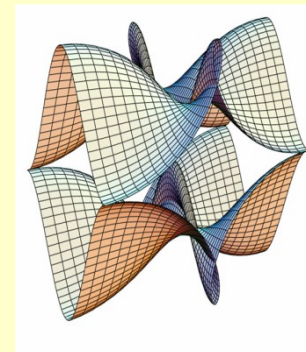




Sveučilište u Zagrebu  
PMF – Matematički odsjek

MREŽE RAČUNALA  
Predavanja 2022/2023



# Poglavlje 10: Algoritmi za usmjeravanje

Sastavio: Robert Manger; Prilagodio: Zvonimir Bujanović  
27.10.2014

# Problem usmjeravanja (1)

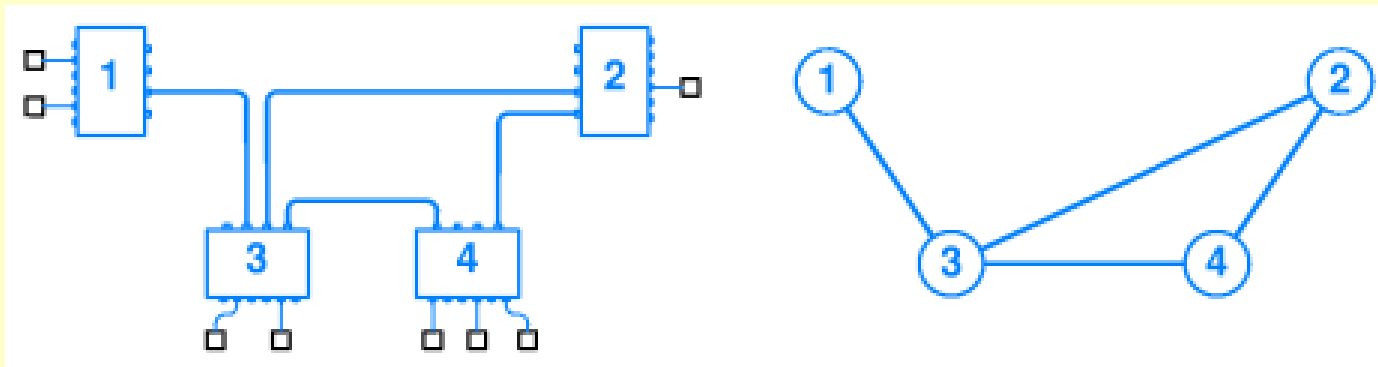
- Vidjeli smo da svaka paketna sklopka u WAN-u mora imati upisanu tablicu usmjeravanja.
- *Problem usmjeravanja* sastoji se od računanja i upisivanja tablice usmjeravanja u svakoj od sklopki, tako da se osigura univerzalno usmjeravanje i optimalnost putova.
- Problem se može riješiti na dva načina:
  - *Ručno*. Administrator mreže upisuje retke u svaku od tablica. Izvedivo za vrlo male mreže.
  - *Automatski*. Tablice se računaju korištenjem softvera koji je instaliran u svakoj sklopki. Taj softver radi u skladu s određenim *algoritmom za usmjeravanje*.

# Problem usmjeravanja (2)

- Automatsko usmjeravanje dalje se dijeli na:
  - *Statičko usmjeravanje*. Softver generira tablicu u trenutku pokretanja sklopke. Tablica se dalje ne mijenja.
  - *Dinamičko usmjeravanje*. Softver u trenutku pokretanja sklopke stvara polaznu verziju tablice. Softver kasnije mijenja tablicu ako se uvjeti u mreži promijene.
- Statičko usmjeravanje je jednostavnije i manje opterećuje mrežu. Dinamičko usmjeravanje je fleksibilnije pa se danas češće koristi.

# Prikaz WAN-a pomoću grafa (1)

- Svi algoritmi za usmjeravanje zasnivaju se na prikazu WAN-a pomoću *grafa*.
- Čvorovi grafa predstavljaju paketne sklopke, a *bridovi* predstavljaju izravne veze između sklopki.



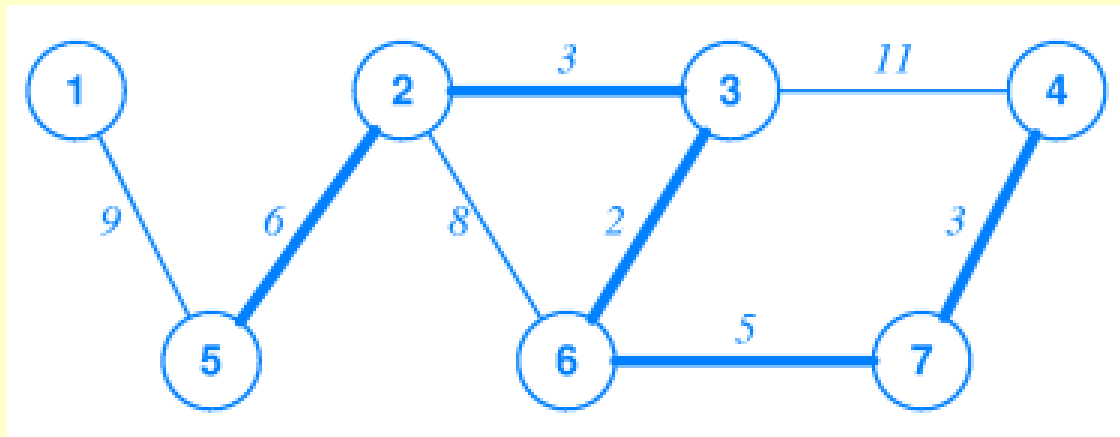
- Graf ističe bitna svojstva povezanosti WAN-a, a zanemaruje nebitne detalje kao što su fizički smještaj sklopki i priključena računala.

# Prikaz WAN-a pomoću grafa (2)

- Svakom bridu grafa pridružuje se njegova *duljina* (težina, cijena). Duljina puta u grafu računa se kao zbroj duljina pripadnih bridova.
- Duljine mogu imati razne interpretacije.
  - Ako svim bridovima zadamo duljinu 1, tada je duljina puta jednaka broju skokova od polazišta do odredišta.
  - Ako duljina brida odgovara vremenu prijenosa duž odgovarajuće komunikacijske linije, tada duljina puta izražava ukupno vrijeme potrebno za prijenos podatka od polazišta do odredišta.
  - Duljina brida može biti i cijena koju plaćamo za najam komunikacijske linije po prenesenom Mbit. Duljina puta tada odgovara ukupnoj cijeni prijenosa jednog Mbit od polazišta do odredišta.

# Grafovski formulacija problema (1)

- Problem usmjeravanja u WAN-u zapravo se svodi na pronalaženje najkraćih putova između svih parova čvorova u pripadnom grafu.
- Na grafu sa slike, najkraći put između čvorova 4 i 5 označen je podebljanim bridovima i njegova duljina je 19. Slično bi mogli odrediti najkraći put za sve parove čvorova.

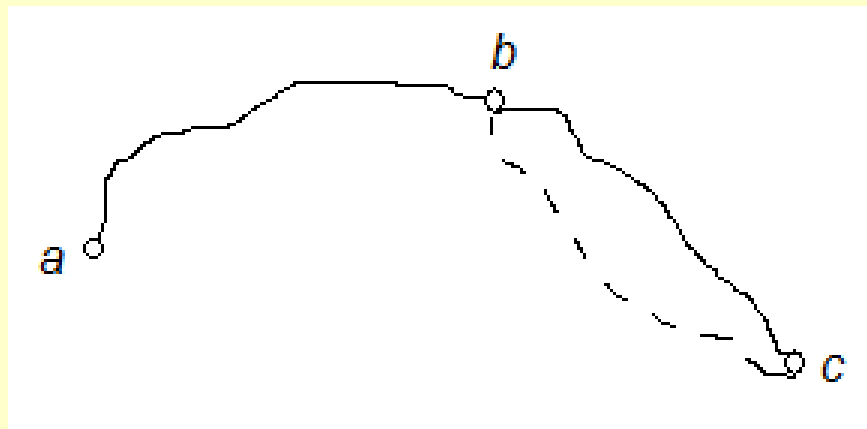


# Grafovska formulacija problema (2)

- Najkraći put je optimalan po onom kriteriju kojeg smo odabrali kad smo zadavali duljine bridova.
  - Na primjer, ako su duljine bridova jedinične cijene najma komunikacijskih linija, tada najkraći put zapravo određuje najjeftiniji način prijenosa podataka.
- Algoritam za usmjeravanje u svakoj sklopki rješava *single-source* varijantu problema najkraćih putova.
  - Dakle, u svakoj sklopki traži se najkraći put od dotičnog čvora kao polazišta do bilo kojeg drugog čvora kao odredišta.
  - Zatim se za svaki takav najkraći put u tablicu usmjeravanja upisuje sljedeći skok koji odgovara početnom bridu puta.

# Bellmanov princip optimalnosti

- Da bi reproducirali bilo koji najkraći put, dovoljno je pamtiti samo njegov početni brid. Razlog zašto je to moguće naziva se *Bellmanov princip optimalnosti*.
- Princip kaže da ako najkraći put od čvora  $a$  do čvora  $c$  prolazi čvorom  $b$ , tada dio tog puta između  $b$  i  $c$  ujedno predstavlja najkraći put između  $b$  i  $c$ .





# Statičko usmjeravanje pomoću Dijkstrinog algoritma

- Graf koji prikazuje građu WAN-a mora biti poznat unaprijed, te se ne može mijenjati. Duljine bridova moraju biti nenegativne.
- Svaka paketna sklopka pohranjuje u svojoj memoriji opis cijelog grafa, te rješava neovisno o drugim sklopkama svoj single-source problem najkraćih putova.
- Jedan single-source problem rješava se poznatim Dijkstrinim algoritmom. Detalji algoritma vidljivi su iz sljedećeg pseudokoda -  $S$  je skup odredišta za koje još nije određena najkraća udaljenost.
- Dokaz korektnosti Dijkstrinog algoritma izvodi se u kolegiju o diskretnoj matematici.

# Pseudokod Dijkstrinog algoritma (1)

Ulaz:

Graf s nenegativnim duljinama bridova i istaknutim čvorom – polazištem.

Izlaz:

Polje  $D$  s najkraćim udaljenostima.  $D[v]$  je najkraća udaljenost od polazišta do čvora  $v$ .

Tablica  $R$  sa sljedećim skokovima.  $R[v]$  je sljedeći skok od polazišta prema  $v$ .

Postupak:

inicijaliziraj skup  $S$  tako da sadrži sve čvorove osim polazišta;

inicijaliziraj  $D$  tako da je  $D[v]$  duljina brida od polazišta do  $v$  ako takav brid postoji, odnosno  $\infty$  inače;

inicijaliziraj  $R$  tako da je  $R[v]$  jednako  $v$  ako postoji brid od polazišta do  $v$ , odnosno prazno inače;

# Pseudokod Dijkstrinog algoritma (2)

```
dok (S nije prazan) {  
    izaberi  $u$  iz  $S$  takav da je  $D[u]$  minimalno;  
    ako ( $D[u]$  je jednak  $\infty$ ) {  
        greška: graf nije povezan; prekini rad;  
    }  
    izbaci  $u$  iz  $S$ ;  
    za svaki  $v$  takav da postoji brid  $(u,v)$  {  
        ako ( $v$  je još uvijek u  $S$ ) {  
             $c = D[u] + \text{duljina brida } (u,v)$ ;  
            ako ( $c < D[v]$ ) {  
                 $R[v] = R[u]$ ;  
                 $D[v] = c$ ;  
            }  
        }  
    }  
}
```

# Dinamičko usmjeravanje pomoću vektora udaljenosti (1)

- Graf koji prikazuje građu WAN-a nije nigdje eksplicitno pohranjen. Pojedina paketna sklopka zna samo koji su njezini susjedi, te kolike su duljine bridova između nje i njezinih susjeda.
- Svaki redak tablice usmjeravanja unutar sklopke sadrži tri polja: odredište, sljedeći skok, udaljenost odredišta duž puta koji odgovara sljedećem skoku (otuda naziv *vektor udaljenosti*).
- Sklopke povremeno šalju podatke iz svoje tablice po mreži svojim susjedima. Riječ je o *porukama usmjeravanja* (routing messages) koje sadrže parove odredišta i udaljenosti.
- Kad god pojedina sklopka primi od svog susjeda poruku navedenog tipa, ona analizira tu poruku i mijenja svoju vlastitu tablicu ukoliko susjed ima kraći put do nekog odredišta.

# Dinamičko usmjeravanje pomoću vektora udaljenosti (2)

- Nakon dovoljnog broja iteracija, informacije o građi grafa implicitno će se proširiti cijelom mrežom. Svaka sklopka sagradit će ispravnu tablicu.
- Ako se naknadno pojave nove veze u mreži, tablice će se i dalje popravljati, dakle sklopke će uvažiti pojavu novih boljih (kraćih) putova.
- Ako se neke veze u mreži naknadno prekinu, potrebno je resetirati tablice i početi cijeli postupak iznova. Nakon dovoljnog broja iteracija sklopke će pronaći alternativne putove koji zaobilaze prekinute veze.
- Detalji algoritma zasnovanog na vektoru udaljenosti vidljivi su iz sljedećeg pseudokoda. Taj pseudokod odnosi se na rad jedne sklopke (čvora).

# Pseudokod algoritma zasnovanog na vektorima udaljenosti (1)

Ulaz:

Identifikator lokalnog čvora, lokalna tablica usmjeravanja, duljine svih bridova prema susjednim čvorovima, niz ulaznih poruka za usmjeravanje.

Izlaz:

Niz ažuriranih verzija lokalne tablice usmjeravanja.

Postupak:

inicijaliziraj tablicu usmjeravanja tako da sadrži samo jedan redak: odredište jednako lokalnom čvoru, sljedeći skok prazan, udaljenost 0;

ponavljaj zauvijek {

čekaj da stigne iduća poruka za usmjeravanje;

neka je pošiljalatelj te poruke susjedni čvor  $N$ ;

# Pseudokod algoritma zasnovanog na vektorima udaljenosti (2)

```
za svaki redak iz poruke {  
  neka je  $V$  odredište u tom retku, a  $D$  udaljenost;  
   $C = D +$  duljina brida kojim je poruka stigla;  
  pregledaj i ažuriraj lokalnu tablicu usmjeravanja:  
  ako (u tablici ne postoji put do  $V$ ) {  
    dodaj redak s odredištem  $V$ , sljedećim  
    skokom  $N$  i udaljenošću  $C$ ;  
  } inače ako (postoji put i sljedeći skok je  $N$ ) {  
    promijeni udaljenost za taj put u  $C$ ;  
  } inače ako (postoji put s udaljenošću  $> C$ ) {  
    promijeni sljedeći skok u  $N$ , udaljenost u  $C$ ;  
  }  
}
```