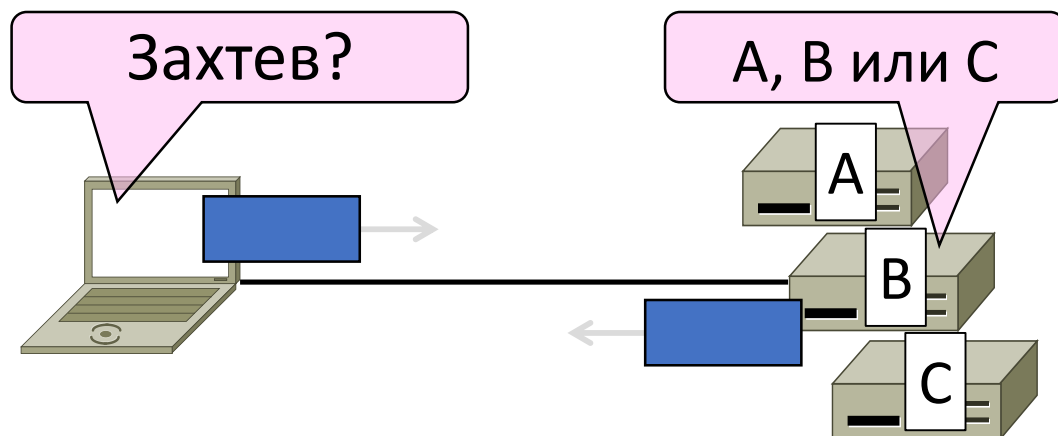
DNS поруке

- Захтеви и одговори (све стаје у један сегмент)
 - Користе UDP, порт 53
 - Користи ARQ за поузданост
 - Поруке се идентификују 16-битном ознаком



DNS поруке (2)

- Боље перформансе и поузданост помоћу реплика
 - Вишеструки DNS сервери за исту зону
 - Клијент се обраћа једном од њих;
па неком другом у случају отказа
 - Помаже и у балансирању протока

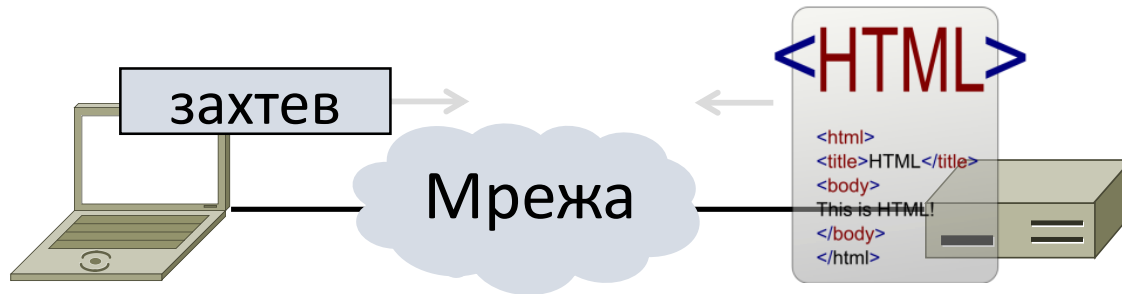


Апликативни слој

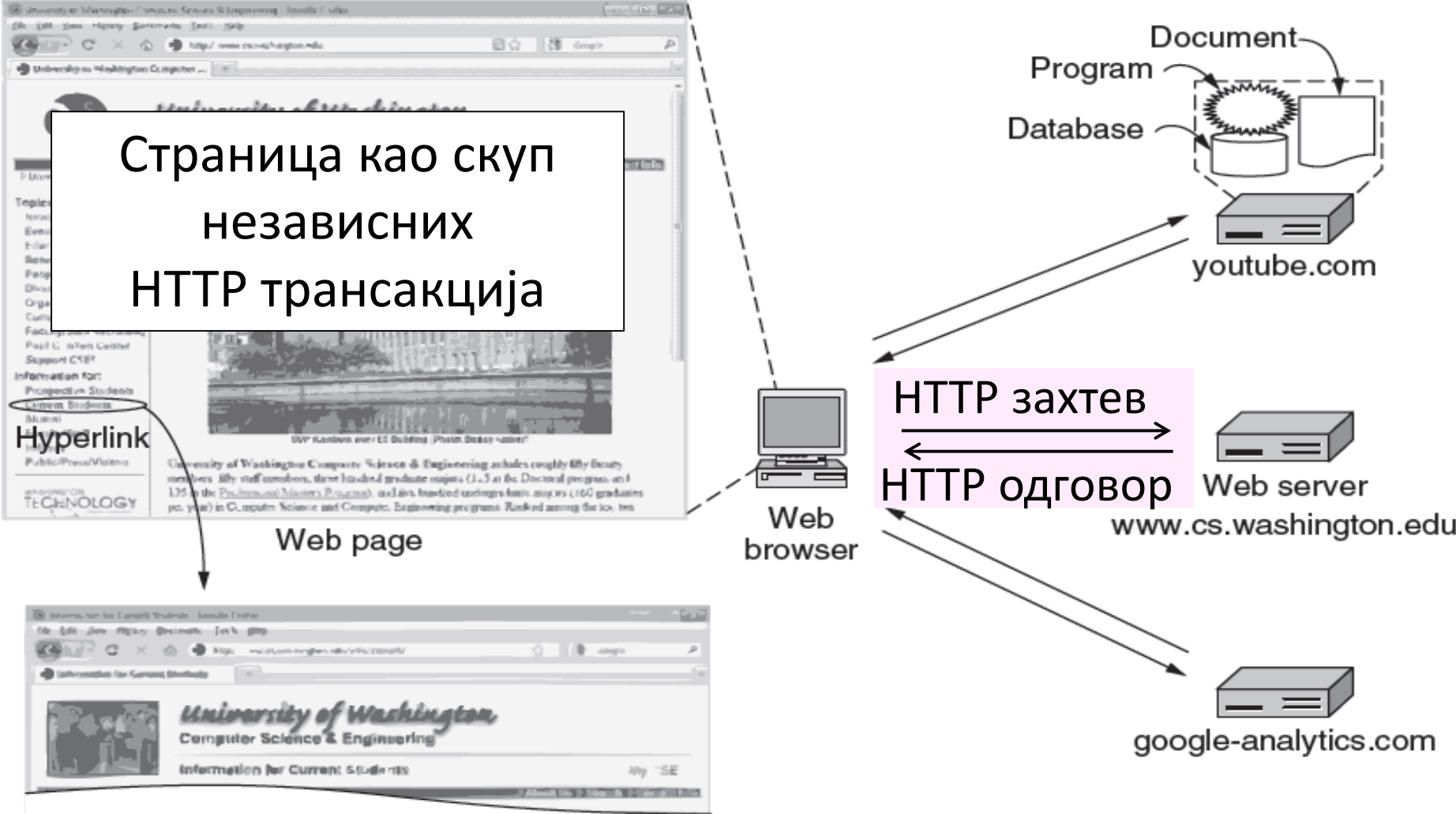
HTTP – протокол за преузимање Интернет докумената
(енг. HyperText Transfer Protocol)

HTTP

- Основни протокол за преузимање Интернет докумената
 - Интернет странице су такође вид документа
- Тачније би било WWW (или само Веб) документи
 - WWW – скуп свих повезаних докумената – логички ниво
 - Интернет се односи на скуп повезаних машина – физички ниво

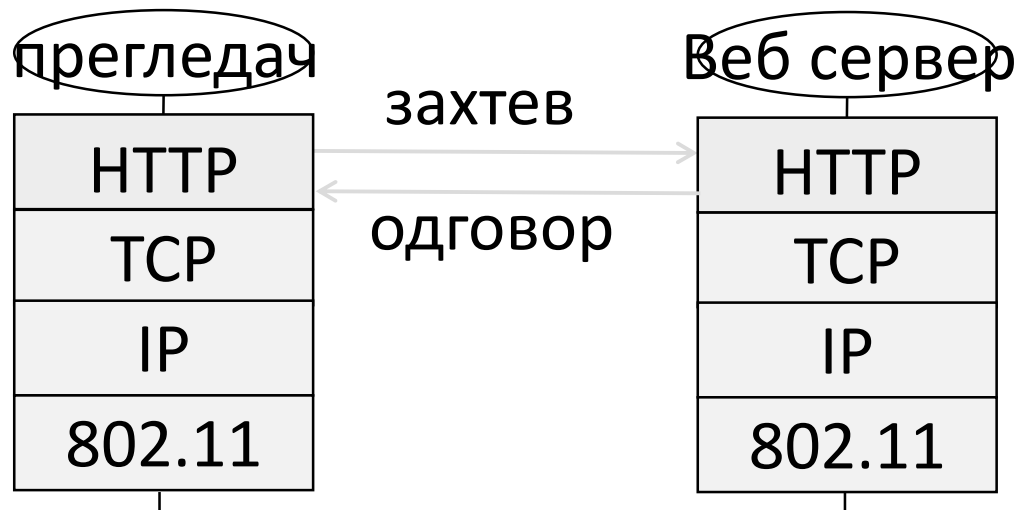


Веб страница



HTTP протокол

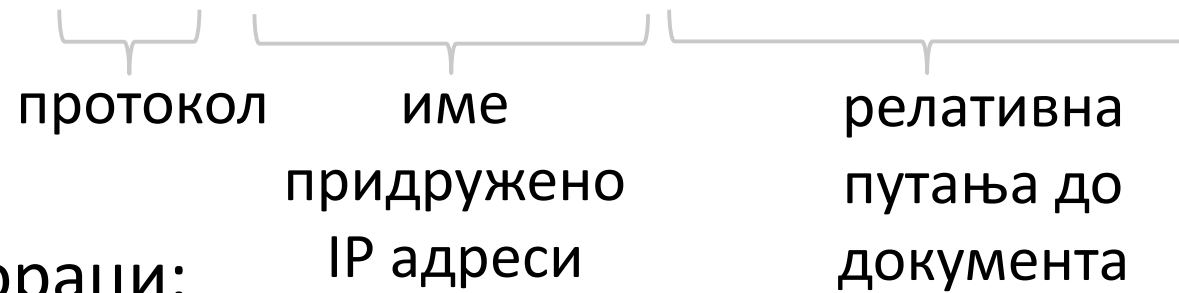
- Користи TCP, обично порт 80
- Користи се и на страни клијента (прегледач) и на страни сервера (Веб сервер)



Кораци у преузимању Веб документа

- Корисник, нпр. човек, укуца адресу Веб документа:

`http://en.wikipedia.org/wiki/Vegemite`

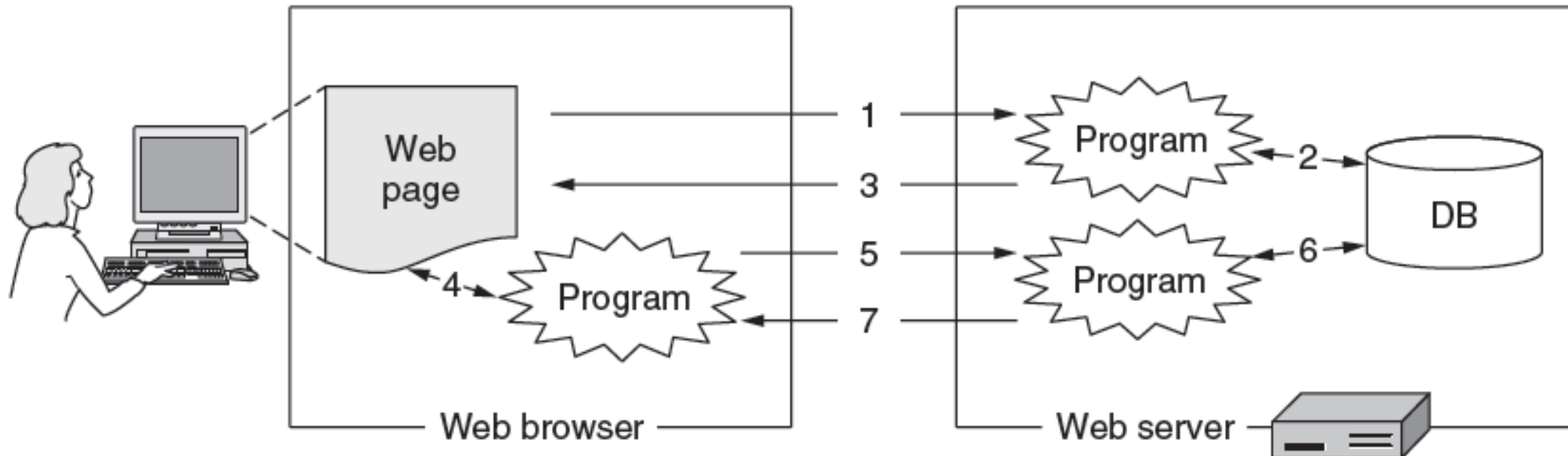


- Кораци:

- Одређивање IP адресе
- Успостављање TCP везе са сервером (машином на циљној IP адреси)
- Слање HTTP захтева за конкретним документом на серверу
- (Чекање HTTP одговора)
- Извршавање или преузимање уграђених докумената (слика, видеа, ...)
- Гашење TCP веза (или веза)

Статички и динамички Веб документи

- Статички Веб документи су датотеке са фиксним садржајем, нпр. слика, видео, статичка html страница, ...
- Динамички Веб документ је датотека добијена као резултат извршавања програма:
 - Нпр. html страна добијена као резултат претраге на Google.



HTTP протокол (2)

- Команде које се користе при формирању захтева

	Метода	Опис
Преузми документ →	GET	Чита и враћа садржај Веб документа
	HEAD	Чита заглавље Веб док.
Пошаљи податке →	POST	Додаје податке Веб док.
	PUT	Складишти Веб док.
	DELETE	Уклања Веб док.
	TRACE	Приказује долазни захтев
	CONNECT	Веза кроз прокси
	OPTIONS	Параметри захтева

HTTP протокол (3)

- Кодови који се враћају кроз одговор (пored самог одговора наравно)

Код	Значење	Примери
1xx	Информација	100 = сервер прихвата захтев
2xx	Успех	200 = захтев успео
3xx	Преусмерење	301 = документ померен
4xx	Грешка клијента	403 = забрањен приступ; 404 = нема документа
5xx	Грешка сервера	500 = интерна логичка грешка

HTTP протокол (4)

- Додатне заглавља захтева или одговора:

Функција	Примери заглавља
Инфо о прегледачу (клијент → сервер)	User-Agent, Accept, Accept-Charset, Accept-Encoding, Accept-Language
Инфо о кеширању (оба смера)	If-Modified-Since, If-None-Match, Date, Last-Modified, Expires, Cache-Control, ETag
Стање прегледача (клијент → сервер)	Cookie, Referer, Authorization, Host
Тип садржаја (сервер → клијент)	Content-Encoding, Content-Length, Content-Type, Content-Language, Content-Range, Set-Cookie

Апликативни слој

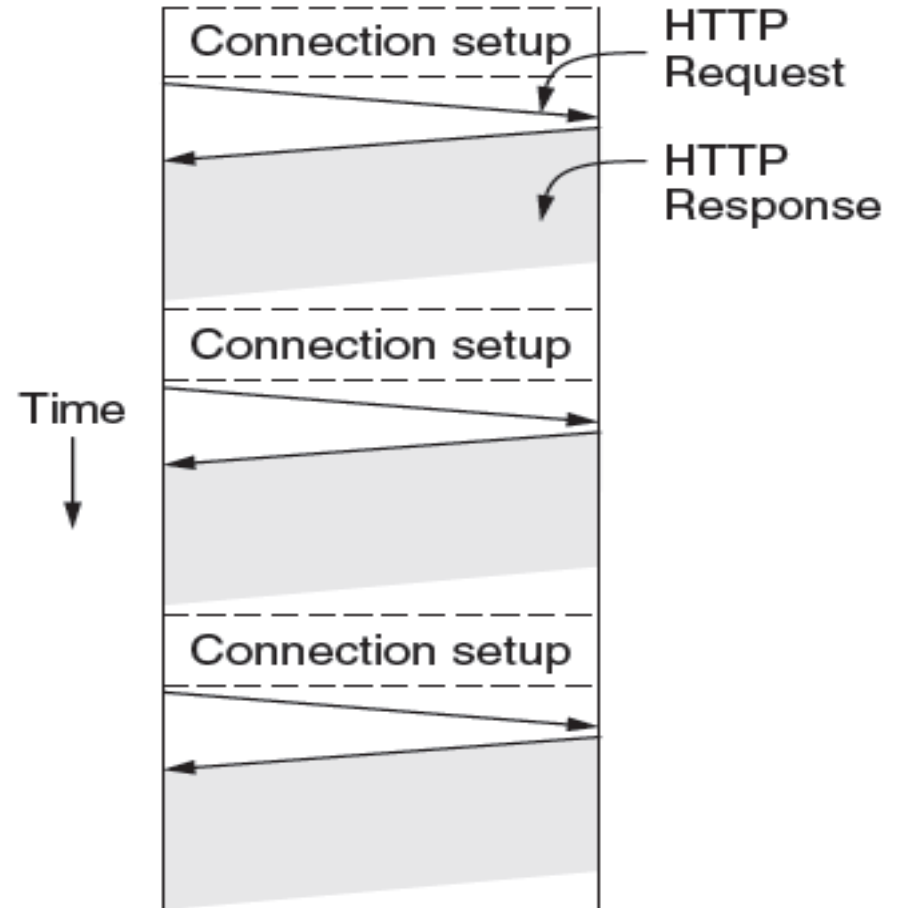
HTTP перформансе

PLT – време учитавања документа (енг. Page Load Time)

- PLT је кључна мера перформанси
 - Време протекло од клика до учитавања документа
 - Чак и мало повећање PLT има велик утицај на смањење задовољства корисника
- PLT зависи од много фактора
 - Структура документа
 - Верзије HTTP (и TCP!) протокола
 - Мрежног протока и RTT

Раније перформансе (2)

- HTTP/1.0 користио једну TCP везу да преузме све документе
 - Једноставна имплементација
 - Веома лош РЛТ...



Начини за смањење РЛТ

1. Смањивање послатог садржаја
 - Мање слике, gzip компресија
2. Прилагођавање HTTP да боље користи проток
3. Избегавање поновних истих HTTP захтева
 - Кеширање и Веб проксији
4. Померање садржаја да буде ближи кориснику
 - CDN

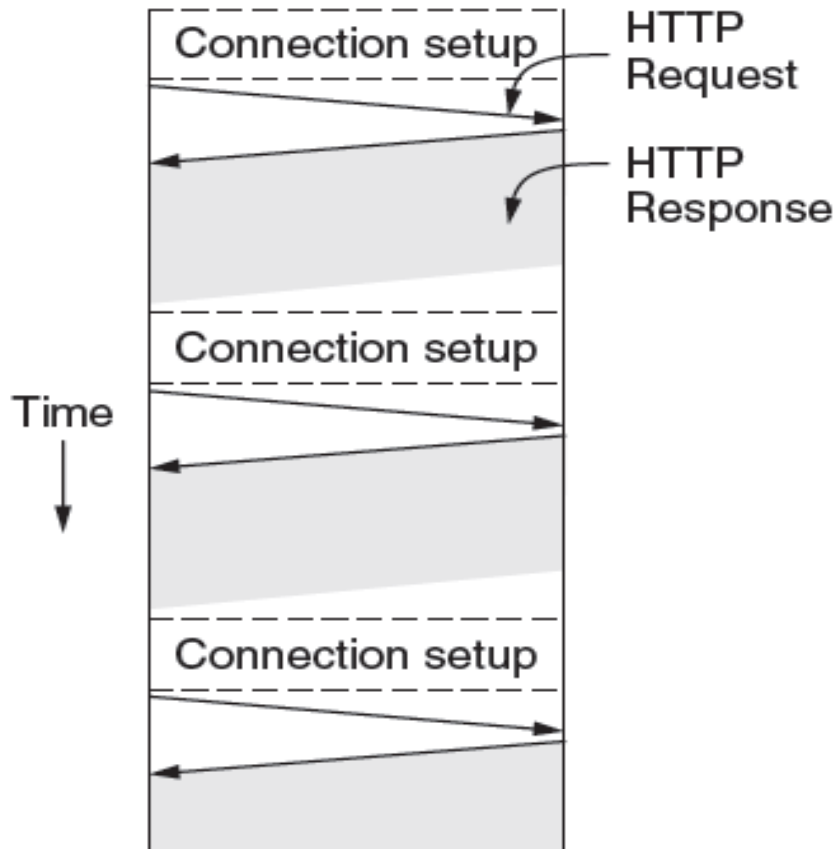
Паралелне ТСП везе

- Једноставан начин за побољшање РЛТ
 - Прегледач нпр. истовремено ради са 8 НТТР контекста
 - Сваки контекст прави одвојену ТСП везу
 - Сервер не захтева промену, само клијент односно прегледач
 - Сада 8 НТТР веза може боље да искористи доступни проток
 - Све раде истовремено, па се не гомилају кашњења која настају због успостава ТСП веза

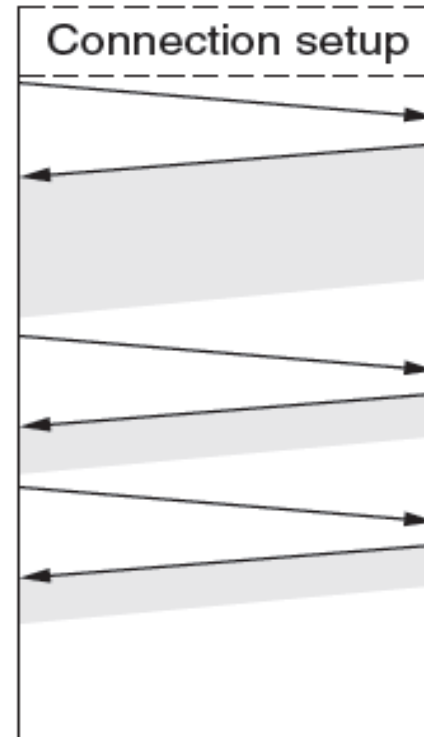
Трајне (вишеструко употребљиве) везе

- Уместо паралелних ТСП веза, може се користити и трајна (вишеструко употребљива) веза
 - Направи се једна ТСП веза за све ресурсе ка једном серверу
 - Она се после дели међу независним НТТР захтевима ка том серверу

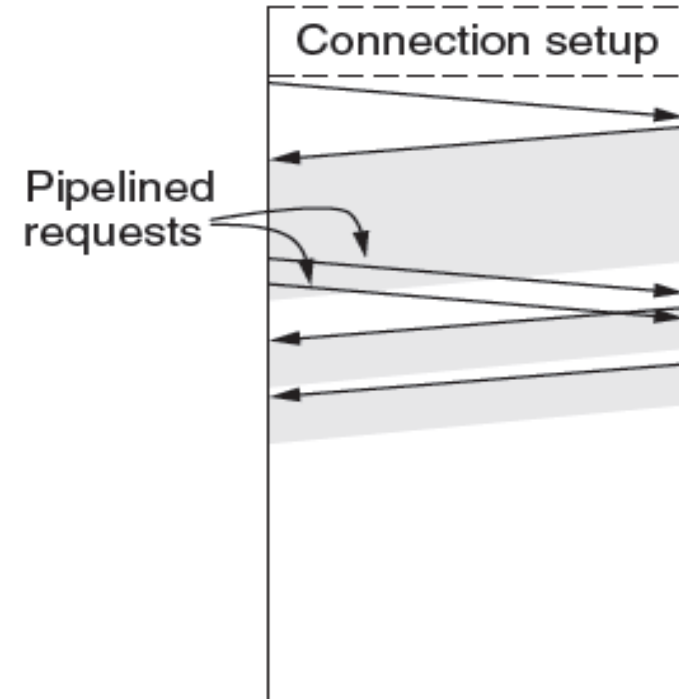
Трајне везе (3)



Почетни сценарио:
један захтев по TCP вези



Сценарио 2:
Једна успостава везе
па више HTTP захтев.



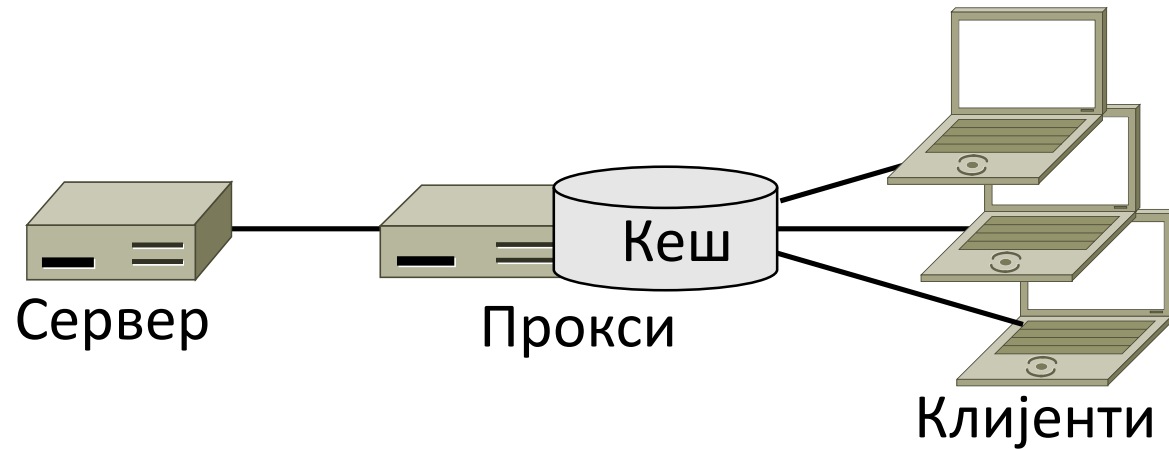
Сценарио 3:
Једна успостава везе,
па проточно слање захтева,
без чекања одговора

Апликативни слој

HTTP кеширање и HTTP проксији

Тема

- HTTP кеширање и HTTP проксији
 - Омогућавање вишеструке употребе истог садржаја



HTTP кеширање

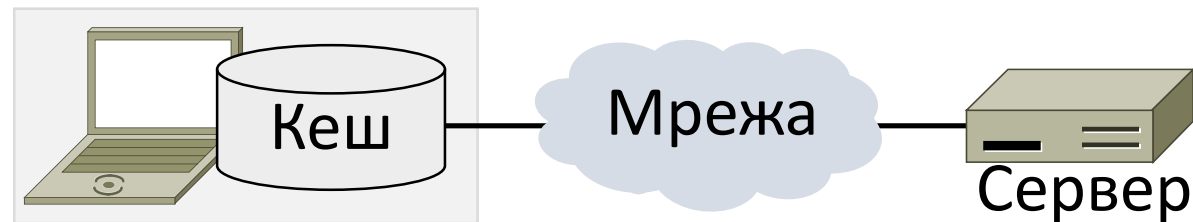
- Корисници често посећују исту страницу више пута
 - Ово омогућава чување локалне копије
 - Кеширање подразумева чување и употребу локалне копије



- Питање:
 - Када је исправно користити локалну копију?

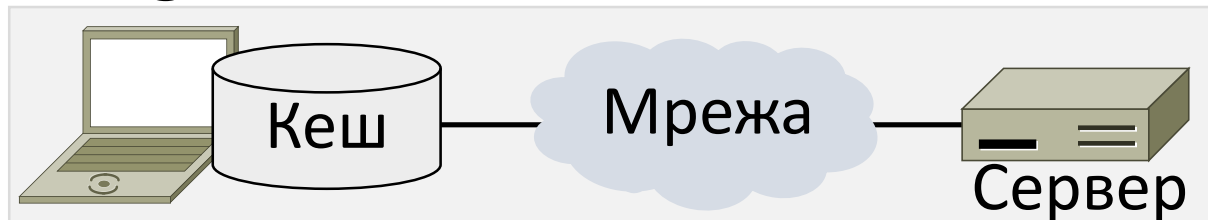
HTTP кеширања (2)

- Копија се сматра за исправну ако није истекла
 - Сервер доставља време истицања садржаја у виду додатног поља (обично се зове „expires“)
- Ако је копија исправна, користи се у тој форми, па није потребно преузимање са сервера

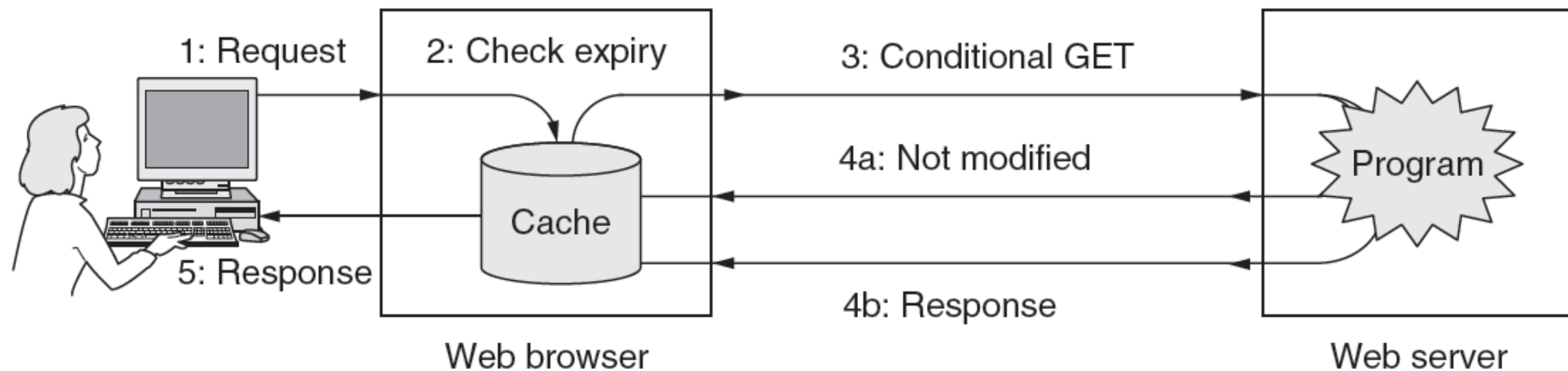


HTTP кеширање (3)

- Ако копија није локално исправна, мора се извршити провера исправности копије на серверу
 - Клијент шаље серверу условни GET захтев (у којем су и информације о локалној копији)
 - На основу њих сервер сазнаје да ли је локална копија застарела и у складу са тим шаље/или не шаље нови садржај
 - Могућа је и провера заснована на садржају употребом „Etag“ заглавља – неки вид хеш кода



HTTP кеширање (4)

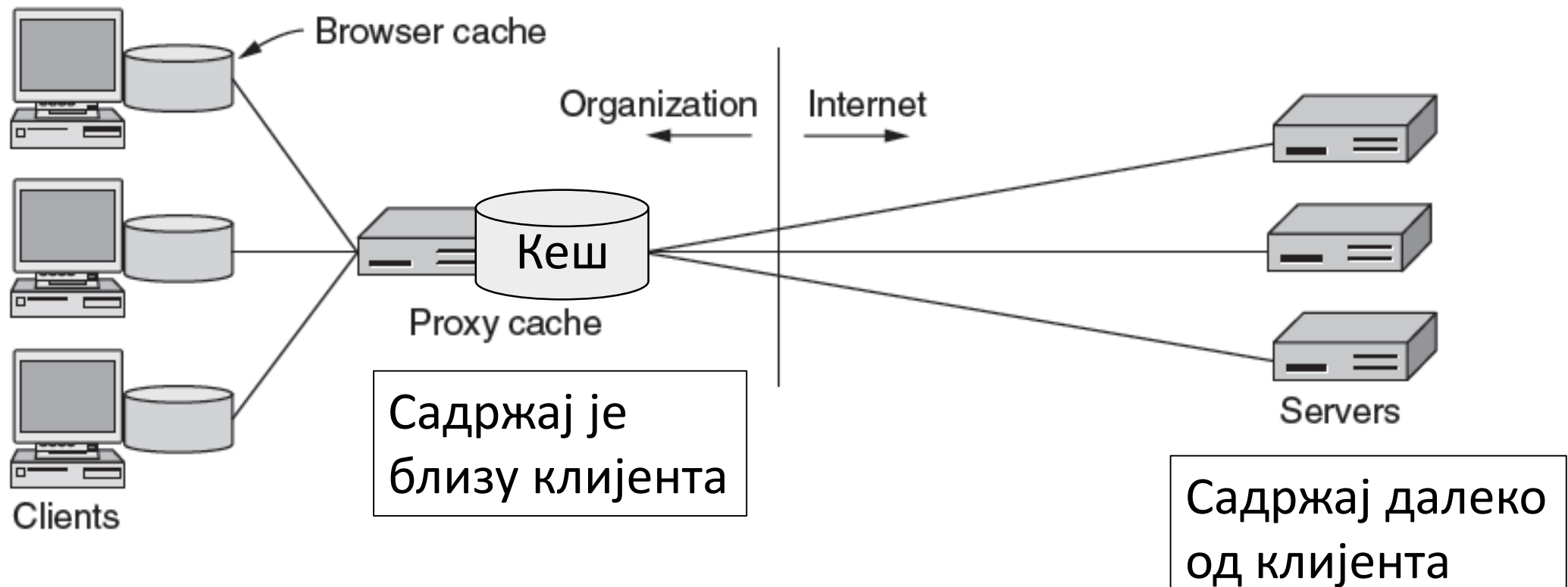


HTTP проксији

- Посредник између групе клијената и веб сервера
 - Различити клијенти такође могу да користе исти садржај тако да има смисла кеширати садржај и за групу клијената, а не само појединца
 - Додатно, у прокси се могу уградити и сигурносни механизми, забрана садржаја (цензура) и слично...
 - Корисно за компаније, организације, ISP, ...

HTTP проксији (2)

- Клиент контактира прокси; прокси потом потенцијално контактира сервер



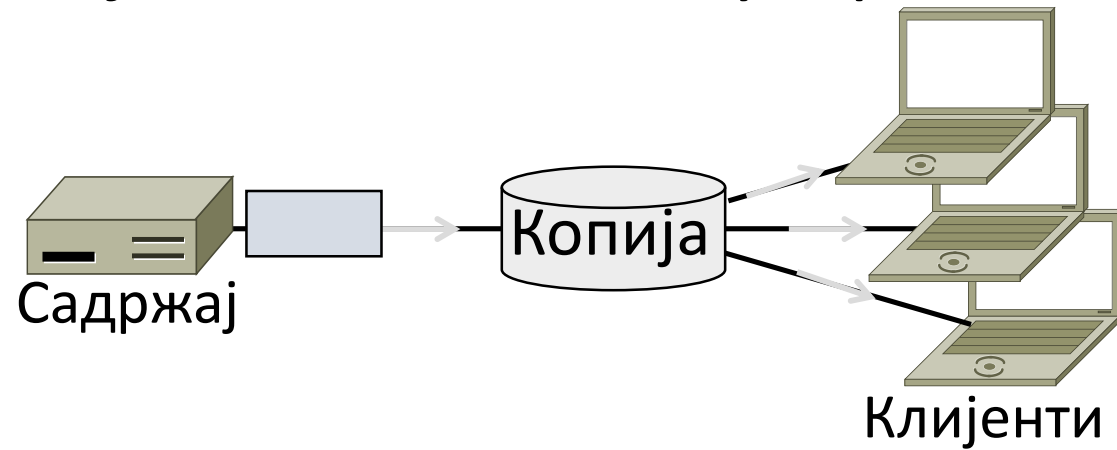
Апликативни слој

CDN – мреже за испоруку садржаја
(енг. Content Delivery Networks)

Тема

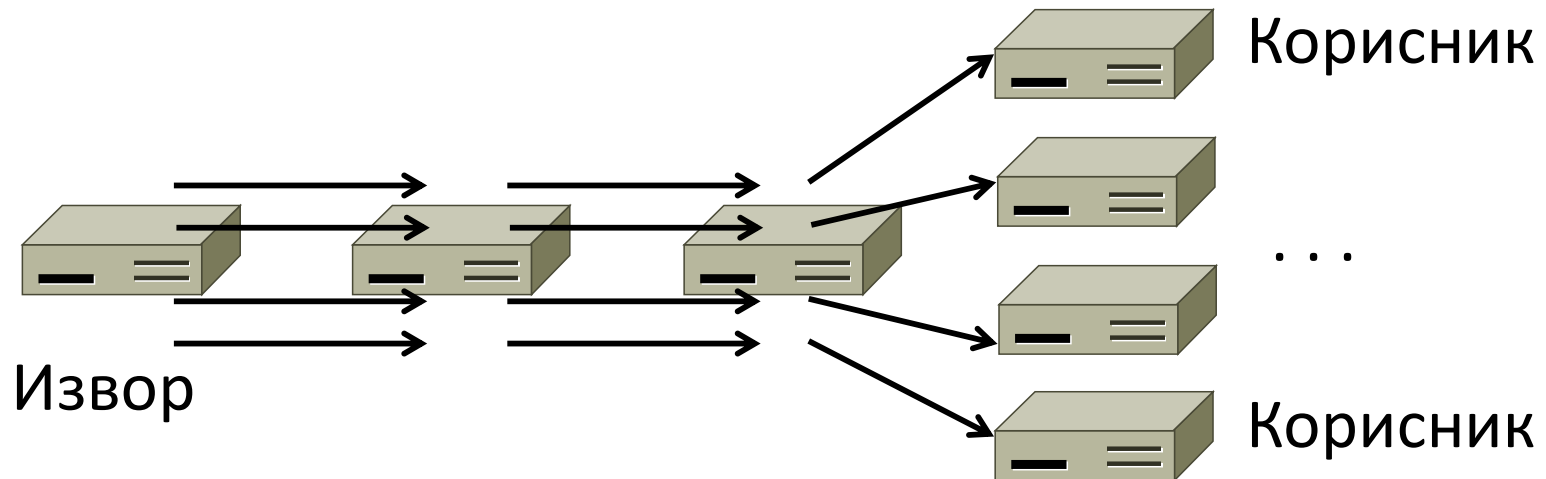
- CDN

- Ефикасна испорука често коришћеног садржаја; смањује чекање; смањење потрошње укупног протока
- Идеја: поставити популаран садржај ближе клијентима



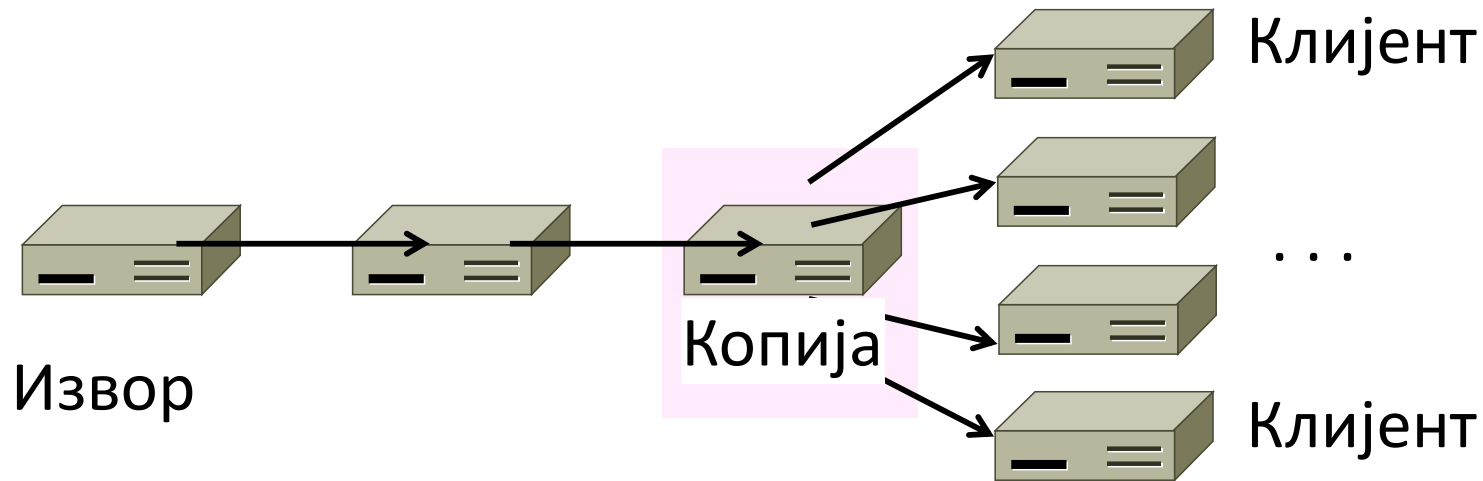
Сценарио без CDN-а

- Слање садржаја од извора ка 4 клијента
- Укупно $4 \times 3 = 12$ “хопова”



Сценарио са CDN-ом

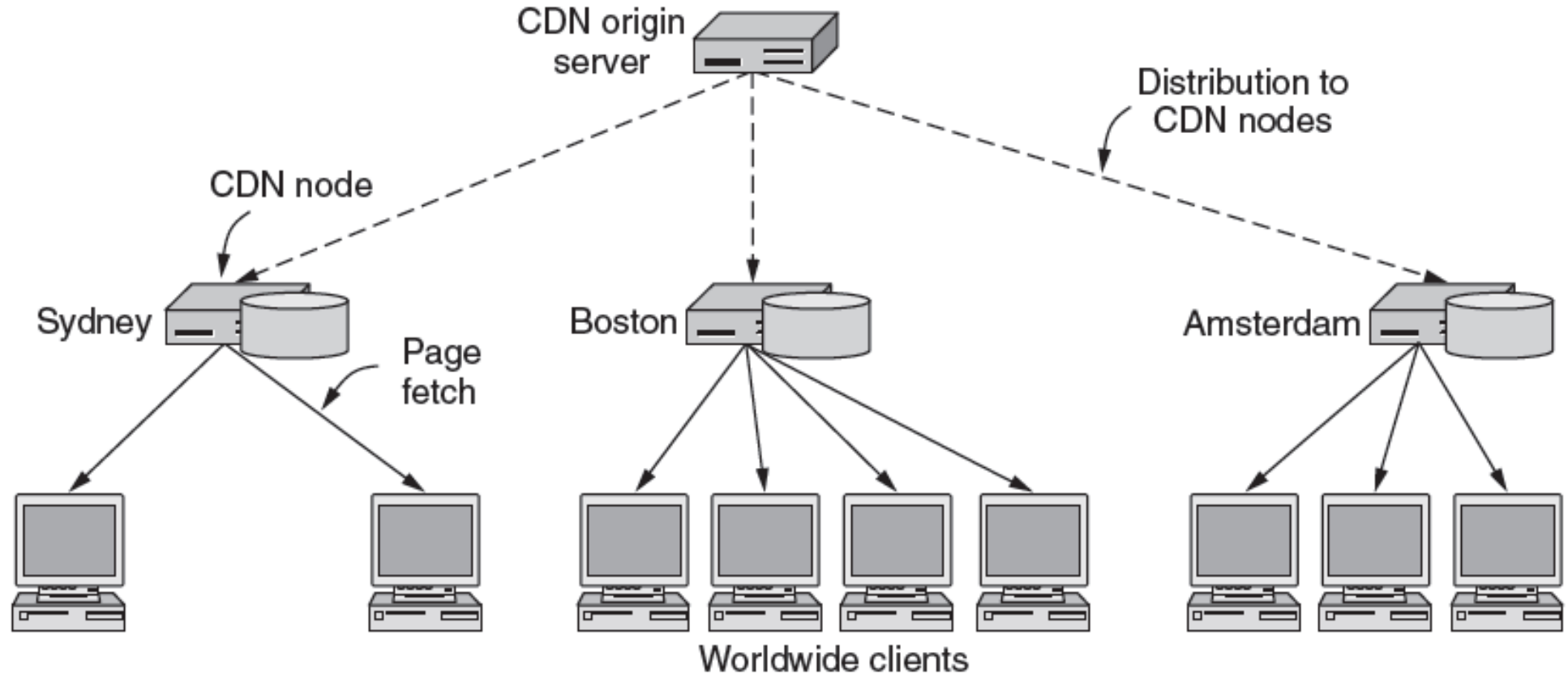
- Слање захтева мањи број $4 + 2 = 6$ “хопова”



Постављање садржаја близу клијената?

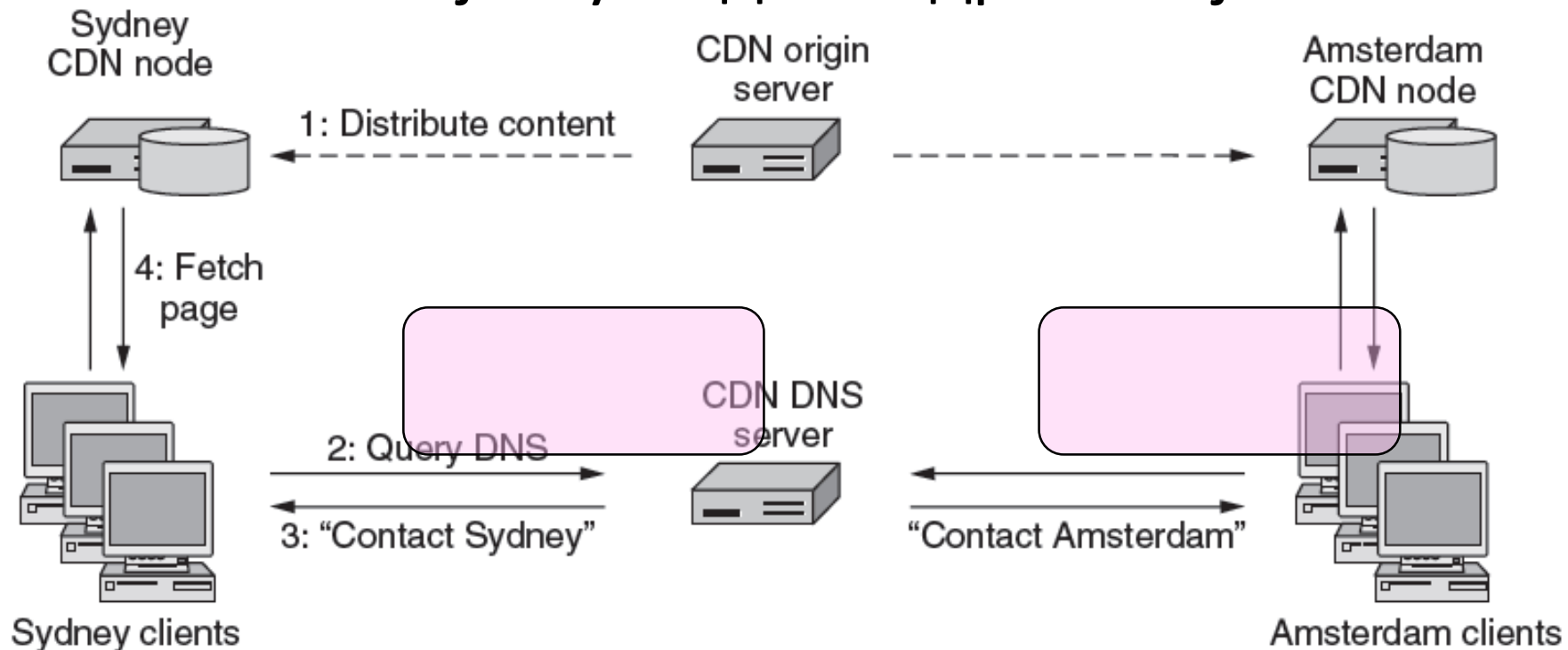
- Први ниво је употреба локално кеширања (прегледач) и кеширања на проксијима
 - Помаже, али је ограничено на садржаје које учестало користи појединац или група појединаца
- Шта ако је садржај популаран на вишем нивоу, нпр. слике са социјалних мрежа, видео спотови итд.
 - Онда копије желимо да поставимо географски циљано
 - Ово се може свести на употребу DNS

CDN



CDN (2)

- DNS одређивање адресе може да додели различите одговоре различитим клијентима
 - Сваком клијенту се да IP адреса најближе копије

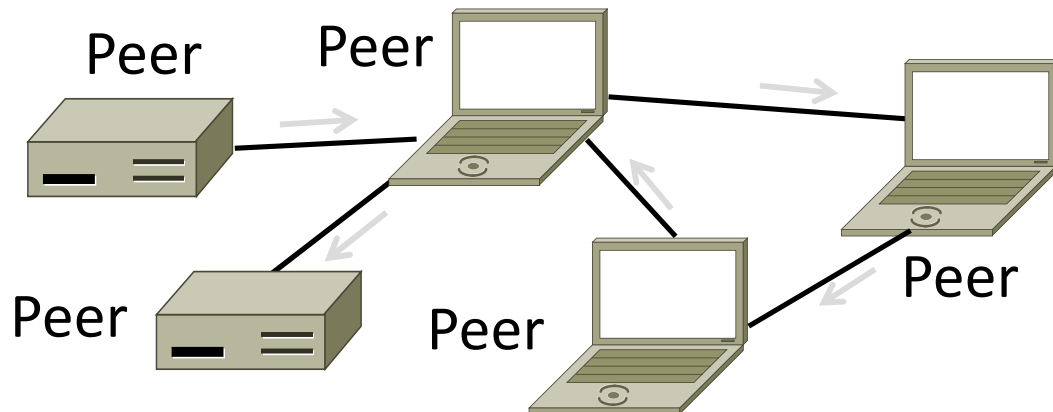


Апликативни слој

P2P модел испоруке садржаја, BitTorrent протокол
(енг. Peer-to-Peer - P2P)

Тема

- P2P модел испоруке садржаја – алтернатива за CDN
 - Нема централних, битнијих чворова
 - Свако је и клијент и сервер (испоручује и добавља садржај)
 - Сви чворови су равноправни
 - BitTorrent је пример реализације оваквог типа система

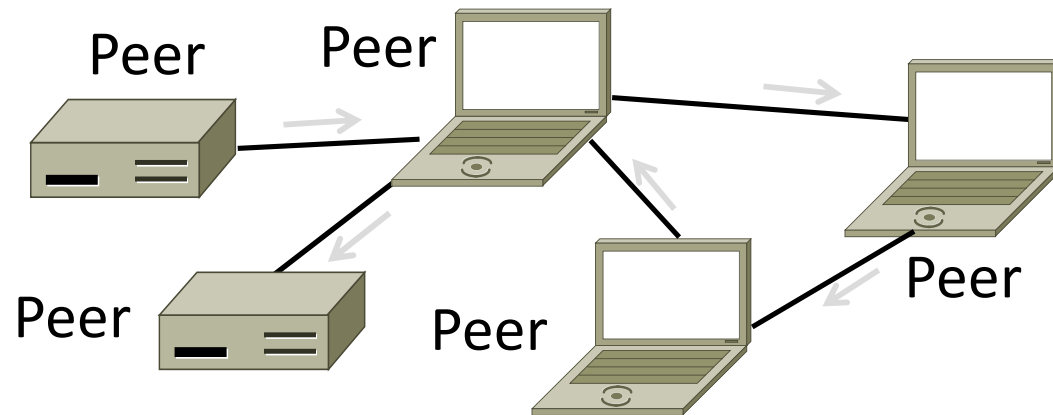


P2P

- Циљ је испорука садржаја без централизованих механизма
 - Ефикасно се понаша и са повећањем броја чворова
 - Може се постићи и висока поузданост
- Кључна идеја је да чворови помажу једни другима
 - Иницијално систем Napster 1999. године за дељење музичких садржаја (угашен због тога што је имао централизован списак садржаја)
 - Од 2001. BitTorrent – нема дељени регистар садржаја, па се не може оптужити појединац

P2P изазови

- Нема сервера на који се може ослонити
 - Комуникација између чворова мора да буде самоодржива и дистрибуирана
 - Постоје изазови при повећању броја чворова ...



P2P изазови (2)

1. Подстицај за учешће чвора

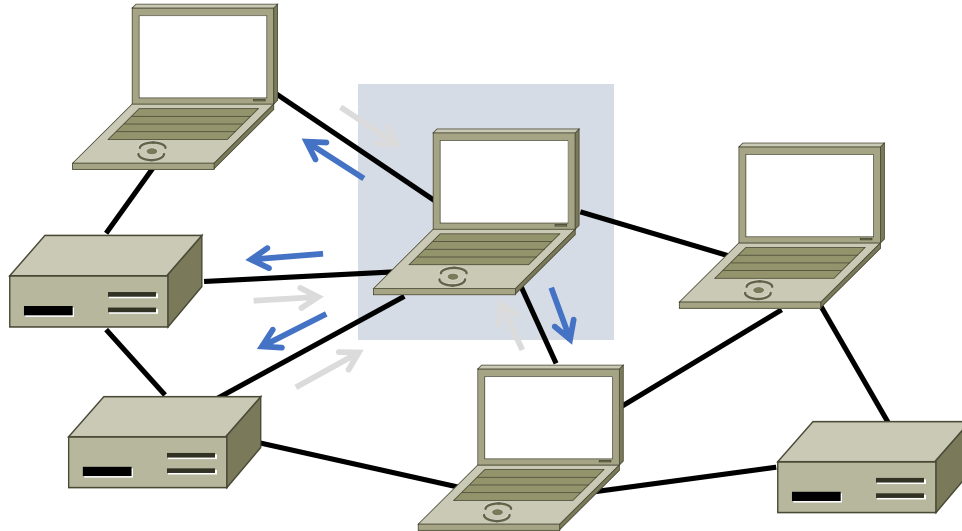
- Зашто би чворови помагали једни другима?

2. Децентрализација

- Како се проналази садржај без централног регистра (индекса)?

Подстицај за учешће чвора

- Чвор испоручује и добавља садржај:
 - Преузима (→) да би помогао себи,
 - Шаље (←) да би помогао другима
 - „Ја ћу послати теби ако ти пошаљеш мени...“



Омогућавање децентрализације

- Чвор мора да научи где се налази садржај
 - DHT дистрибуиране хеш табеле (енг. Distributed Hash Tables)
- DHT користе алгоритме за рад са дистрибуираним индексом
 - Индекс је дистрибуиран преко свих чворова
 - Индекс за тражени садржај даје списак свих чворова који га садрже
 - Сваки чвор има приступ индексу

BitTorrent

- Стандардни P2P систем који се данас користи
 - Веома брз раст, могућност преноса великих датотека
 - Чини велики део Интернет саобраћаја данас
 - Користи се за легални и илегални садржај
- Достављање података посредством „торената“:
 - .torrent датотека садржи мета податке, попут списка датотека, величине...
 - Садржај се шаље у деловима због паралелизма
 - Не узимају се сви делови редом, зашто?
 - Списак чворова не мора бити у DHT
 - Може бити и наведен на тзв. Tracker серверима – рачунари чија је намена да памте спискове чворова који поседују одређену датотеку (данас се све мање користе због појаве DHT)

BitTorrent протокол

- Кораци у раду протокола:

1. Започиње са са torrent датотеком

- У њој су или локација Tracker сервера
- Или информације потребне да би се контактирао DHT

2. Размена података са различитим чворовима

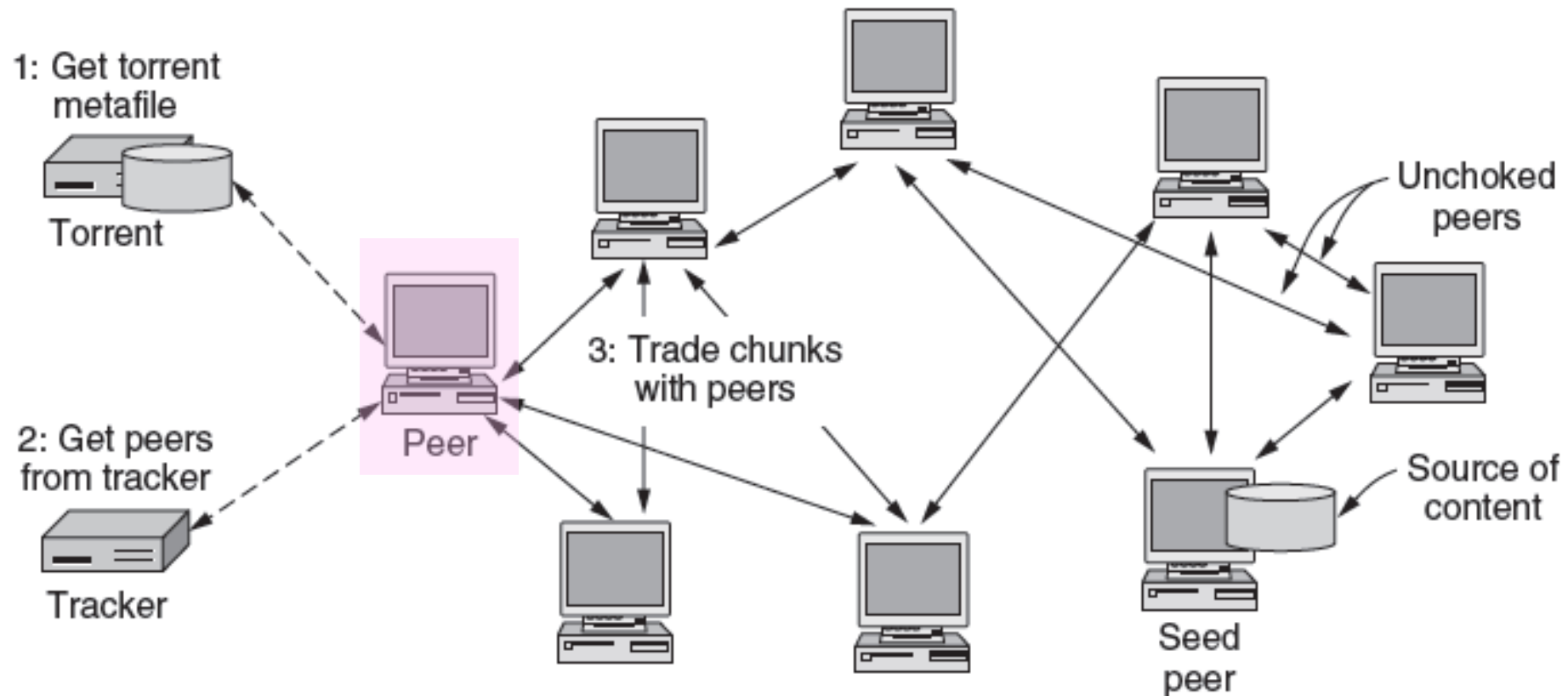
- Имам делове садржаја који треба другима,
други имају делове садржаја који треба мени → трговина

3. Што више делова шаљем, више и добијам?

- Одлазни саобраћај према неком чвору ће пратити
долазни саобраћај са њега

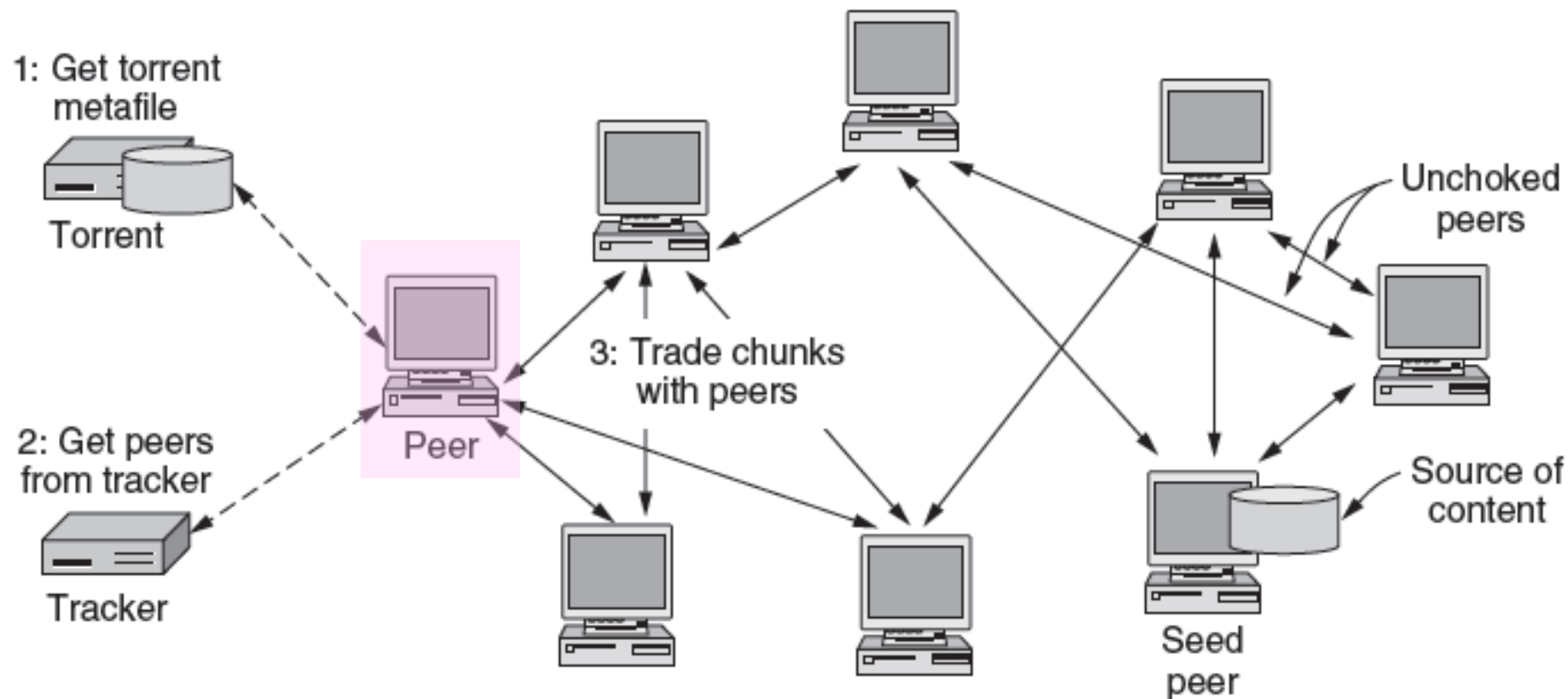
BitTorrent протокол (2)

- Чворови који имају садржај или желе да га преузму се повежу и започињу размене...



BitTorrent протокол (3)

- Делови који се шаљу су обично насумично изабрани
 - Ако би се слали редом, могућност трговине би се значајно смањила



BitTorrent протокол (4)

- Обуставља се саобраћај према чворовима који не сарађују

