

# Основе веб програмирања

Борисав Живановић (borisavz)

17. децембар 2022.

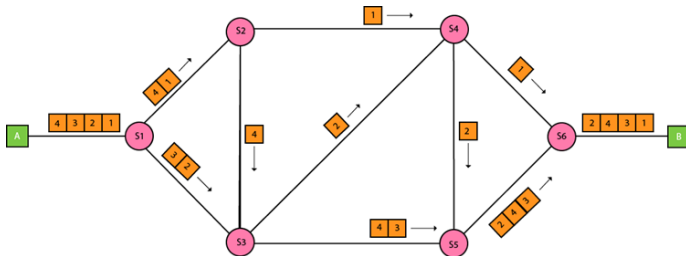
# Садржај

- 1 Основни појмови мрежног програмирања
- 2 Клијент-сервер архитектура
- 3 Еволуција веб апликација
- 4 HTTP протокол
- 5 Рад са базом података
- 6 Архитектура веб апликације
- 7 Аутентификација и ауторизација

# Packet switching I

- Потребно је да поруку пошаљемо примаоцу
- Директна веза са сваким примаоцем није остварива
- Идеја: повезивање пошиљаоца/примаоца у мрежу, дељење комуникационог канала
- Решење: **комутација пакета (packet switching)**
  - Поруку изделимо на пакете
  - Пакетима додамо заглавље (header) са адресом пошиљаоца и примаоца
  - Систем зна путање до примаоца
  - Поруку шаљемо пакет по пакет
  - Само један пакет заузима комуникациони канал
  - Пакети могу да путују различитим путањама кроз мрежу, да дођу у различитом редоследу до примаоца, или да нестану

# Packet switching II

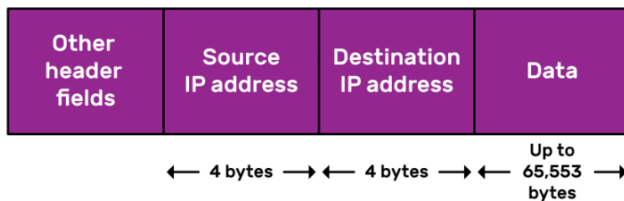


Слика: комутација пакета (packet switching)

# Internet Protocol I

- Како би комуницирали у мрежи, потребно је да сваки учесник у комуникацији има додељену **јединствену** адресу
- Поруци придружујемо **заглавље (header)** које садржи:
  - Адресу пошиљаоца (source address)
  - Адресу примаоца (destination address)
  - Додатна поља (верзија IP протокола, flags, TTL, checksum, ...)
- Захваљујући овом заглављу систем зна коме да проследи поруку
- У одговори су адресе пошиљаоца и примаоца **замене**не!

# Internet Protocol II

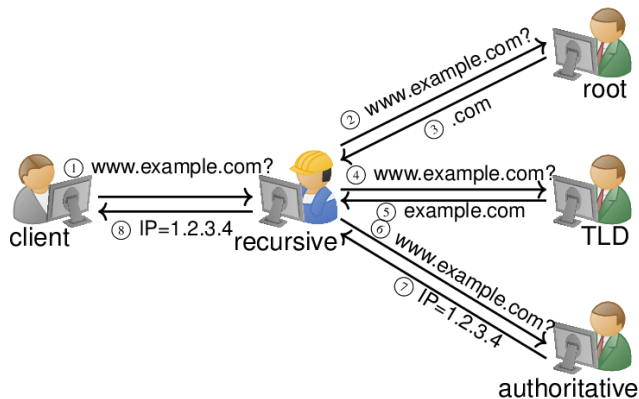


Слика: упрошћена структура IP пакета

# DNS I

- Проблем: све више сервера на мрежи
- Није практично памтити сваку адресу у бројчаном облику
- Идеја: систем за придруживање имена, сличан телефонском именику
- Решење: **DNS (Domain Name System)**
  - IP адреси додељујемо симболичко име (домен)
  - Домени су хијерархијски (структура стабла)
  - DNS је одговоран за одређени део хијерархије
  - Као одговор враћа IP адресу или адресу одговорног DNS сервера
  - Морамо знати IP адресу DNS сервера!

## DNS II



Слика: DNS упит



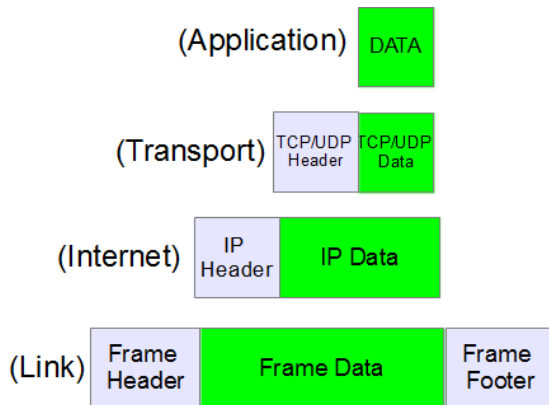
# Transmission Control Protocol I

- Решили смо проблем адресирања уређаја на мрежи...
- ...али нисмо проблеме редоследа пристиглих пакета и нестајања пакета
- Додатни проблем: шта ако имамо више мрежних апликација на истом рачунару, како да проследимо поруку одговарајућој апликацији?

# Transmission Control Protocol II

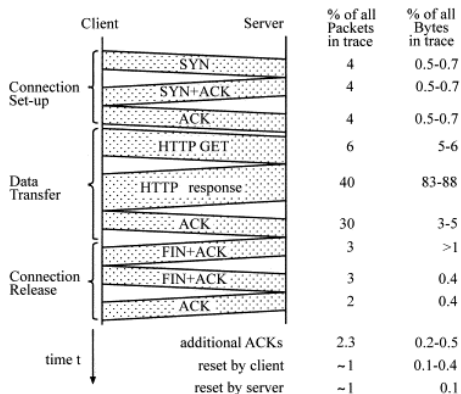
- Решење: **TCP (Transmission Control Protocol)**
  - Додајемо додатно заглавље на нашу поруку
  - Заглавље садржи source и destination port (слично адреси пошilhaоца и примаоца, али се односи на апликацију), sequence number (редослед поруке)
  - Уколико пакет нестане, шаље се поново
  - Оперативни систем осигурава да само једна апликација користи одређени порт

# Transmission Control Protocol III



Слика: енкапсулација пакета

# Transmission Control Protocol IV

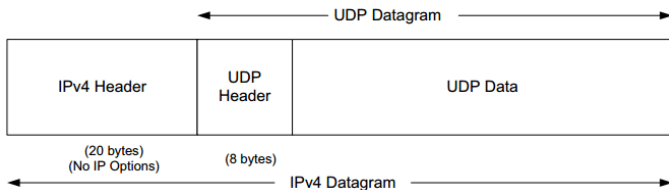


Слика: Ток TCP комуникације

# User Datagram Protocol I

- Успостављање конекције траје одређено време
- За поруке које стају у један пакет, можемо користити једноставнији **UDP (User Datagram Protocol)**
- Задржавамо адресирање апликација, али губимо гаранцију испоруке
- DNS користи UDP

# User Datagram Protocol II



Слика: енкапсулација пакета

# User Datagram Protocol III



Слика: Садржај заглавља