

Tempo a disposizione per lo svolgimento: 1 ora e 30 minuti

Avvertenza: Si ricordi di indicare sui fogli consegnati nome, cognome e numero di matricola

Esercizio 1

Ad una organizzazione è stato assegnato l'indirizzo di rete 192.62.2.0/24. Da questo indirizzo occorre ricavare un numero X di sottoreti, ciascuna in grado di contenere almeno 32 host. Si richiede di:

1. indicare qual è il numero massimo (X) di sottoreti che si possono ricavare ed il numero massimo di host per ciascuna di tali sottoreti. Giustificare la risposta.
2. definire un prefisso di sottorete che permetta la creazione delle sottoreti di cui al punto 1 (ovvero, le X sottoreti in grado di contenere almeno 32 host), indicandolo sia in formato binario che dotted-decimal.
3. Specificare gli indirizzi binari ed in notazione dotted-decimal delle sottoreti
4. Indicare gli indirizzi di broadcast delle varie sottoreti, in formato binario e dotted-decimal

Soluzione:

1) Sono necessari 6 bit per indirizzare i 32 host ($2^6 - 2 = 62$ host). Restano dunque a disposizione 2 bit per il campo "Subnet ID", che consentono di creare X=4 sottoreti, ciascuna in grado di contenere appunto fino a 62 host.

2) Il prefisso di sottorete è: 11111111. 11111111. 11111111.11000000 in binario, 255.255.255.192 in notazione dotted-decimal (ovvero, /26)

3) Basta sostituire le 4 combinazioni (00,01,10,11) ai primi due bit dell'ultimo ottetto (192.62.2.xx000000), ottenendo quindi le seguenti subnet:

-192.62.2.00000000 /26, ovvero 192.62.2.0 /26

-192.62.2.01000000 /26, ovvero 192.62.2.64 /26

-192.62.2.10000000 /26, ovvero 192.62.2.128 /26

-192.62.2.11000000 /26, ovvero 192.62.2.192 /26

4) Basta sostituire "111111" al campo "host ID" delle 4 subnet appena create:

-192.62.2.00111111, ovvero 192.62.2.63

-192.62.2.01111111, ovvero 192.62.2.127

-192.62.2.10111111, ovvero 192.62.2.191

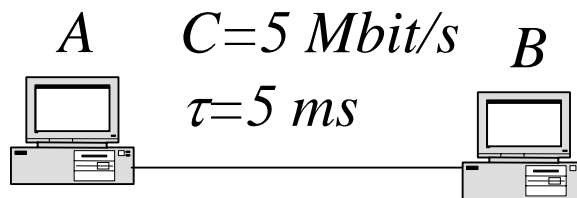
-192.62.2.11111111, ovvero 192.62.2.255

Esercizio 2

Si consideri il collegamento in figura tra i due host A e B. A deve trasferire un messaggio applicativo di 26 kbit usando TCP. Si illustri anzitutto con chiarezza e precisione che cosa rappresentano i parametri *MSS* e *SSTHRESH* di una connessione TCP.

Si calcoli quindi il tempo necessario per trasferire tale messaggio supponendo:

- *MSS*=1000 bit
- Lunghezza degli header di tutti i livelli trascurabile
- La connessione viene aperta da A e la lunghezza dei segmenti di apertura della connessione è trascurabile
- La lunghezza degli ACK è trascurabile
- La *SSTHRESH* è pari a 8 *MSS*



Soluzione:

a) Per la definizione esatta dei parametri MSS e Ssthresh si vedano i lucidi delle lezioni relativi al protocollo TCP. In sintesi:

-MSS=Maximum Segment Size, rappresenta la dimensione massima dei segmenti che vengono usati nella connessione TCP.

-Ssthresh=Slow-Start Threshold, è la variabile di stato di una connessione TCP che consente di distinguere alla connessione stessa se essa si trova in fase di Slow Start (Congestion Window < Ssthresh), oppure in fase di Congestion Avoidance (CWND > Ssthresh).

b) Si veda l'esercizio svolto b.5, della serie di esercizi "b-esercizi su IP e TCP".

Numero segmenti necessari per trasferire i 26 kbit = 26000 bit: $n = 26000/1000 = 26$ segmenti.

Tempo trasmissione 1 segmento sul canale da 5 Mbit/s: $T = 1000/5 * 10^6 = 0.2 \text{ ms}$

Quindi, con riferimento ad una figura del tutto analoga a quella dell'esercizio b.5, la sorgente TCP trasmette: (segmenti in fase di handshake) + 1 + 2 + 4 + 8 + 9 + 2 segmenti.

Infatti la connessione TCP deve anzitutto essere stabilita (fase di scambio dei messaggi di handshake); quindi, poiché CWND parte da 1 MSS, la connessione è inizialmente in fase di slow-start (Ssthresh=8, per cui CWND < Ssthresh), e durante tale fase la CWND raddoppia ad ogni Round Trip Time (passando da 1, a 2, a 4 e a 8). Quindi la connessione passa in fase di congestion avoidance, e la crescita della CWND diventa lineare, incrementandosi di 1 MSS per ogni RTT. La CWND passa dunque a 9, ed infine mi serve trasmettere solo gli ultimi 2 segmenti.

Quindi il tempo totale impiegato è dato dalla seguente espressione:

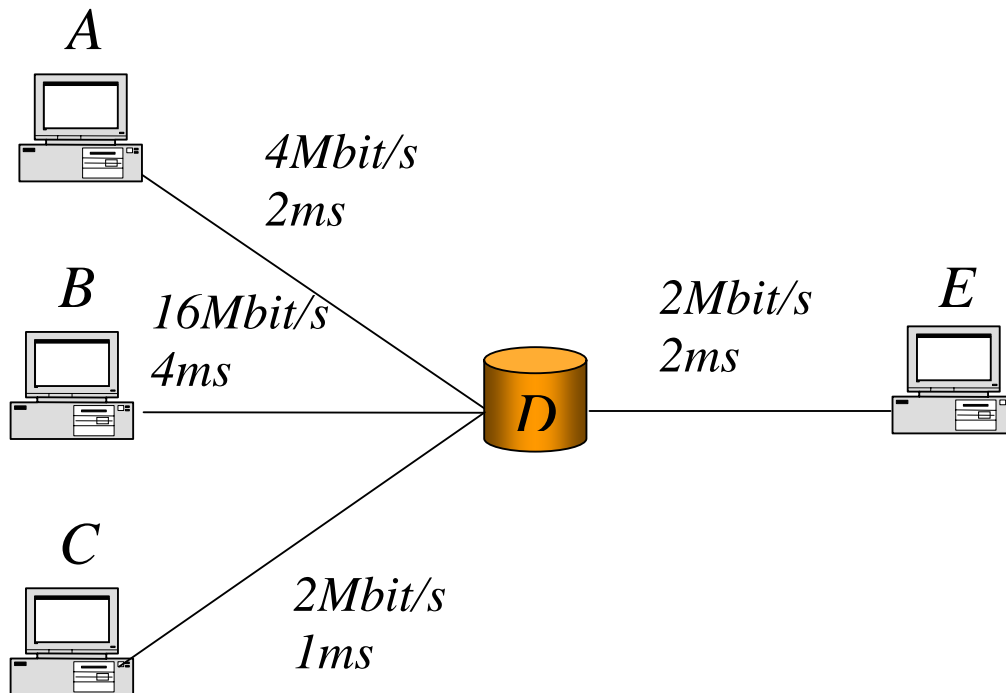
$$2\tau + (T + 2\tau) + (T + 2\tau) + (T + 2\tau) + (T + 2\tau) + (T + 2\tau) + (T + 2\tau) + (2-1)T = 10\text{ms} + 6 * (10.2\text{ms}) + 1 * 0.2\text{ms} = 10 + 61.2 + 0.2\text{ms} = 71.4\text{ms}$$

Esercizio 3

Nella rete in figura, A deve trasferire un pacchetto di 1000 byte verso E, B deve trasferire un pacchetto di 4000 byte verso E, C deve trasferire un pacchetto di 1000 byte anch'esso verso E. Tutti i trasferimenti incominciano nello stesso istante $t=0$. Sui collegamenti sono indicati capacità e ritardo di propagazione.

I nodi eseguono commutazione di pacchetto, e l'accodamento verso il nodo E avviene su base primo arrivato. Supponendo che i tempi di processing nei vari nodi siano trascurabili, si calcolino:

- 1) gli istanti di ricezione al nodo E del primo e dell'ultimo bit del pacchetto inviato da A
- 2) gli istanti di ricezione al nodo E del primo e dell'ultimo bit del pacchetto inviato da B
- 3) gli istanti di ricezione al nodo E del primo e dell'ultimo bit del pacchetto inviato da C



Soluzione:

Tempo Trasmissione di 1 pck da 8000 bit su link da 2 Mbit/s (link C->D e anche D->E)= $8000/(2 \cdot 10^6)=4\text{ms}$

Tempo Trasmissione di 1 pck da 8000 bit su link da 4 Mbit/s (link A->D)= $8000/(4 \cdot 10^6)=2\text{ms}$

Tempo Trasmissione di 1 pck da 32000 bit su link da 16 Mbit/s (link B->D)= $32000/(16 \cdot 10^6)=2\text{ms}$

Tempo Trasmissione di 1 pck da 32000 bit su link da 2 Mbit/s (link D->E)= $32000/(2 \cdot 10^6)=16\text{ms}$

Il pck proveniente da A (che chiameremo PA) arriva completamente al nodo D all'istante $2+2=4\text{ms}$ (ovvero: il suo ultimo bit arriva in questo istante al nodo D)

Il pck proveniente da B (che chiameremo PB) arriva completamente al nodo D all'istante $2+4=6\text{ms}$ (ovvero: il suo ultimo bit arriva in questo istante al nodo D)

Il pck proveniente da C (che chiameremo PC) arriva completamente al nodo D all'istante $4+1=5\text{ms}$ (ovvero: il suo ultimo bit arriva in questo istante al nodo D)

Quindi il nodo D trasmette questi 3 pacchetti verso la destinazione E nell'ordine in cui gli sono arrivati: prima PA, poi PC, infine PB. Tenendo conto dei tempi di trasmissione e di propagazione sul link D->E risulta che:

-PA arriva definitivamente alla destinazione E all'istante $4+4+2=10\text{ms}$ (infatti D comincia a trasmettere PA verso E all'istante 4ms, 4ms è il tempo di trasmissione di PA sul link D->E e il ritardo di propagazione di D->E è 2ms)

-PC arriva definitivamente alla destinazione E 4 ms dopo l'arrivo di PA (4ms è infatti il tempo di trasmissione di PC sul link D->E), e dunque all'istante $10+4=14\text{ms}$

-PB arriva definitivamente alla destinazione E 16 ms dopo l'arrivo di PC (16ms è infatti il tempo di trasmissione di PB sul link D->E), e dunque all'istante $14+16=30\text{ms}$

Quindi:

1) Il primo bit di PA arriva in E all'istante 6ms, l'ultimo bit all'istante 10ms

1) Il primo bit di PB arriva in E all'istante 14ms, l'ultimo bit all'istante 30ms

1) Il primo bit di PC arriva in E all'istante 10ms, l'ultimo bit all'istante 14ms

Domande:

1) Si indichi qual è il significato dell'acronimo ARP. Si illustri poi con chiarezza e precisione qual è lo scopo e qual è il funzionamento di tale protocollo.

2) Si illustrino con chiarezza e precisione i meccanismi di Slow Start, Congestion Avoidance, Fast Retransmit e Fast Recovery all'interno del protocollo TCP. Si indichi poi in quale situazione tipica questi due ultimi meccanismi (Fast Retransmit e Fast Recovery) non risultano efficaci.

Si vedano i lucidi delle lezioni.