

# Оперативни системи и Рачунарске мреже

Александар Картељ

[aleksandar.kartelj@gmail.com](mailto:aleksandar.kartelj@gmail.com)

Рачунарска гимназија

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.

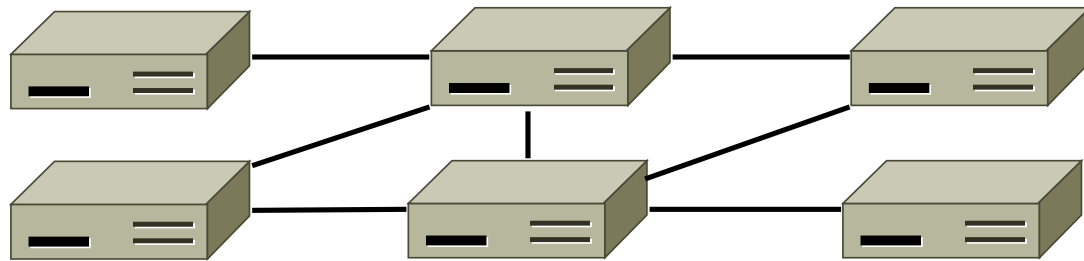
Electronically reproduced by permission of Pearson Education, Inc., Upper Saddle River, New Jersey

# Мрежни слој

Преглед

# Шта ће нам мрежни слој?

- Већ можемо да изградимо мрежу употребом веза и скретница...

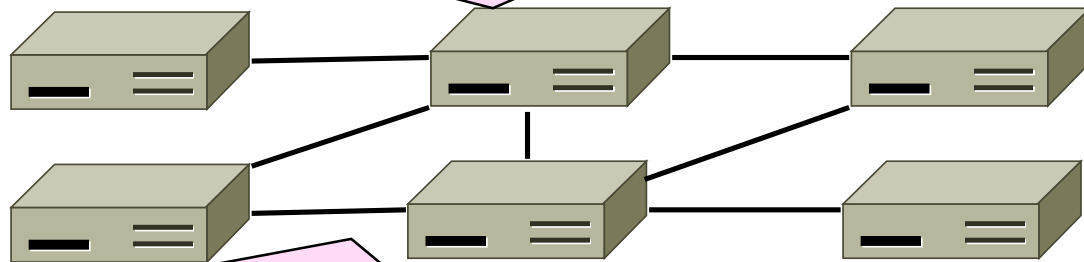


# Проблеми са скретницама

## 1. Не скалирају се добро

- Табела релација би постала огромна
- Иницијално емитовање целом свету

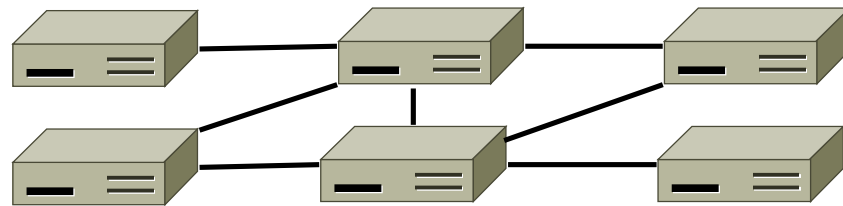
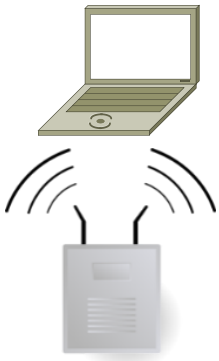
Табела релација за цео свет!



Емитовање нових чворова ка целом свету!

# Проблеми са скретницама (2)

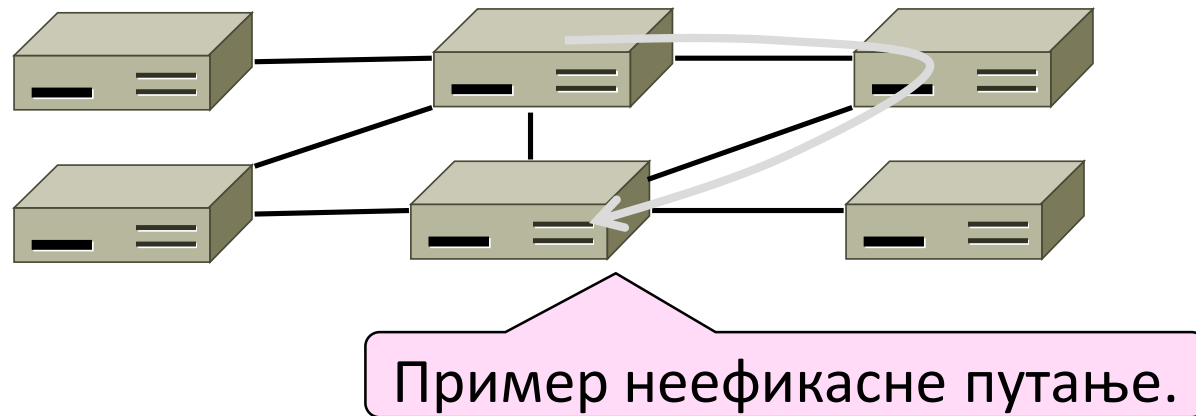
2. Не раде ако су технологије слоја везе различите
- Чворови на Етернету + 3G + 802.11 ...



## Проблеми са скретницама (3)

### 3. Не омогућавају контролу саобраћаја

- Хоћемо да планирамо руте и проток

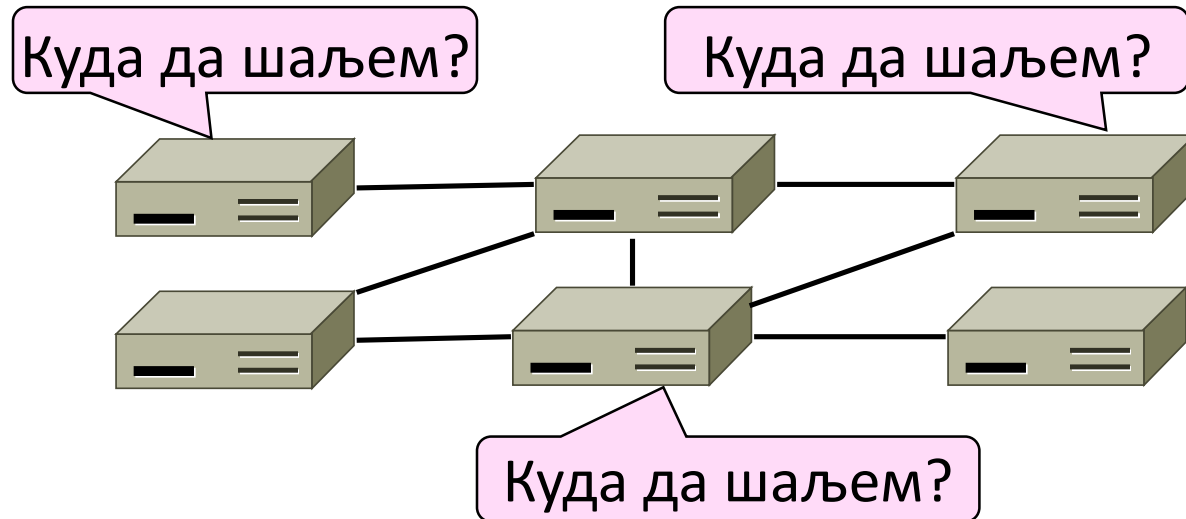


# Теме

- Типови мрежних сервиса
  - Датаграмски модел (пакети), Модел виртуелног кола
- IP (Интернет протокол)
  - Прослеђивање (алгоритам најдужег одговарајућег префикса)
- Алгоритми рутирања

# Рутирање или прослеђивање

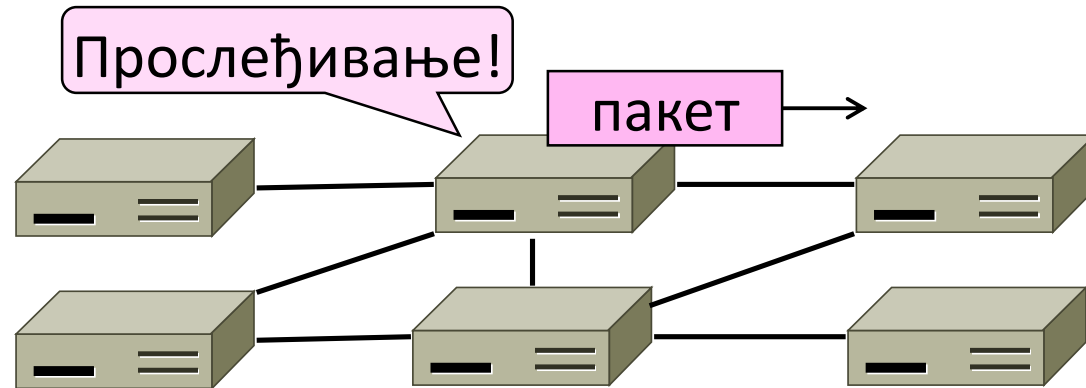
- Рутирање је процес одлучивања у ком правцу треба послати саобраћај
  - Овакво одлучивање је скупо!
  - Морају се развити паметни алгоритми...





## Рутирање или прослеђивање (2)

- Прослеђивање је процес слања пакета на основу локалне табеле
  - Чворови то раде локално и брзо



# План рада

- Прослеђивање ћемо прво обрадити
  - Како чворови шаљу пакете другим чворовима
  - Претпостављамо да је већ одлучено где шаљу
- Рутирање следећи пут
  - Одлучивање где се пакети шаљу
  - Логички иде пре прослеђивања, али засад посматрамо као „црну кутију“
  - Алгоритми над графовима

# Мрежни слој

Типови сервиса

# Два типа мрежних сервиса

- Датаграми,  
сервис без успостављања везе

- Попут поште
- (IP протокол ради ово)



- Виртуелна кола,  
сервис са успостављањем везе

- Попут фиксне телефоније
- Прескачемо

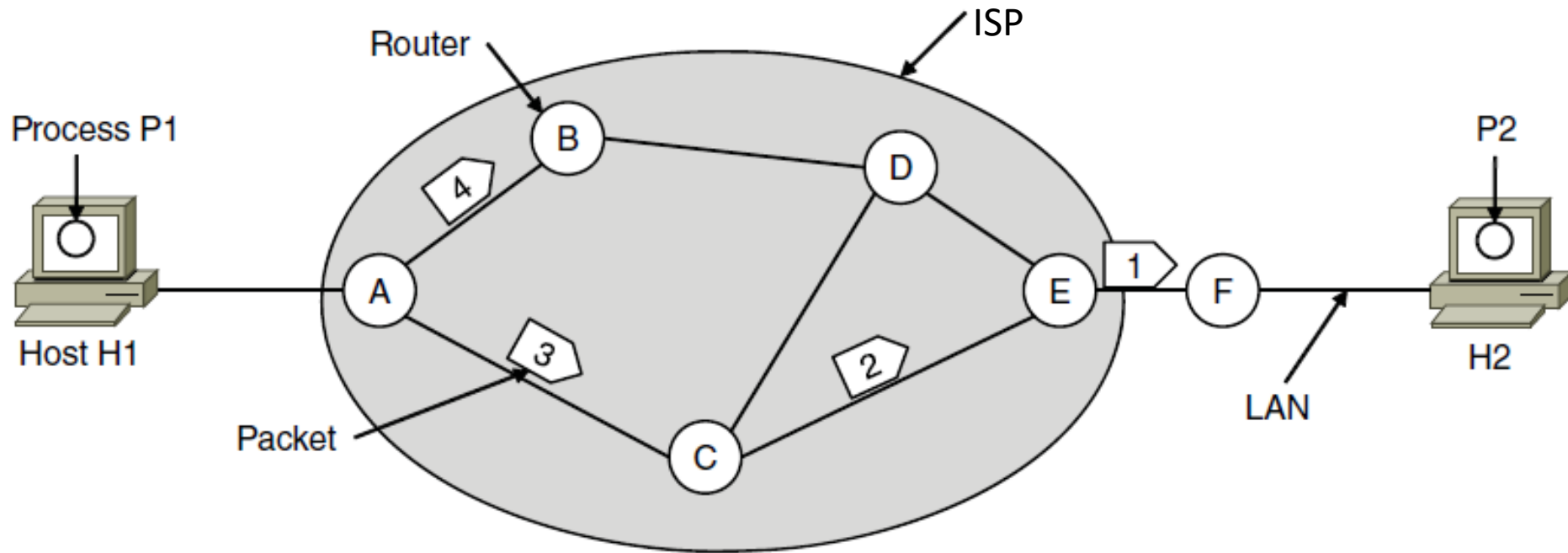


# Сачувај-и-проследи механизам

- Оба типа сервиса се реализију техником сачувај-и-проследи (store-and-forward)
  - Усмеривачи (рутери) добијају пакет, и привремено га чувају све док га не проследи даље (користе баферисање)
  - Користи технику статистичког мултиплексирања за управљање протоком

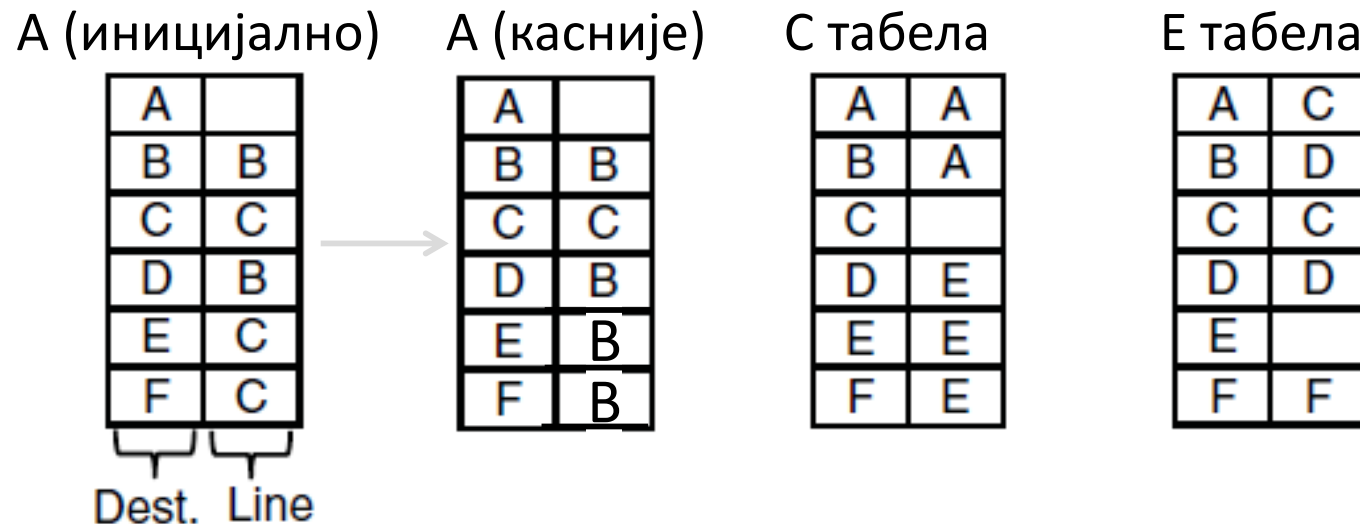
# Датаграмски сервис

- Пакет садржи циљну адресу на основу које усмеривач прослеђује пакет даље



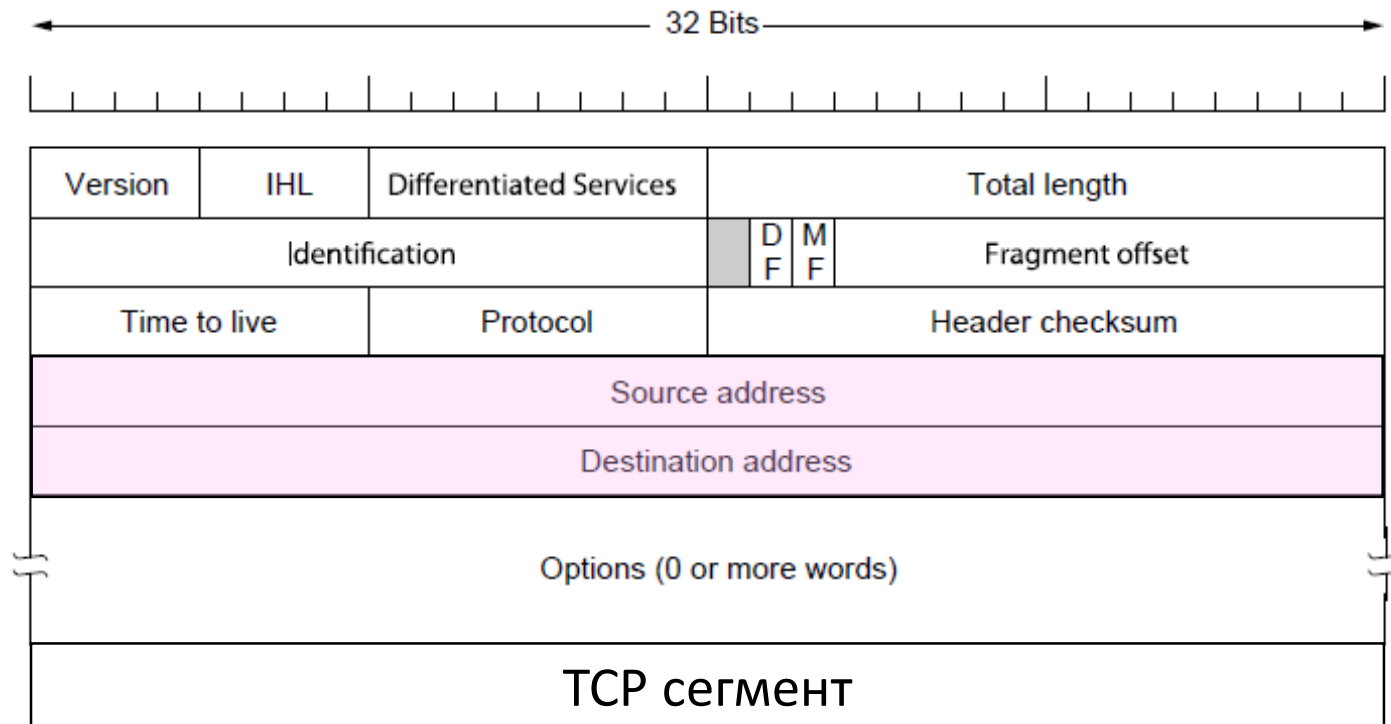
## Датаграмски сервис (2)

- Сваки усмеривач има табелу прослеђивања (forwarding table) – табела се мења!
  - Ред у табели за дату циљну адресу одређује суседни чвор којем ће се пакет проследити (следећих хоп)



# IP (Интернет протокол)

- Мрежни слој користи датаграмски сервис
  - IPv4 пакет има 32-битну адресу и обично је велик око 1.5 KB



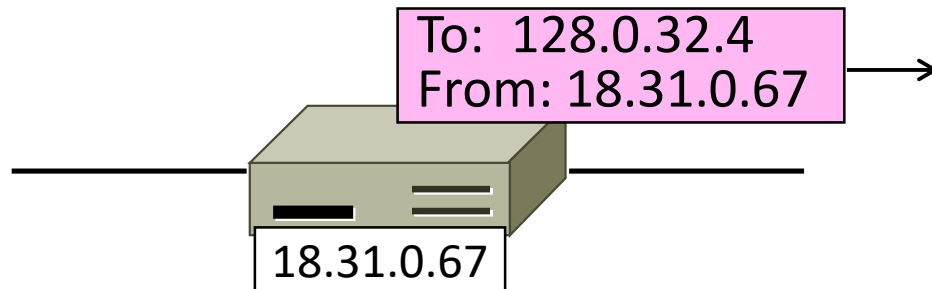


# Мрежни слој

IP префикси

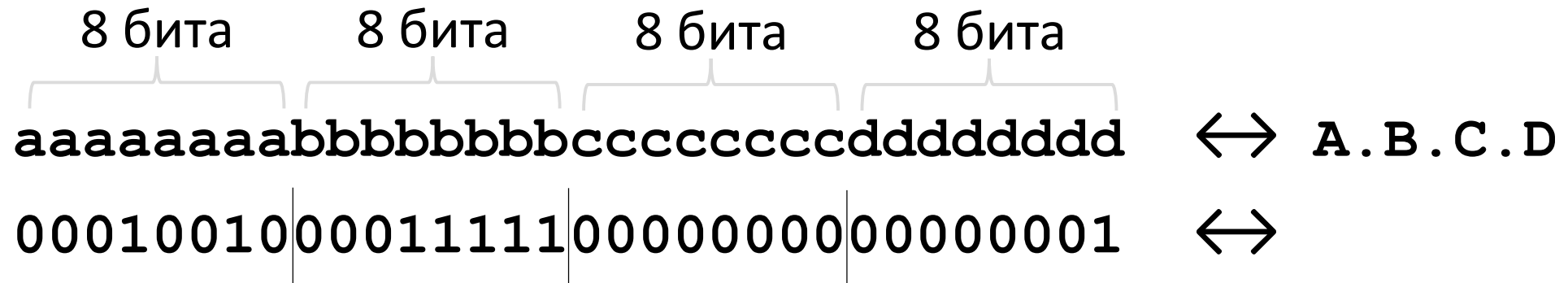
# Тема

- Како изгледају IP адресе?
  - Шта су IP префикси (или блокови адреса)
  - Све ово се односи на IPv4



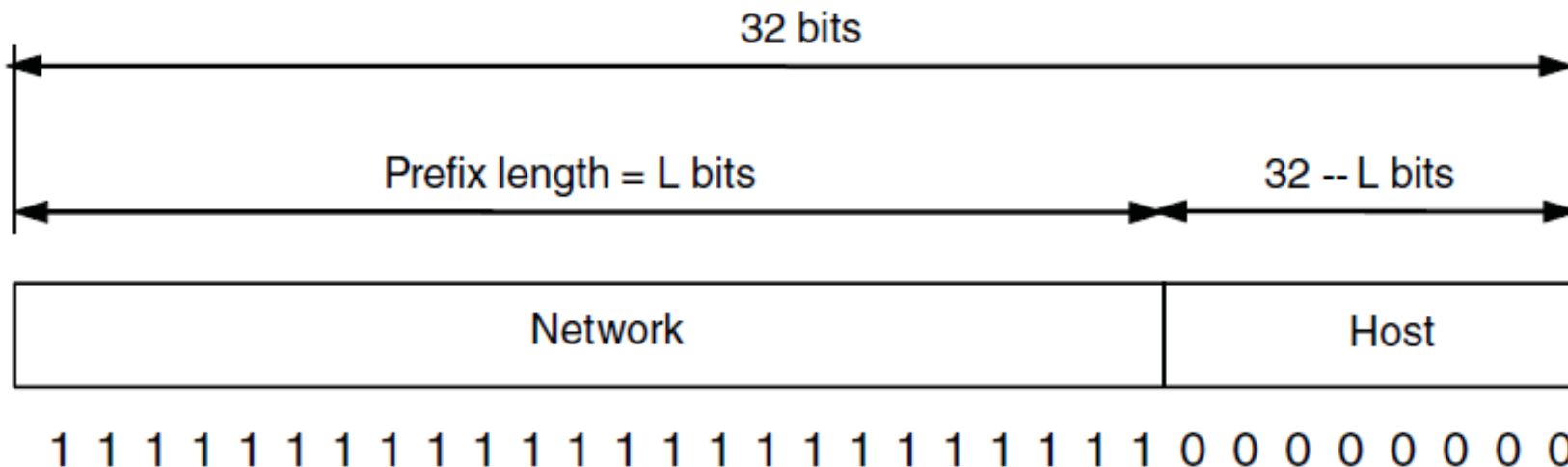
# IP адреса

- IPv4 користи 32-битну адресу
  - IPv6 користи 128-битну адресу
- Нотација у виду квартета декадних бројева
  - Четири 8-битна броја одвојена тачкама



# IP префикси

- Адресе се групишу у блокове који се називају префикси
  - L-битни префикс је група адреса које имају исти префикс дужине L бита
  - То значи да L-битни префикс има  $2^{32-L}$  различитих адреса
  - Префикси описују мреже рачунара односе опсеге адреса



## IP префикси (2)

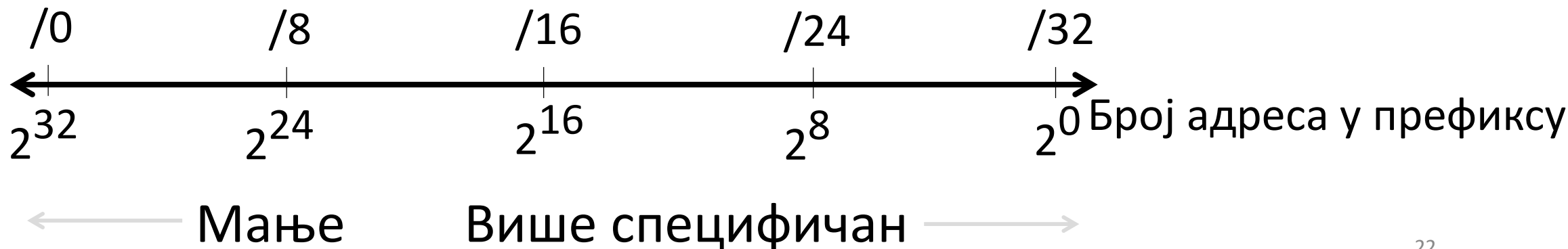
- Нотација облика “IP адреса/дужина префикса”
  - Нпр., 128.13.0.0/16 је опсег 128.13.0.0 до 128.13.255.255
  - Префикс облика /24 одговара опсегу са 256 адреса док /32 одговара јединственој адреси

00010010|00011111|00000000|**xxxxxxx** ↔

↔ 128.13.0.0/16

# IP префикси (3)

- IP адреса припада различитим префиксима
- Више специфичан префикс
  - Је дужи префикс, јер ближе одређује опсег око те адресе
- Мање специфичан префикс
  - Је краћи префикс. Који је најмање специфичан?

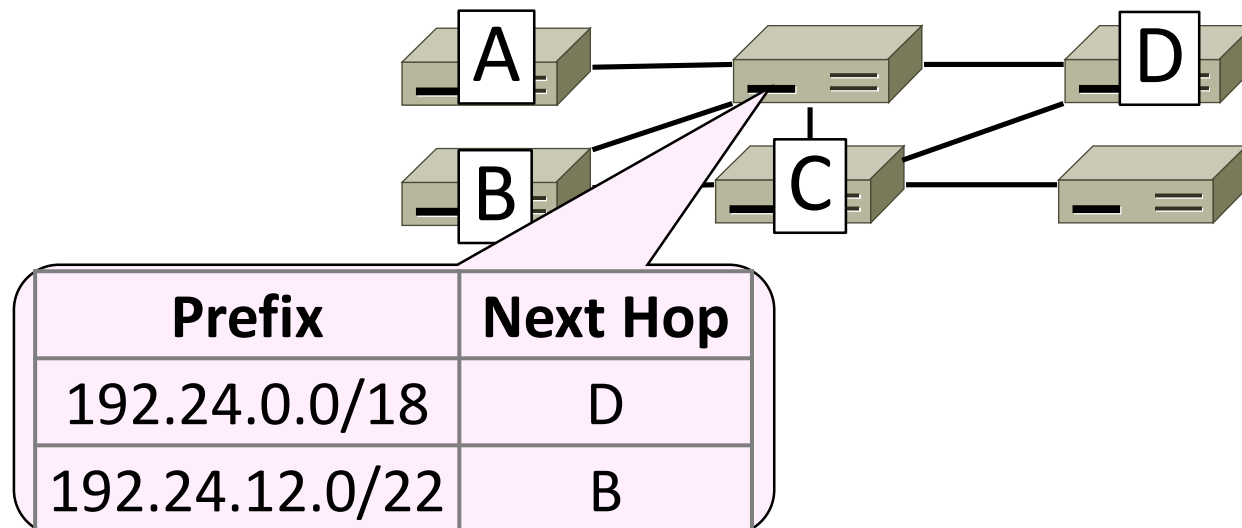


# Мрежни слој

Прослеђивање у IP протоколу

# IP прослеђивање

- Све IP адресе једне мреже припадају истом префиксу
- Сваки усмеривач поседује табелу уређених парова облика (префикс, следећи чвор – хоп)
- Зашто не (циљна адреса, следећи чвор)?

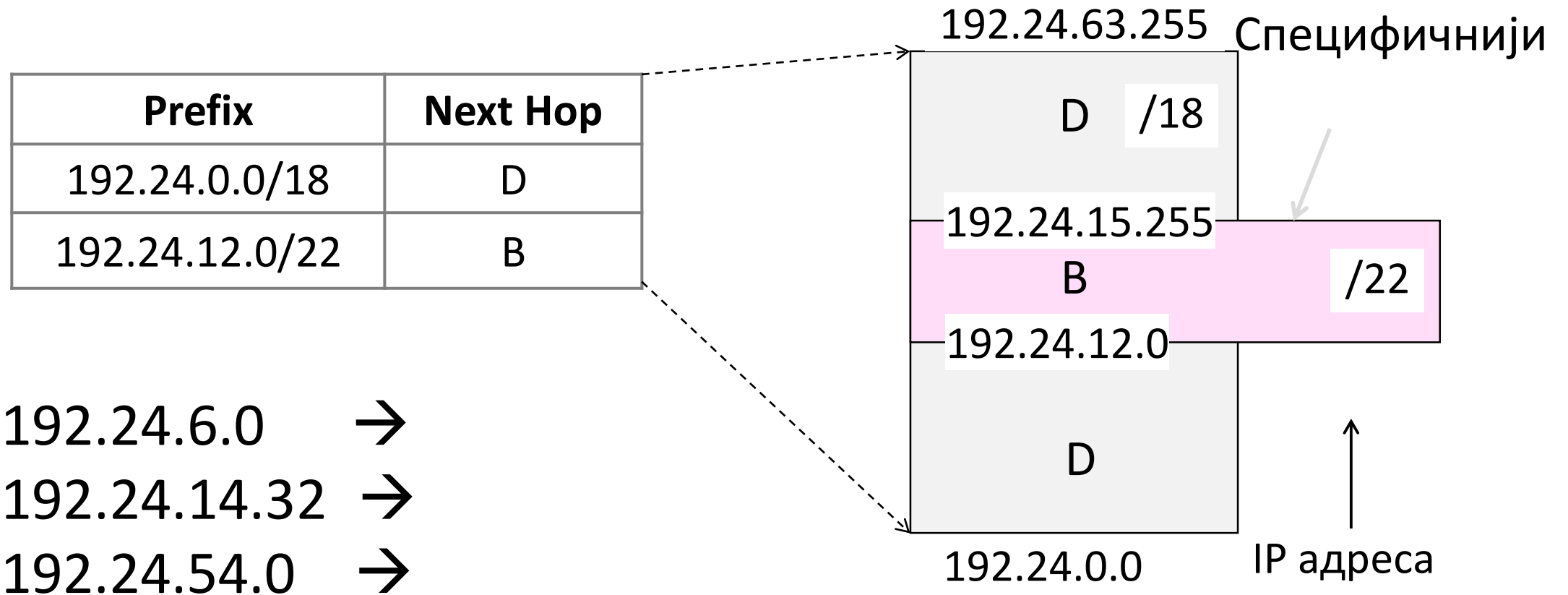




# Најдужи одговарајући префикс

- Префикси у табели се могу преклапати!
- Правило најдужег одговарајућег префикса:
  - За сваки пакет, пронаћи најдужи префикс који садржи циљну адресу, односно најспецифичнији префикс
  - Проследити пакет чвору који одговара том префиксу

# Најдужи одговарајући префикс (2)



# Најдужи одговарајући префикс (3)

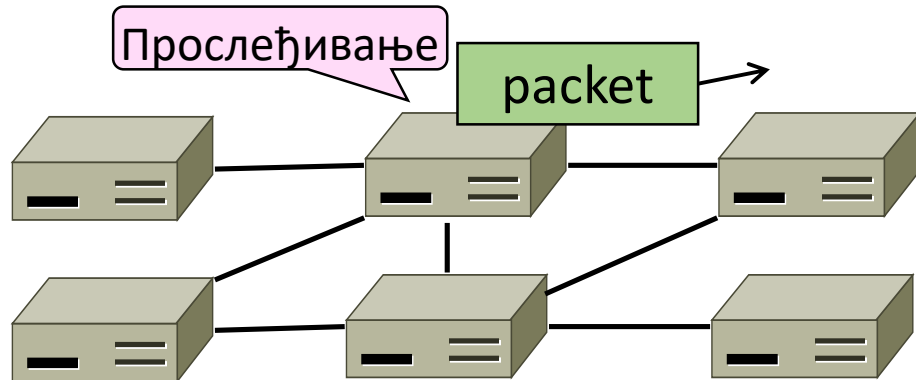
- Могу да пруже подразумевано понашање, са мање специфичним префиксом
- Могу да пруже специјално понашање, са специфичнијим префиксом
  - Због перформанси, сигурности итд.
- Огромна табела је распарчана хијерахијски на велики број усмеривача
  - Компактније табеле → подразумеваније (мање ефикасно, сигурно, ...) понашање
  - Веће табеле → специфичније (ефикасније, сигурније, ...) понашање
  - Ово је компромис између временске и просторне сложености који стално срећемо у рачунарству!

# Рутирање

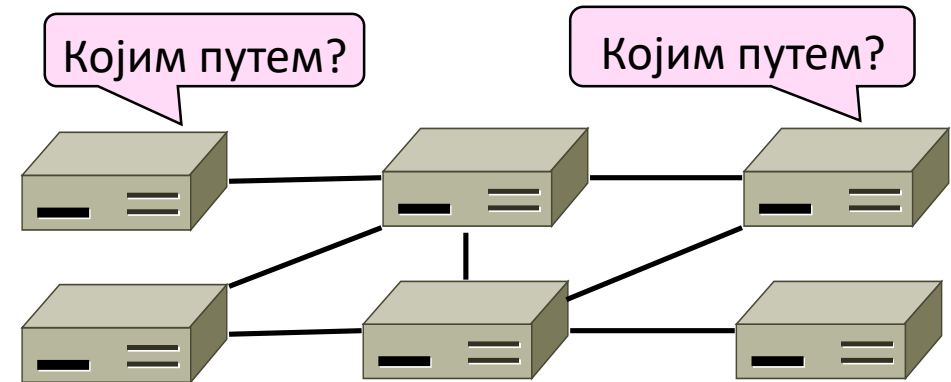
Преглед

# Рутирање и прослеђивање

- Прослеђивање је процес слања пакета суседним чворовима



- Рутирање је процес одређивања путања којима ће се прослеђивање вршити



# Циљеви рутирања

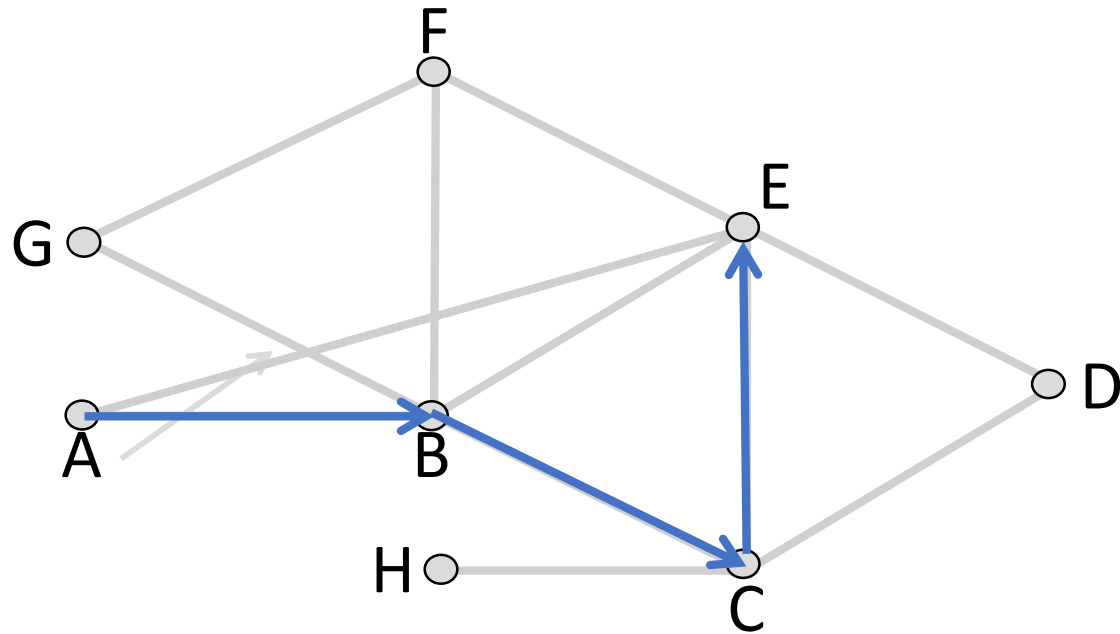
Својство	Значење
Тачност	Проналази путању која ради
Ефикасност	Рационално троши проток
Равноправност	Подједнака права чворова, не изгладњује неке чворове
Брза конвергенција	Брз опоравак након промена (отказа, нових чворова)
Скалабилност	Ради добро и када мрежа (број чворова, веза, ...) расте

# Рутирање

Рутирање са најкраћим путевима (најмањим трошком)

# Тема

- Који пут је најбољи?
  - Мора се прво дефинисати према чему најбољи: дужини, цени, кашњењу или комбинацији истих...





# Мере трошкова

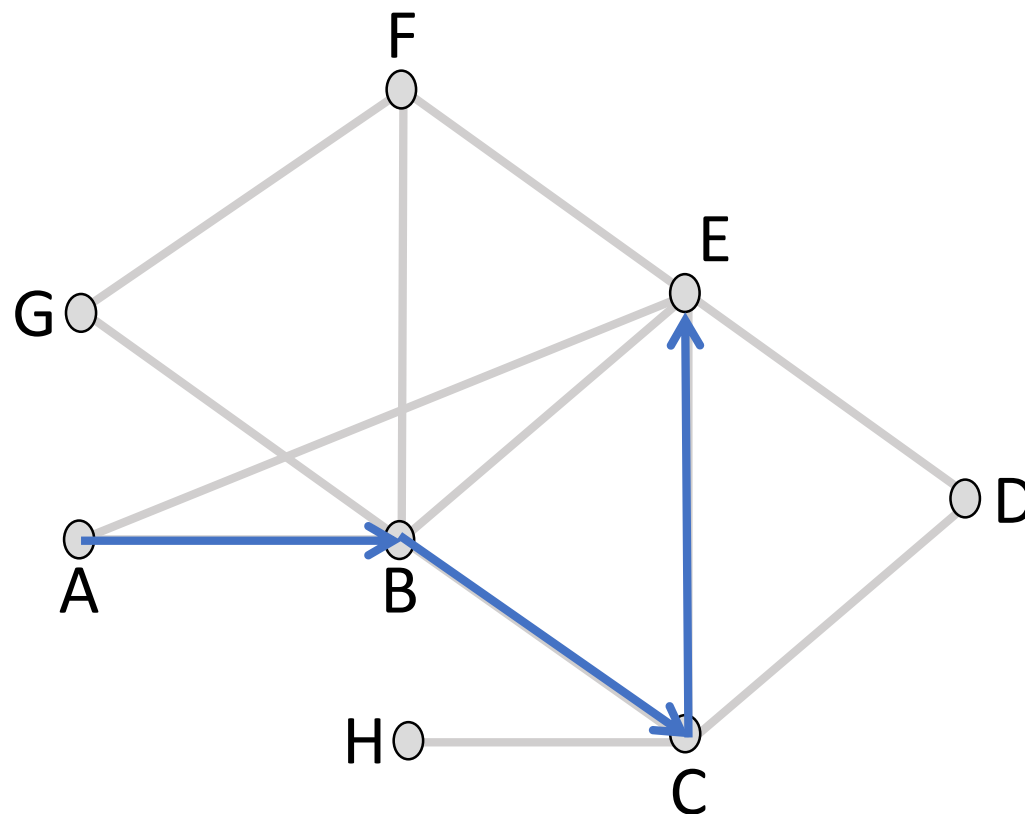
- Велики број могућности:

- Кашњење:  
избегава заобилазне путеве

- Проток:  
избегава споре везе

- Новац:  
избегава скупе везе

- Број хопова:  
смањује искоришћеност комуникационе опреме



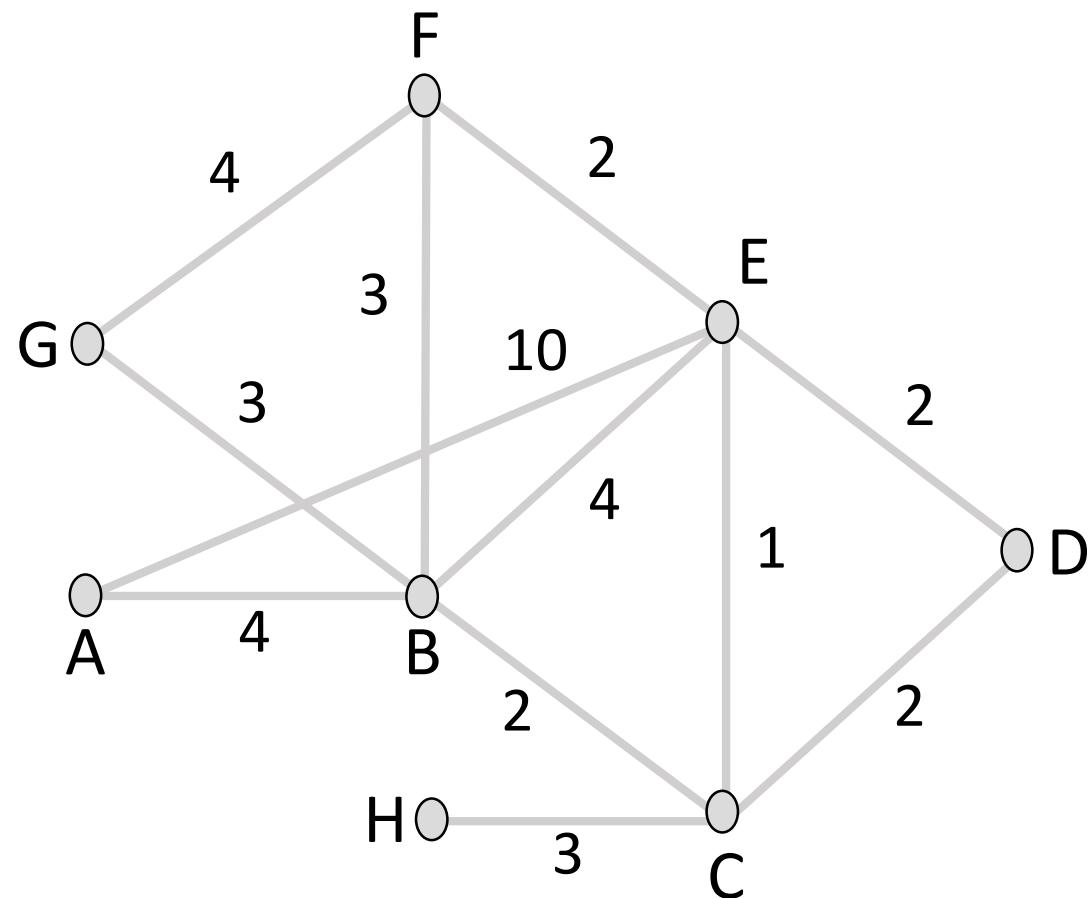
# Најкраћи путеви

Пронаћи најкраћи пут за  $A \rightarrow E$

Претпоставимо да је граф:

1. Неусмерен (пакети у оба смера)
2. И да има симетричне трошкове

Могуће је моделовати алгоритме и за неусмерене, асиметричне графове.



## Најкраћи путеви (2)

ABCE је најкраћи пут

$$\text{dist}(ABCE) = 4 + 2 + 1 = 7$$

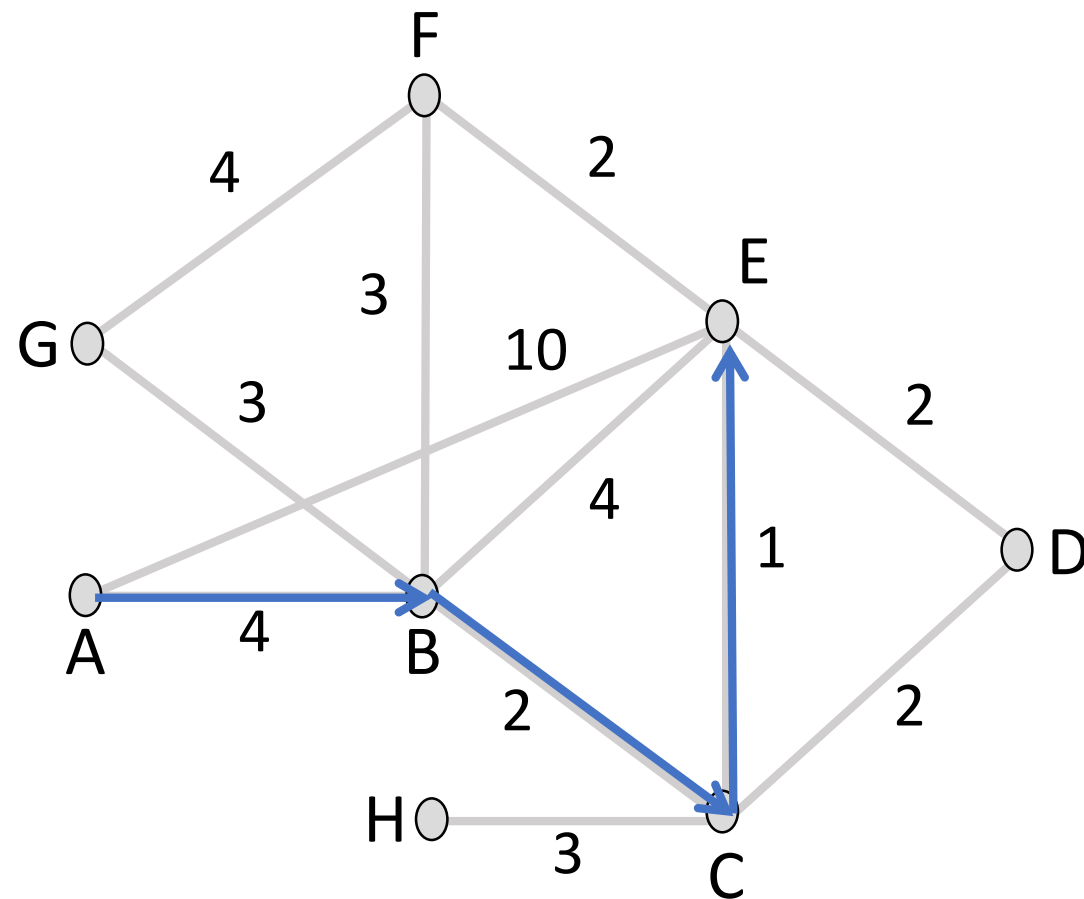
Она има трошак мањи од осталих путева:

$$\text{dist}(ABE) = 8$$

$$\text{dist}(ABFE) = 9$$

$$\text{dist}(AE) = 10$$

$$\text{dist}(ABCDE) = 10$$



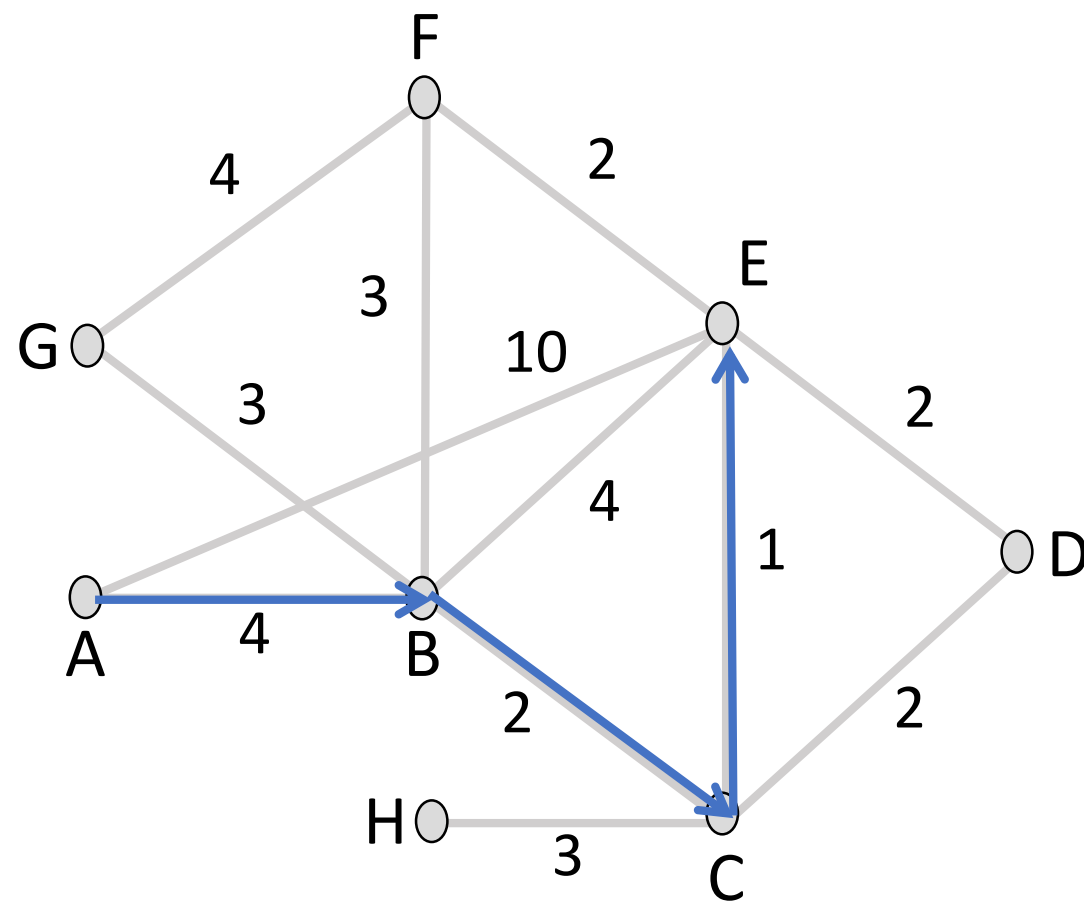
# Најкраћи путеви (3)

## Принцип оптималности:

Приметити да су сегменти оптималних (најкраћих путева) такође најкраћи путеви

ABCE је најкраћи пут између A и E

→ Али су и ABC, AB, BCE, BC, CE најкраћи путеви између A и C, A и B, итд.

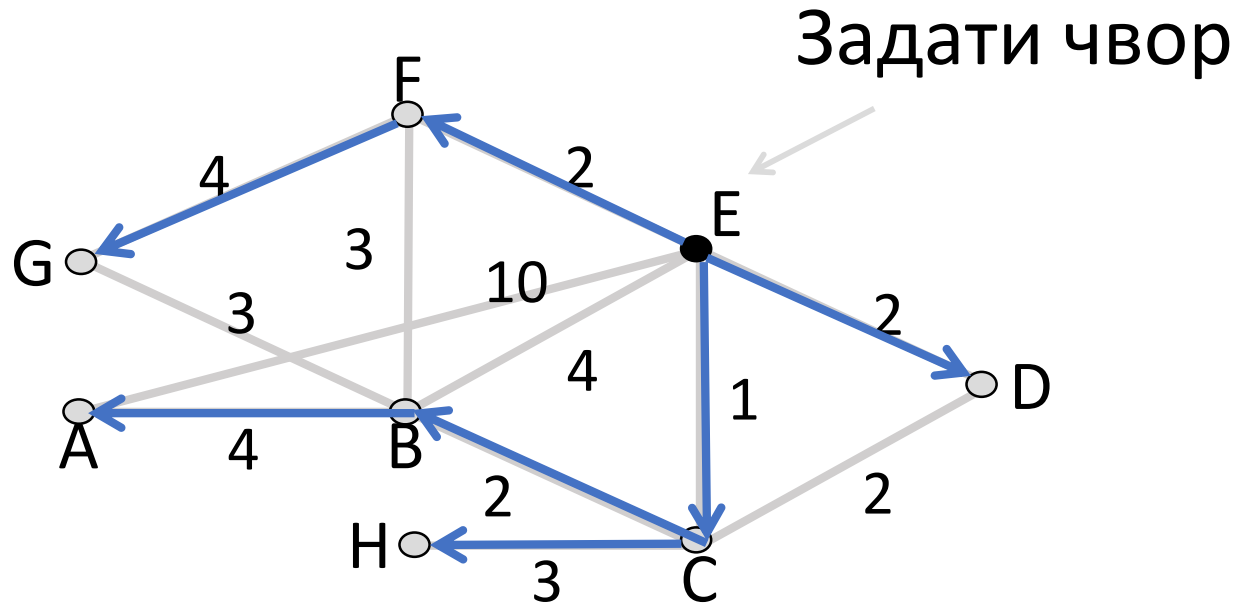


# Рутирање

Дијкстрин алгоритам за најкраће путеве

# Тема

- Дијкстрин алгоритам
  - Рачуна најкраће путеве између задатог чвора и свих осталих чворова
  - Резултат је дрво

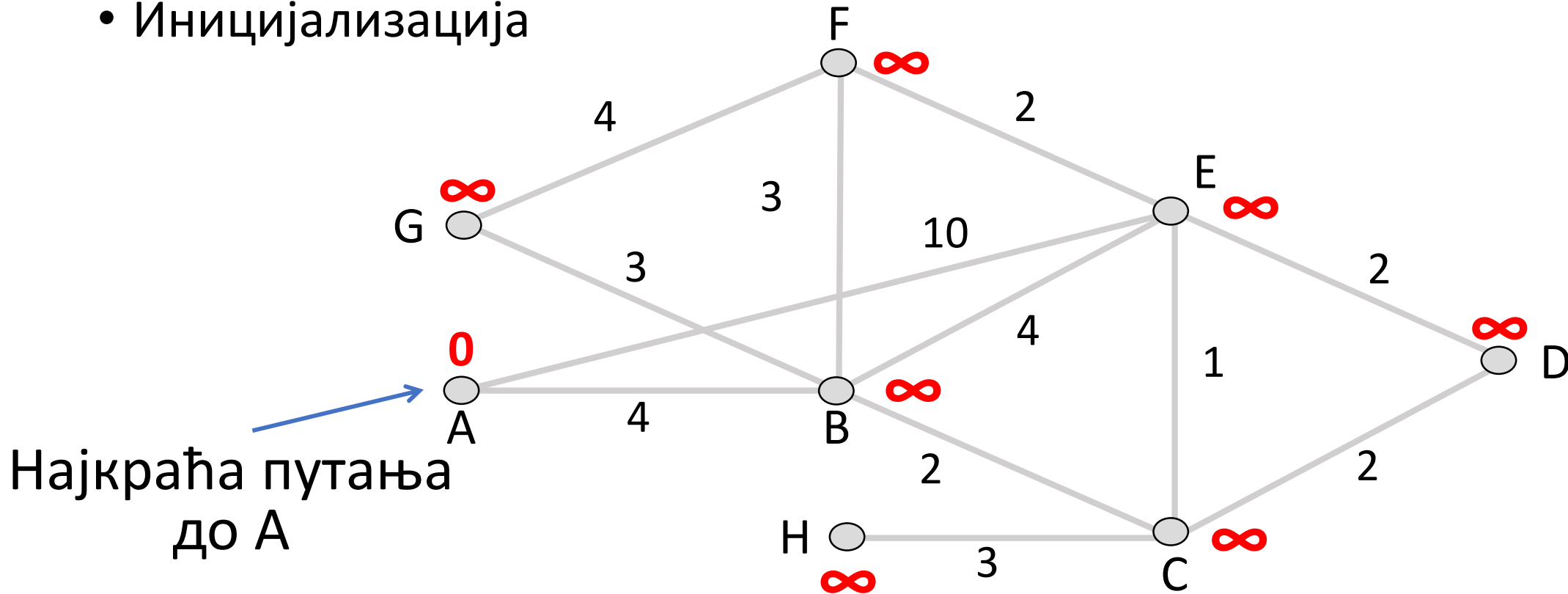


# Дијкстрин алгоритам

- Поставимо све чворове као привремене
- Поставимо актуелне удаљености између задатог и свих чворова:
  - На вредност 0 ако је удаљеност до самог себе
  - На вредност  $\infty$  (бесконачно) за све остале чворове
- Док има привремених чворова:
  - Узми привремени чвор  $X$ , који има најмању удаљеност од задатог чвора
  - Избаци  $X$  из скупа привремених чворова и додај одговарајућу везу ка њему у дрво
  - Умањи удаљености чворова суседних са  $X$  у складу са новододатом удаљеношћу

# Дијкстрин алгоритам (2)

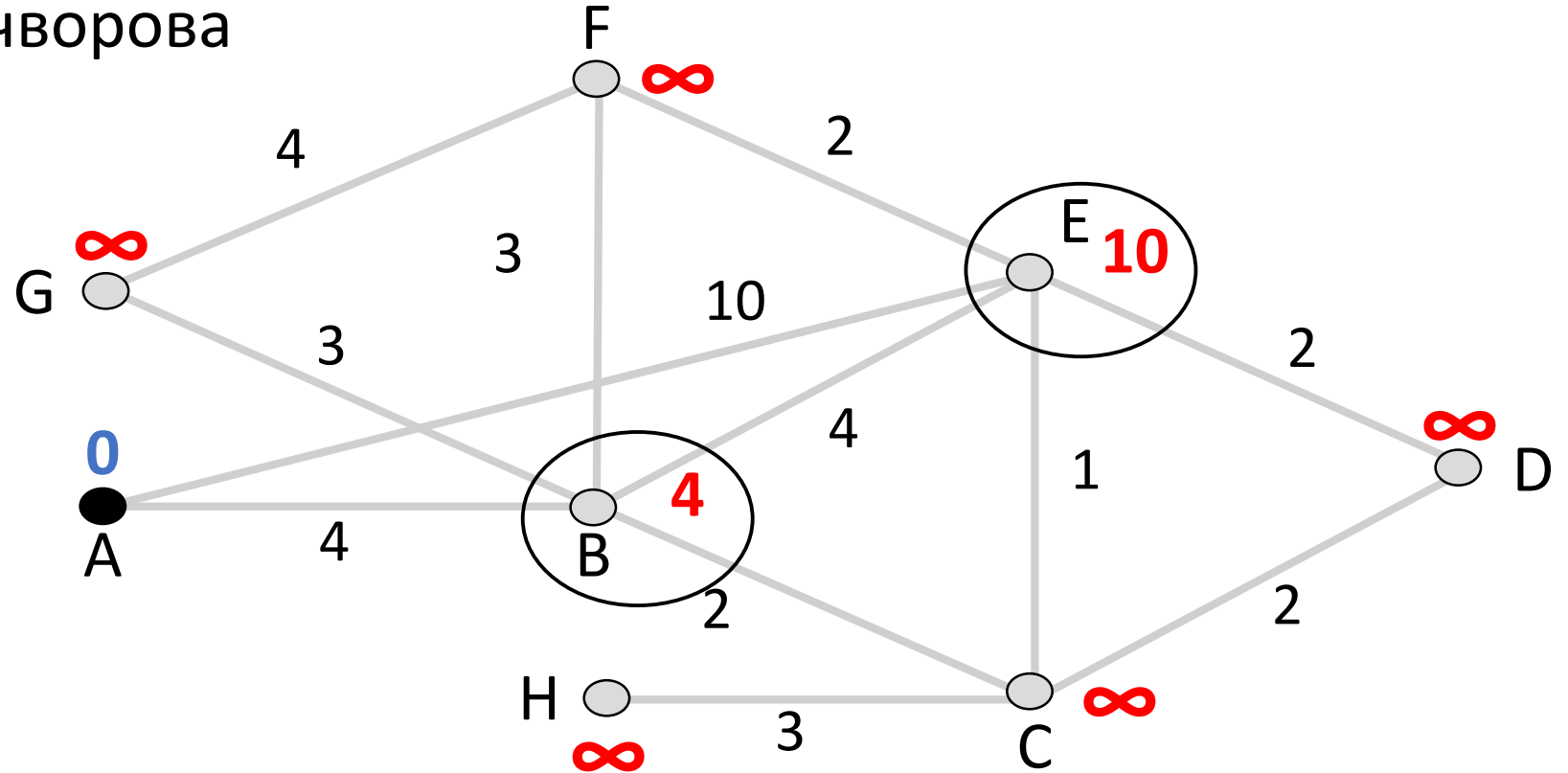
- Иницијализација





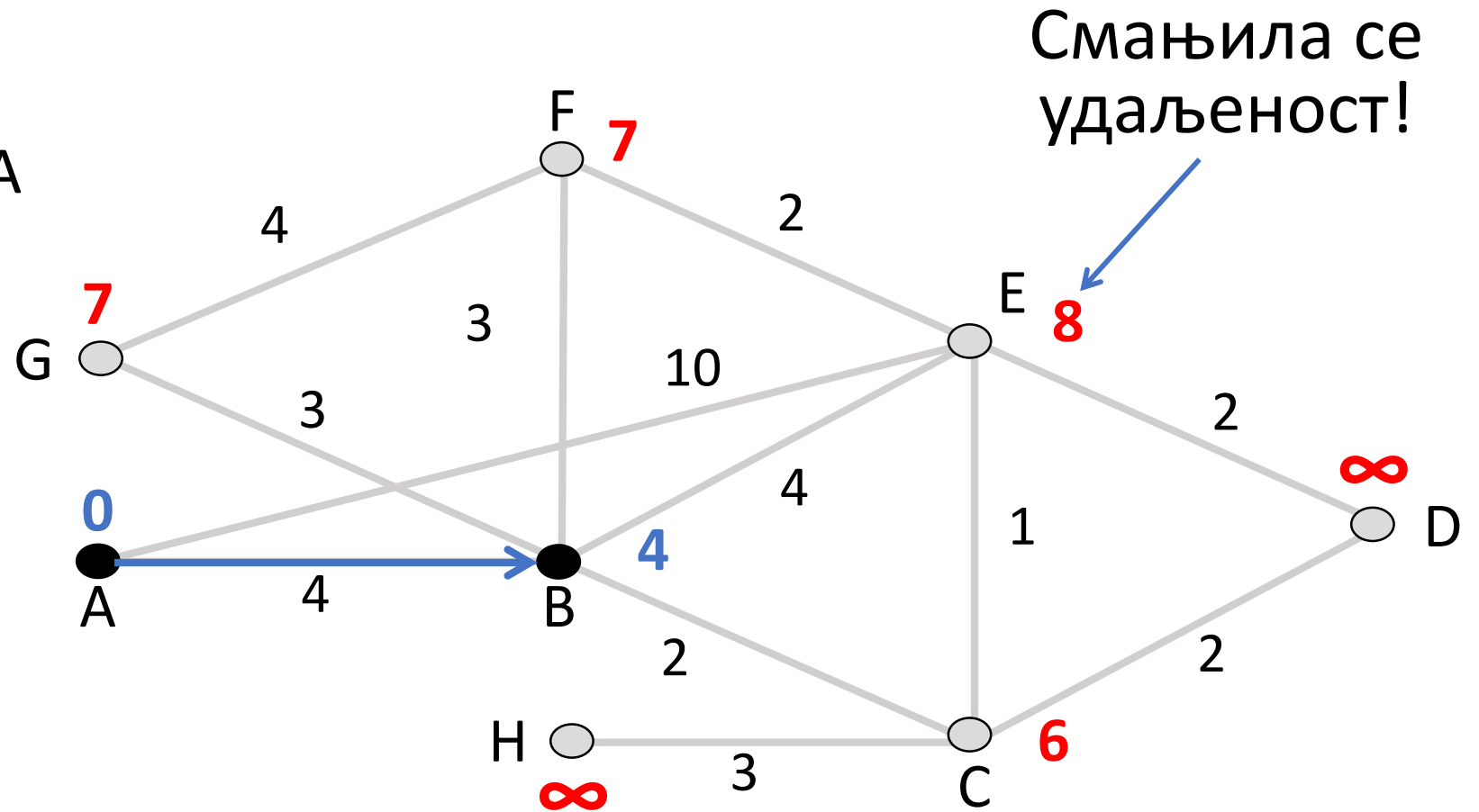
# Дијкстрин алгоритам (3)

- Умањи удаљености чворова суседних са А



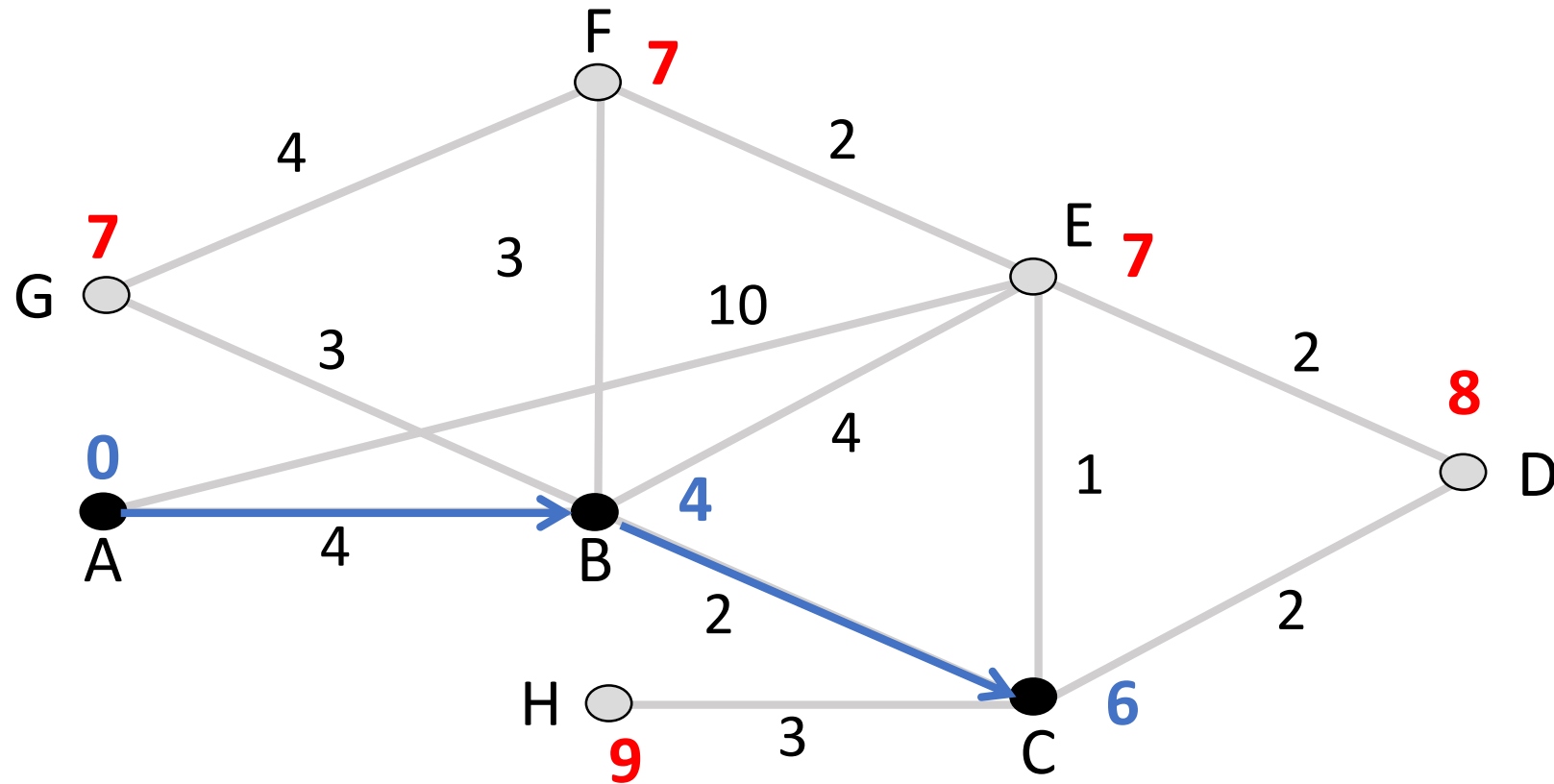
# Дијкстрин алгоритам (4)

- Бирамо В, јер је он најближи А
- Такође ажурирамо чворове који су суседни са В
- У следећем кораку бирамо С, јер је он нови најближи



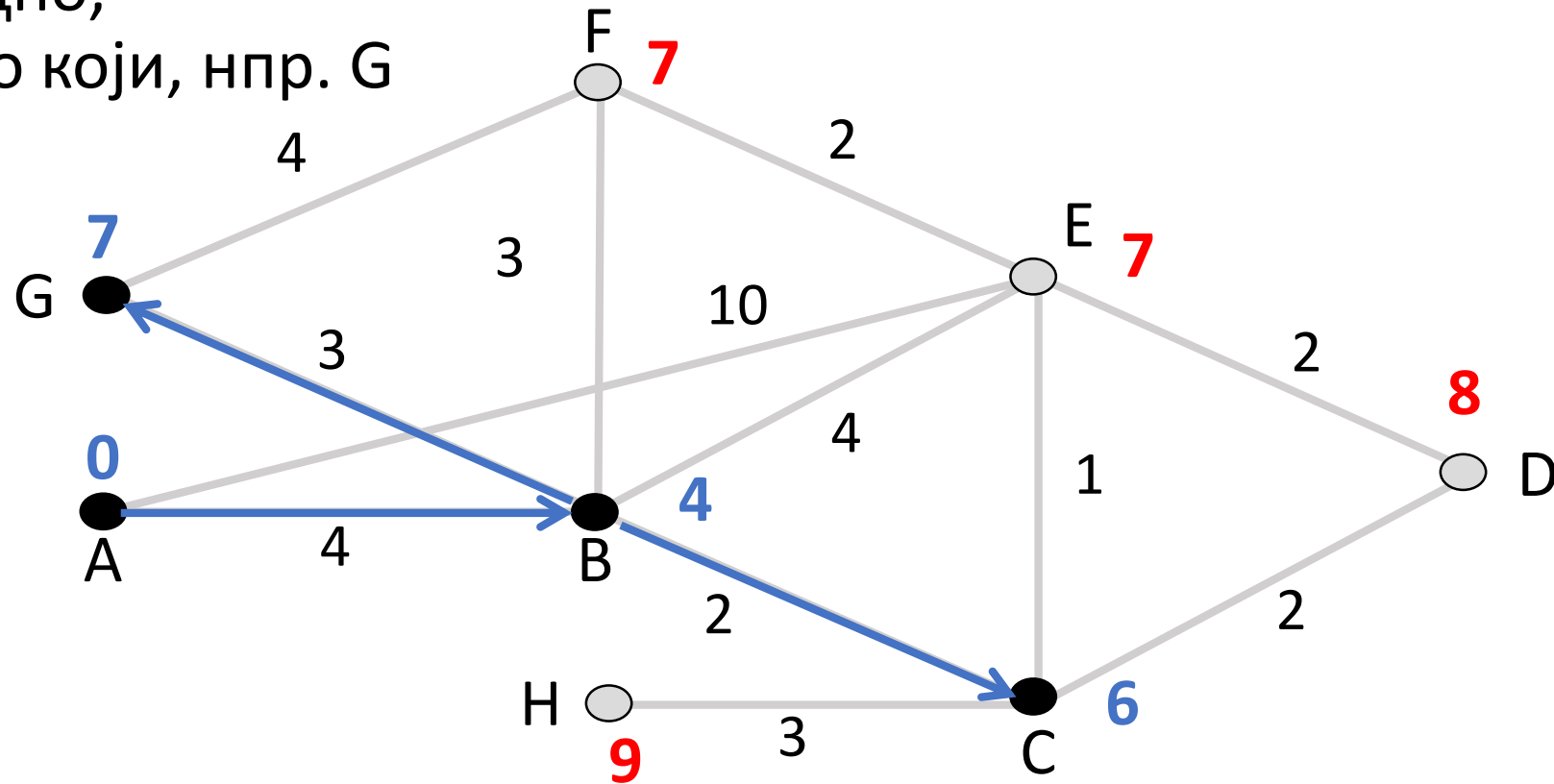
# Дијкстрин алгоритам (5)

- Бирамо С



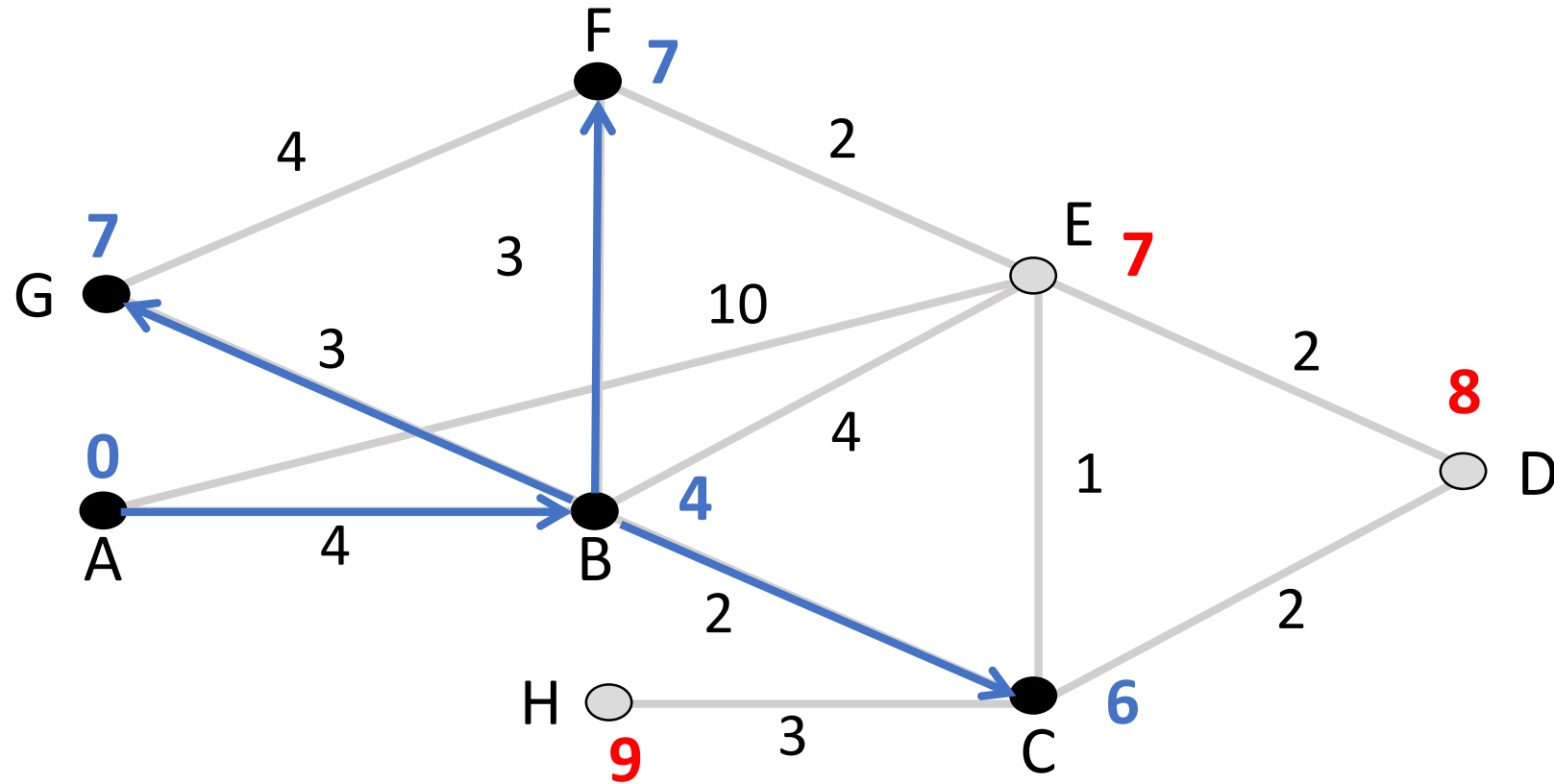
# Дијкстрин алгоритам (6)

- Кад је свеједно, бирамо било који, нпр. G



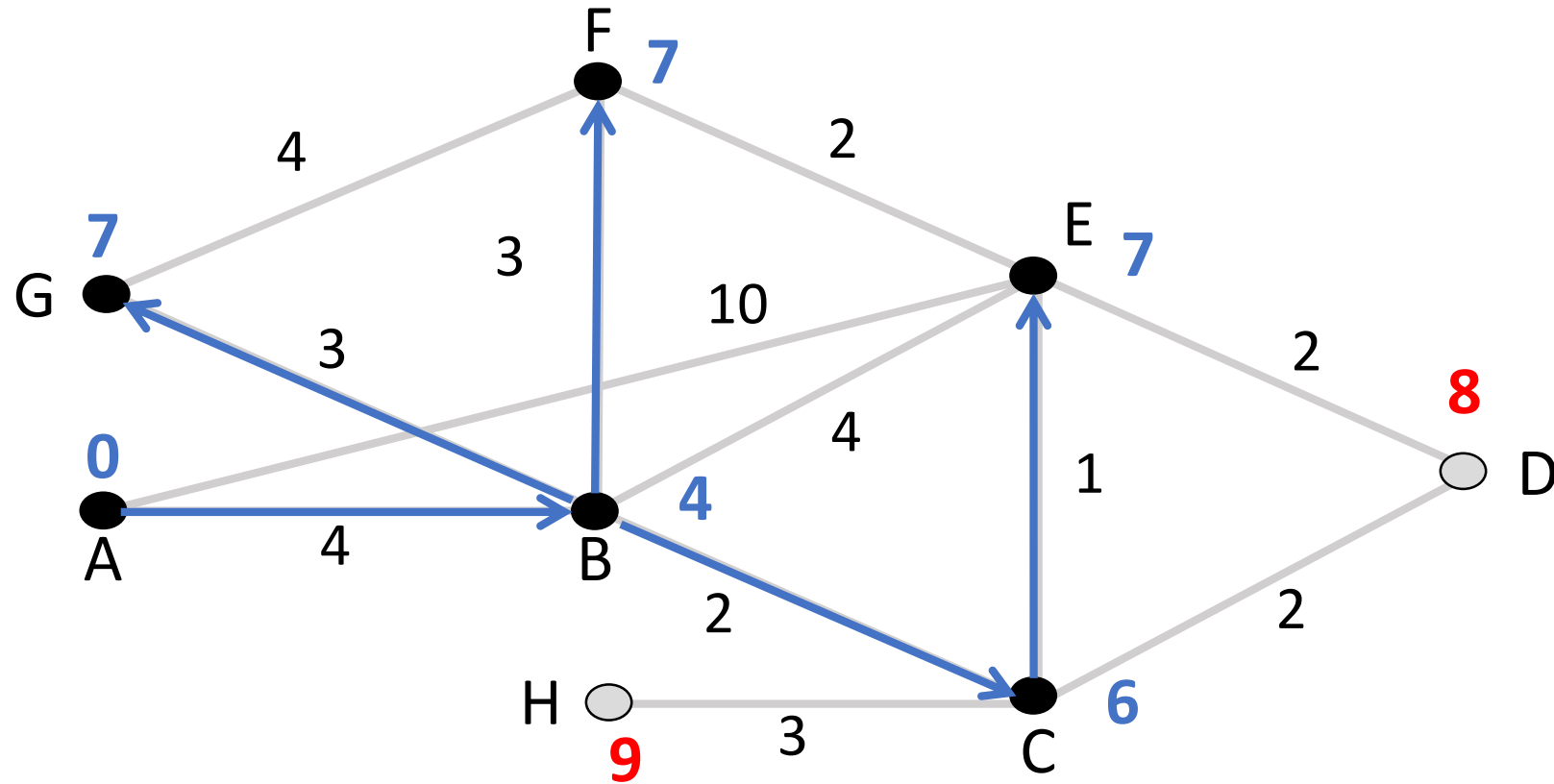
# Дијкстрин алгоритам (7)

- Бирамо F



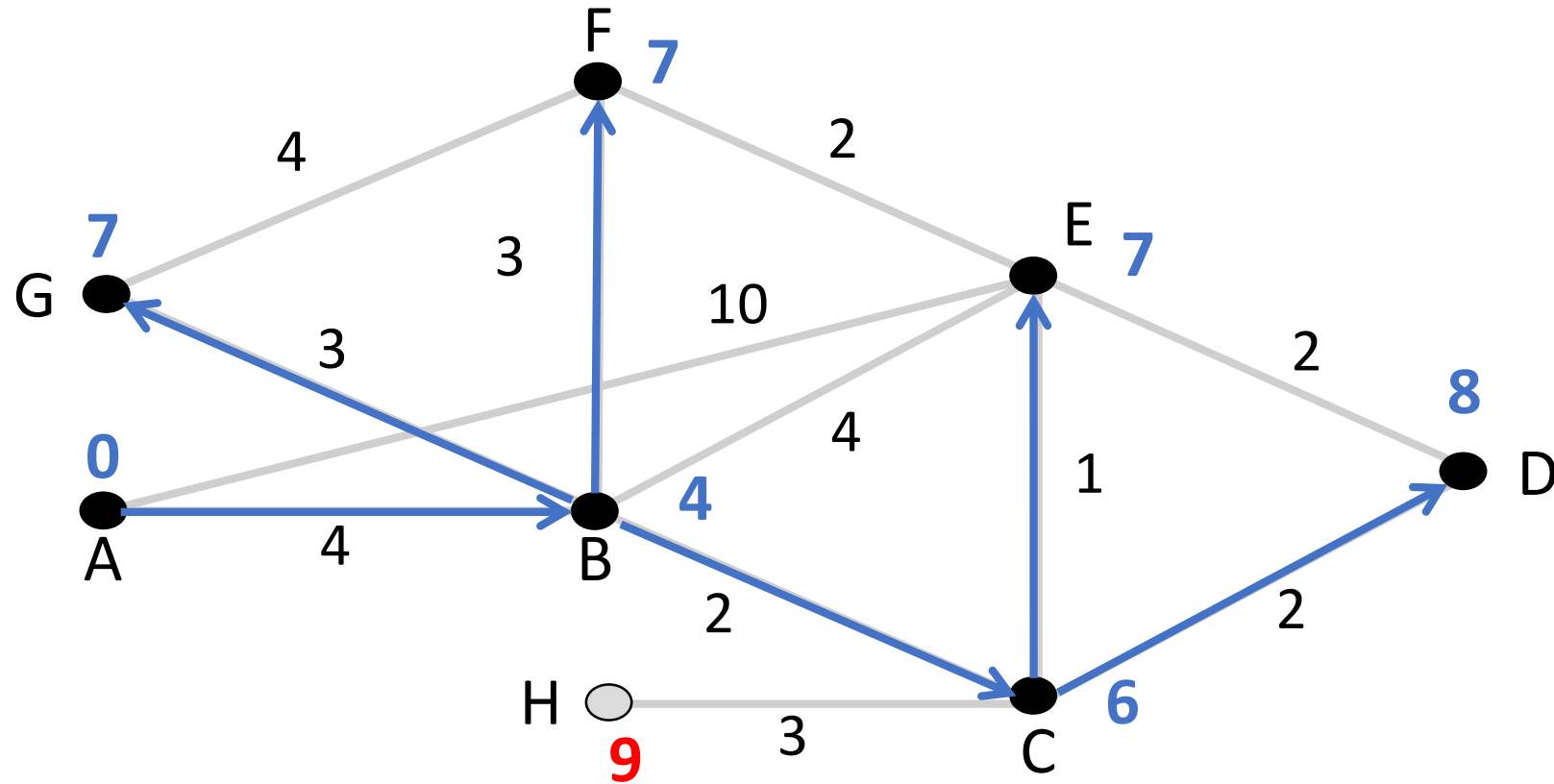
# Дијкстрин алгоритам (8)

- Бирамо E



# Дијкстрин алгоритам (9)

- Бирамо D



# Дијкстрин алгоритам (10)

- И коначно Н

