



# Reti di calcolatori

Prova scritta del 26 luglio 2013  
(3° appello sessione estiva AA 2012/13)

## Istruzioni

Svolgere ciascun esercizio su un **foglio (non pagina) separato**, riportando nome, cognome e numero di matricola. Svolgere gli esercizi possibilmente con ordine, riportando e descrivendo la procedura seguita in modo da consentire, durante la correzione, di distinguere errori concettuali da errori di distrazione e veniali.

Chiarimenti sulle correzioni potranno essere chiesti (anche per gli esami insufficienti) martedì 30 e mercoledì 31 luglio nello studio del docente prima e durante gli esami orali. Uno scritto insufficiente non consente di completare l'esame con l'orale; eventuali prove "al limite" verranno segnalate come "18-".

## Segnare la preferenza per l'orale Orale

- lunedì 29 pomeriggio  (i posti sono pochi)
- martedì 30
- mercoledì 31

## In questo appello non si può fare l'orale in altra data.

Entro lunedì 29 luglio alle 14.00 verranno pubblicati gli esiti dello scritto con la scaletta del colloquio orale. La mancata presenza all'orale implica non passare l'esame e dover rifare anche lo scritto, a meno di giustificati motivi comunicati in anticipo via mail.

## Esercizio 1 - Domande (e risposte!) brevi (11 punti)

1. Qual'è la più semplice topologia che ammette una ridondanza di percorsi? Perché? Quanti collegamenti mono-direzionali ha in funzione del numero di nodi collegati  $N$ .
2. Si consideri un protocollo Data Link per connessioni punto-punto che usa "stop-and-wait" come funzione di controllo della trasmissione. Data una capacità  $C$  Mbit/s ed una PDU di dimensione  $N$  byte, si calcoli il throughput  $S$  del protocollo in funzione del tempo di propagazione  $t_p$ .
3. Quali livelli protocollari trovo in un router di Internet?
4. In un sistema Client/Server, cosa caratterizza il processo client e cosa il processo server?
5. Dato l'URL  
`ftp://www.myFileSource.org/pict/landscapes/Mountains/Bondone001.jpg`  
si spieghino le diverse parti dell'identificativo, il loro ruolo e se sono o meno case-sensitive.
6. A livello Data-Link è necessario formare le trame (frame) di trasmissione. Si spieghi cos'è la funzione di "framing" e si faccia un esempio di una tecnica usata.

## Esercizio 2 (11 punti)

Si consideri il protocollo di trasporto TCP.

1. Si spieghi come viene calcolato il timer di ritrasmissione RTO (Retransmission Time-Out)
2. Cos'è il "backoff" esponenziale? Quando viene usato?
3. Dopo un evento di perdita multipla dello stesso pacchetto, quando viene ricalcolato per la prima volta il timer di ritrasmissione?
4. Cosa sono e come funzionano gli algoritmi noti come Fast Retransmit e Fast Recovery?

Si consideri ora una connessione che trasferisce 100 segmenti (ad esempio il trasferimento di un file di 100000 byte con MSS 1000 byte) numerati da 1 a 100. Si assuma un Round Trip Time RTT approssimativamente costante, dominato dal ritardo di propagazione, tale per cui TCP calcola un SRTT = 170ms e una varianza RTTVAR = 20ms. La rete su cui avviene la trasmissione non è mai un collo di bottiglia, ed ha una velocità di trasmissione tale per cui il ritardo di trasmissione dei pacchetti è trascurabile. La finestra del ricevitore è posta a 64 kbyte. Il router di accesso alla rete, tuttavia, è difettoso, e scarta in modo sistematico su ciascuna delle sue interfacce d'ingresso, un pacchetto ogni 20. Si supponga, per semplicità, che non ci sono altre connessioni coinvolte e che il "conteggio" per decidere quali pacchetti verranno scartati parta dopo l'apertura della connessione TCP.

5. Calcolare quanti byte verranno trasmessi in rete comprese le ritrasmissioni e gli headers dei protocolli TCP, IP ed Ethernet, supponendo che questo sia il livello Data-Link utilizzato (se non si ricordano le dimensioni dell'header Ethernet lo si approssimi partendo dal fatto che deve contenere almeno gli indirizzi MAC e un campo di controllo).
6. Si calcoli il tempo di trasferimento del file (senza considerare apertura a chiusura della connessione):
  - a) nel caso in cui TCP usi solamente il timeout di ritrasmissione;
  - b) nel caso in cui TCP usa anche Fast Retransmit, ma non Fast Recovery;
  - c) nel caso in cui TCP usa sia Fast Retransmit che Fast Recovery.
7. Commentare brevemente i risultati del punto 6.
8. Si provi a generalizzare il risultato ottenuto come una funzione di RTT.

### Esercizio 3 (11 punti)

Si consideri la rete disegnata in figura 1. Tutti i router usano OSPF come protocollo di calcolo delle rotte per l'instradamento dei pacchetti. Il costo dei link non è simmetrico.

1. Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento dei nodi A ed E e si disegni il Minimum Spanning Tree corrispondente con radice in A ed E rispettivamente.
2. Quanti pacchetti transitano nella rete per la distribuzione in flooding del costo dei link del nodo C?

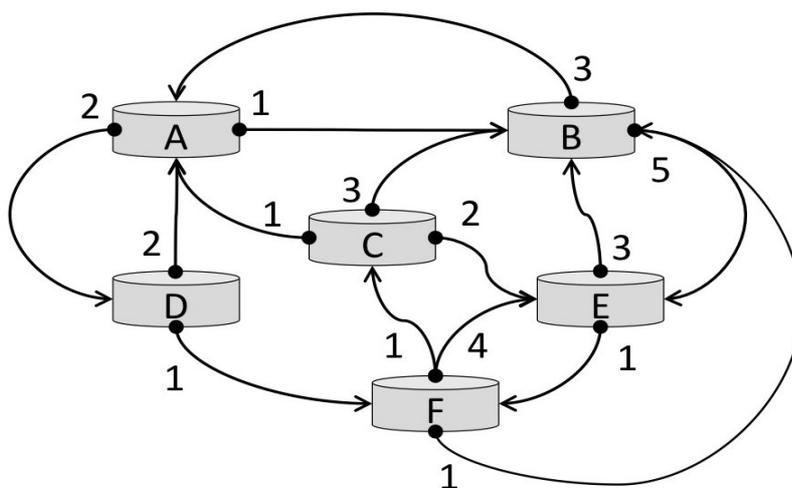


Figura 1. Topologia logica della rete IP con il costo dei link asimmetrici. Il pallino indica il nodo di partenza e la freccia quello di destinazione, il costo è vicino al nodo di partenza.

## Alcune indicazioni per soluzioni "corrette"

### Es. 1

Cercate di essere ragionevolmente precisi e sul tema, spesso ci sono tante parole ma che girano intorno alla risposta.

Se una domanda contiene formule e/o unità di misura cercate almeno di fare qualche "sanity check", evitare cioè risposte che sono assurde.

Ad esempio nel punto 2) rispondere

- $Thr = C \cdot N$
- $Thr = N \cdot Tp$

o similari sono palesemente sbagliate: il throughput o è un numero puro riferito ad una capacità oppure è misurato in bit/s ... entrambe le formule sopra non lo sono. Per di più la seconda porta alla conclusione assurda che aumentando il tempo di propagazione sul link aumenta il throughput! La risposta corretta è semplicemente  $Thr = N/RTT$ ;  $RTT = 2Tp + N \cdot 8/C$  dove  $N \cdot 8/C$  è il tempo di trasmissione del pacchetto, assumendo che l'ACK abbia un tempo di trasmissione trascurabile. Poi si possono anche mettere a posto gli ordini di grandezza (milli, Mega, kilo, ...) per avere una risposta coerente.

### Es. 2

Se l'esercizio chiede espressamente il "calcolo", riportare solo il risultato (MST) non è considerato soddisfacente.

### Es. 3

L'esercizio inizia con "Si consideri il protocollo di trasporto TCP. " Evidentemente tutte le domande si riferiscono a questo protocollo. "Tirare in ballo" Ethernet, Aloha o qualsiasi altra cosa è evidentemente sbagliato. TCP ha un suo algoritmo di backoff esponenziale, che viene usato in caso di perdite multiple dello stesso pacchetto.

Per quanto riguarda il punto 3) si tratta di un argomento discusso a lezione in relazione all'ultima domanda (retorica) della slide 70 sul livello di trasporto. Se TCP va in backoff esponenziale, RTO non viene più "calcolato", ma è definito dal valore di backoff. Un nuovo valore di RTO viene calcolato con la formula standard  $RTO = SRTT + 4 \cdot RTT_{VAR}$  solamente quando è disponibile un nuovo valore affidabile di RTT, ovvero quando viene ricevuto l'ACK relativo ad un segmento "nuovo", ovvero un segmento che non era nella finestra di trasmissione quanto è scaduto il primo timeout.

Il punto 6) era complesso, e non per caso, almeno una domanda "difficile" per dare volti alti mi pare indispensabile.

Data la complessità di TCP il modo per dare una risposta corretta senza "rischi" è seguire tutta la procedura di scambio dei segmenti. Ovviamente richiede tempo, anche se in un'oretta si fa senza problemi. In alternativa si può provare a seguire solamente l'andamento della finestra, suddividendo le fasi di comunicazioni in cicli determinati da RTT oppure dallo scadere di un timeout. Indichiamo con  $n(m)$  l'invio dell' $n$ -esimo segmento, con  $m$  che identifica il ciclo di "scarto" del router, modulo 20. Lo schema che segue implica 1 riga per ciascun ciclo e "risolve" 6.a).  
 $SSTHR = 32$  segmenti ... ma porlo a 64 non cambia nulla.

RTT(1): 1(1)  
RTT(2): 2(2),3(3)  
RTT(3): 4(4),5(5),6(6),7(7)  
RTT(4): 8(8),9(9),10(10),11(11),12(12),13(13),14(14),15(15)  
RTT(5): 16(16),17(17),18(18),19(19),20(0)<<<PERSO,(21)(1),22(2),23(3),24(4),25(5),26(6),27(7),28(8),29(9),30(10),31(11)

RTO(1): si attende lo scadere del timeout, non avendo FR-FR gli ACK duplicati non contano

RTT(6): 20(12) - Pacchetto ritrasmesso, si attende l'ACK, che risconterà tutto (ACK cumulativi) fino al segmento 31 incluso. La finestra viene posta a 1 segmento SSTHR a 16 segmenti e si riparte in Slow-Start. Il router scarta pacchetti, non segmenti, quindi il calcolo mod. 20 riparte subito. Si può anche ripartire da finestra 2.

RTT(7): 32(13)  
RTT(8): 33(14),34(15)  
RTT(9): 35(16),36(17),37(18),38(19)  
RTT(10): 39(0)<<<PERSO, 40(1),41(2),42(3),43(4),44(5),45(6),46(7)

RTO(2):

RTT(11): 39(8); SSTHR=8, l'ACK riscontra 46 incluso  
RTT(12): 47(9)  
RTT(13): 48(10),49(11)  
RTT(14): 50(12),51(13),52(14),53(15)  
RTT(15): 54(16),55(17),56(18),57(19),58(0)<<<PERSO,59(1),60(2),61(3)

RTO(3):

RTT(16): 58(4); SSTHR=4, l'ACK riscontra 61 incluso  
RTT(17): 62(5)  
RTT(18): 63(6),64(7)  
RTT(19): 65(8),66(8),67(9),68(10); Si entra in Congestion Avoidance  
RTT(20): 69(11),70(12),71(13),72(14),73(15)  
RTT(21): 74(16),75(17),76(18),77(19),78(0)<<<PERSO,79(1)

RTO(4):

RTT(22): 78(2); SSTHR=3, l'ACK riscontra 79 incluso  
RTT(23): 80(3)  
RTT(24): 81(4),82(5)  
RTT(25): 83(6),84(7),85(8); Si entra in congestion avoidance  
RTT(26): 86(9),87(10),88(11),89(12)  
RTT(27): 90(13),91(14),92(15),93(16),94(17)  
RTT(28): 95(18),96(19),97(0)<<<PERSO,98(1),99(2),100(3)

RTO(5):

RTT(29): 97(4); l'ACK riscontra la fine del file e la connessione si può chiudere

Tempo di trasferimento = 29 RTT + 5 RTO = (29\*170 + 5\*250)ms = 6,18s

6.b) e 6.c) sono analoghi; 6.c) complicato dalla trasmissione "sostenuta" del Fast Recovery.

Da notare come con perdite casuali l'impatto di FR e FR-FR sia minore che nel caso di perdite da congestione e anche che è impossibile trovare una relazione semplice in funzione di RTT, perché le perdite casuali non consentono di trovare una regolarità.