

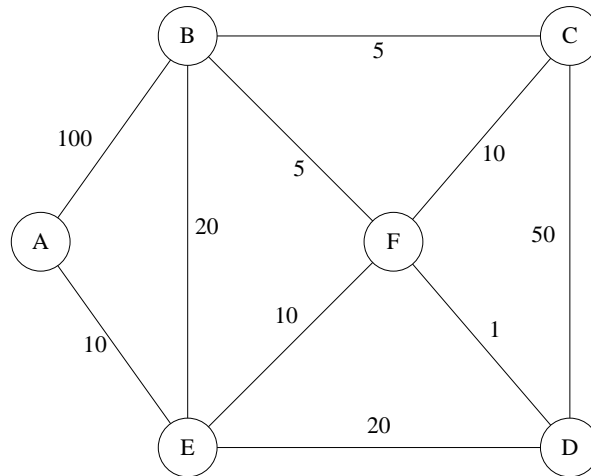
## Prova scritta

Mauro Brunato

Giovedì 15 giugno 2006

### Esercizio 1

Si consideri la rete di sei nodi (A, ..., F) rappresentata in figura.



Le etichette degli archi rappresentano la loro velocità di trasmissione in megabit al secondo; tutti i collegamenti sono simmetrici.

**1.1)** Determinare la tabella di instradamento del nodo C supponendo che l'algoritmo di instradamento si basi sui cammini minimi con metrica data dal numero di archi attraversati, indipendentemente dalla velocità di trasmissione.

Nelle stesse ipotesi, disegnare inoltre l'albero dei cammini minimi avente il nodo A come radice.

**1.2)** Calcolare la tabella di instradamento del nodo C supponendo che l'algoritmo di instradamento si basi sui cammini minimi se il costo di un arco è inversamente proporzionale alla sua velocità di trasmissione.

Nelle stesse ipotesi, disegnare inoltre l'albero dei cammini minimi avente il nodo A come radice.

**1.3)** Supponendo che la latenza di ogni arco sia 5 millisecondi, quanto tempo è necessario per trasferire 1000 pacchetti dal nodo A al nodo C con protocollo *stop and wait* nella rete se il routing è stato determinato come specificato al punto 1.1?

Quanto tempo è invece necessario nell'ipotesi del punto 1.2?

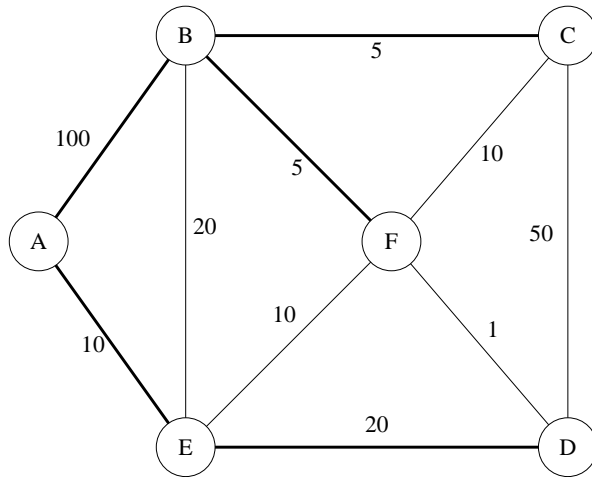
Si supponga che i pacchetti siano molto piccoli, trascurare dunque i tempi di immissione nella rete.

### Soluzione —

**1.1)** La tabella di instradamento di C è la seguente:

Destinazione	Prossimo	Costo
A	B	2
B	B	1
C	—	0
D	D	1
E	F	2
F	F	1

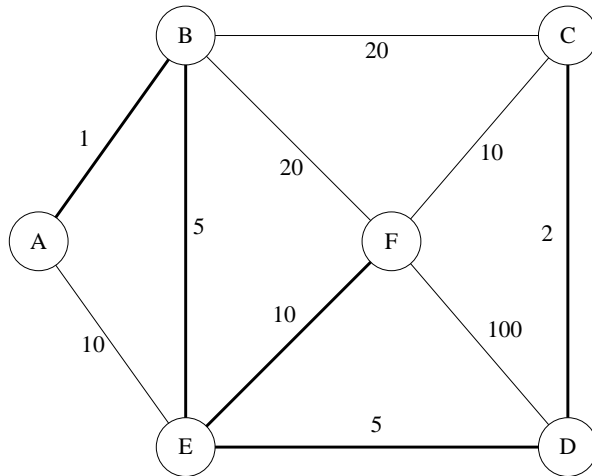
L'albero dei cammini minimi dal nodo A è il seguente



1.2) La tabella di instradamento di C è la seguente (per i costi, vedere la figura successiva):

Destinazione	Prossimo	Costo
A	D	13
B	D	12
C	—	0
D	D	2
E	D	7
F	F	10

L'albero dei camini minimi dal nodo A è il seguente (notare che le velocità sono sostituite da costi inversamente proporzionali):



1.3) Sono possibili due interpretazioni:

- Se il protocollo *stop and wait* è applicato a livello trasporto, allora il protocollo agisce end-to-end, e l'invio degli ACK è realizzato dal nodo C.  
 In tal caso, nelle condizioni del punto 1.1 ciascun pacchetto attraversa 2 archi, e solo alla ricezione il nodo C invia il corrispondente ACK, che viaggia per gli stessi due archi. Dalla trasmissione del pacchetto alla ricezione dell'ACK, dunque, il nodo A deve attendere  $4 \times 5\text{ms}$ , e la trasmissione di 1000 pacchetti richiede 20s.  
 Nelle condizioni del punto 1.2, la situazione è peggiore, perché ciascun pacchetto da A a C deve attraversare 4 archi, raddoppiando così il tempo, che sale a 40s.
- Se invece il protocollo *stop and wait* è realizzato a basso livello, e viene attuato indipendentemente per ogni arco, allora il nodo A potrà emettere un pacchetto ogni 10ms, e la trasmissione durerà 10s indipendentemente dal numero di archi da attraversare.

## Esercizio 2

Ai tempi in cui l'indirizzamento IP era basato su classi, una grossa azienda acquistò la sottorete di classe B 137.129.0.0. Col tempo, due grosse porzioni della rete sono state utilizzate all'interno dell'azienda, l'intervallo

137.129.10.0 ... 137.129.33.255

e l'intervallo

137.129.65.0 ... 137.129.255.255

Con l'avvento del sistema CIDR, l'azienda ha l'opportunità di rivendere le porzioni inutilizzate del proprio spazio di indirizzamento.

**2.1)** Nell'ipotesi di suddividere la porzione inutilizzata della propria sottorete in segmenti di classe C, quante sottoreti sarà in grado di rivendere l'azienda?

**2.2)** Qual è, in notazione CIDR, la più grande sottorete che può essere ricavata dallo spazio lasciato libero dalla rete aziendale?

**2.3)** Supponendo che sia economicamente vantaggioso rivendere la parte libera in segmenti più grandi possibile, qual è la suddivisione dello spazio libero nel minor numero possibile di sotto reti? Elencare le sottoreti disponibili in notazione CIDR.

**Soluzione** — Per tutto l'esercizio, considereremo fissi i primi due ottetti della sottorete, mentre il quarto sarà sempre libero di variare. Tratteremo quindi soltanto il terzo ottetto.

**2.1)** Le sottoreti di classe C sono quelle che lasciano libero solo il quarto ottetto. Le reti ancora libere sono quindi le dieci fra 0 e 9, e le trentuno fra 34 e 64. In totale, dunque, l'azienda può rivendere 41 reti di classe C.

**2.2)** La sottorete più grande che può essere ricavata nello spazio libero è 137.129.48.0/20. Vedere il prossimo punto per la spiegazione.

**2.3)** Consideriamo il terzo ottetto in notazione binaria. Il primo valore libero è 00000000. Per ottenere la sottorete più grande, possiamo lasciare liberi gli ultimi tre bit. Se ne lasciassimo liberi quattro, infatti, la rete occuperebbe anche i valori da 10 a 15, che non sono liberi. Quindi, la più grande sottorete che contiene 0 è

137.129.0.0/21 (137.129.0.0—137.129.7.255).

Rimangono libere le reti 8 e 9 (rispettivamente 00001000 e 10001001, in binario), che possono essere accorpate:

137.129.8.0/23 (137.129.8.0—137.129.9.255).

Con ciò, il primo intervallo libero è completamente impegnato.

In seguito, dobbiamo accorpare il più possibile le sottoreti da 34 (00100010) a 64 (01000000). La rete più grande contenente la 34 lascia libero solo un bit del terzo ottetto (corrispondente alle reti di classe C 34 e 35):

137.129.34.0/23 (137.129.34.0—137.129.35.255).

Dopo aver impegnato questa rete, il successivo valore 36 (00100100) è la base per una sottorete che lascia liberi due bit:

137.129.36.0/22 (137.129.36.0—137.129.39.255).

Dopo aver impegnato questa rete, il successivo valore 40 (00101000) è la base per una sottorete che lascia liberi tre bit:

137.129.40.0/21 (137.129.40.0—137.129.47.255).

Dopo aver impegnato questa rete, il successivo valore 48 (00110000) è la base per una sottorete che lascia liberi quattro bit:

137.129.48.0/20 (137.129.48.0—137.129.63.255).

Rimane ancora libera la sottorete 64, che non può essere accorpata a nulla, visto che la 65 è occupata:

137.129.64.0/24 (137.129.64.0—137.129.64.255).

### **Esercizio 3**

Un'applicazione vuole inviare alcuni byte di informazione a un'altra applicazione, non necessariamente nello stesso calcolatore e possibilmente in sottoreti diverse, per mezzo di un solo pacchetto UDP.

**3.1)** Quali dati deve conoscere l'applicazione mittente per far giungere correttamente le informazioni all'applicazione destinataria?

**3.2)** Fornire una descrizione schematica dei passi compiuti dalle pile protocollari delle macchine interessate al trasporto del pacchetto, dalla sorgente alla destinazione.

#### **Soluzione —**

**3.1)** L'indirizzo IP dell'interfaccia di destinazione e la porta UDP su cui l'applicazione è in ascolto.

**3.2)** La risposta a questa domanda dovrebbe riassumere almeno alcuni fra i seguenti argomenti:

- il ruolo degli strati da trasporto a fisico nella comunicazione di un pacchetto (aggiunta di intestazioni, multiplexing tra diversi protocolli possibili...),
- la risoluzione degli indirizzi tramite ARP,
- il ruolo del gateway,
- il funzionamento dei router e degli switch.

Si tenga conto che una semplice descrizione delle funzioni della pila protocollare sarebbe completamente fuori tema, visto che il testo richiede di parlare esplicitamente di UDP.