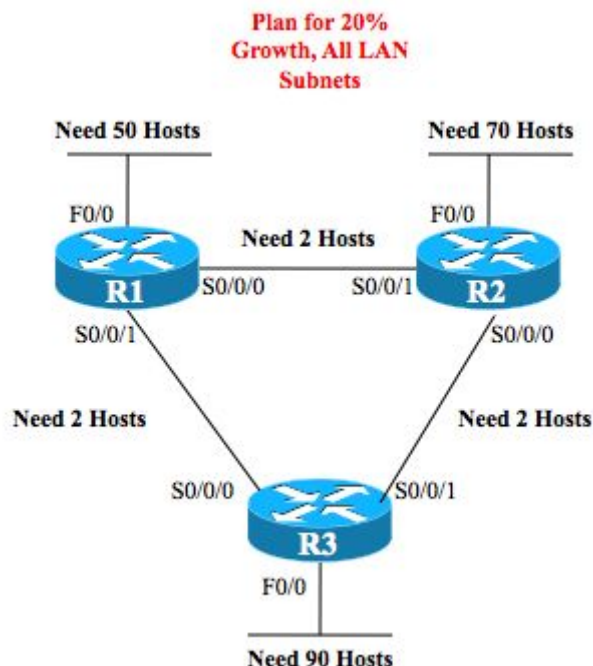


# **Esame Reti 31 Ottobre 2017**

## **Parte A**

- **Subnetting FLSM e VLSM con crescita**
  - **Tabelle routing**
- **IDLE RQ (eff. e valore ottimale fin.)**
  - **Timeout TCP e Jacobson**
  - **Delayed Ack**

**Es 1) Considerate la rete in figura, subnettata a livello 3. Partite con un netid privato classe B. Tenete conto delle esigenze di crescita riportate in figura.**



**Domanda 1 (8 punti) Nel caso di utilizzo di maschera di sottorete a lunghezza fissa (FLSM), specificate:**

- 1. la subnet mask**
- 2. la subnet id delle 3 sottoreti e gli intervalli di host address corrispondenti**
- 3. Gli indirizzi delle interface di routing e la tabella d'instradamento del router R1.**

**Domanda 2 (8 punti) Nel caso di utilizzo di maschera di sottorete a lunghezza variabile (VLSM), specificate: 1. le tre subnet mask**

- 2. il rapporto tra indirizzi IP effettivamente assegnati ad host e indirizzi IP allocati a ciascuna sottorete.**

Domanda 1)

Ipotesi: utilizzo l'ip di classe B 172.25.0.0/16

Ipotesi: ignoro la RFC sul non utilizzo di net ID con tutti zeri e tutti uni

1) La rete che ha bisogno di più host è quella che parte da R3, che considerando il piano di crescita, necessita di 108 host.

Il minimo numero di bit per coprire questo fabbisogno di host è di 7 bit. Riservando quindi 7 bit per gli host, posso utilizzare una come maschera /25, e cioè 255.255.255.128.

2) Subnetting reti per gli host

RETE	N. HOST	SUBNET ID	HOST RANGE	BROADCAST IP
net1	50 (+20%) = 60	172.25.0.0	172.25.0.1-126	172.25.0.127
net2	70 (+20%) = 84	172.25.0.128	172.25.0.129-254	172.25.0.255
net3	90 (+20%) = 108	172.25.1.0	172.25.1.1-126	172.25.1.127

Considerando però anche reti punto punto, avremo

RETE	N. HOST	SUBNET ID	HOST RANGE	BROADCAST IP
------	---------	-----------	------------	--------------

net4	2	172.25.1.128	172.25.1.129-254	172.25.1.255
net5	2	172.25.2.0	172.25.2.1-126	172.25.2.127
net6	2	172.25.2.128	172.25.2.129-254	172.25.2.255

3) Le interfacce di routing avranno degli indirizzi compresi nel range riservato agli host, e idealmente potrebbero essere i seguenti:

R1

- S0/0/0 → net4 → 172.25.1.129
- S0/0/1 → net5 → 172.25.2.1

R2

- S0/0/0 → net6 → 172.25.2.129
- S0/0/1 → net4 → 172.25.1.130

R3

- S0/0/0 → net5 → 172.25.2.2
- S0/0/1 → net6 → 172.25.2.130

Il router R1 avrà una tabella di routing strutturata come segue:

Destination	Gateway	Genmask	Flag	MSS	Iface
172.25.0.0	0.0.0.0	255.255.255.128	U	40	F0/0
172.25.0.128	172.25.1.130	255.255.255.128	UG	40	S0/0/0
172.25.1.0	172.25.2.2	255.255.255.128	UG	40	S0/0/1
172.25.1.128					
→ 172.25.1.129	0.0.0.0	255.255.255.255	U	40	LO
→ 172.25.1.128	172.25.1.130	255.255.255.128	UG	40	S0/0/0
172.25.2.0					
→ 172.25.2.1	0.0.0.0	255.255.255.255	U	40	LO
→ 172.25.2.0	172.25.2.2	255.255.255.128	UG	40	S0/0/1
172.25.2.128	172.25.1.130	255.255.255.128	UG	40	S0/0/0

Non specifico un default gateway, in quanto la rete è completamente conosciuta, e non possono essere inviati pacchetti a rotte non conosciute (se si provasse, ICMP provvederebbe a fornire il messaggio di errore apposito).

Domanda 2)

Se si effettuasse un subnetting con VLSM otterremmo quanto segue

RETE	N. HOST	MASK	SUBNET ID	HOST RANGE	BROADCAST IP
net3	90 (+20%) = 108	/25	172.25.0.0	172.25.0.1-126	172.25.0.127
net2	70 (+20%) = 84	/25	172.25.0.128	172.25.0.129-254	172.25.0.255
net1	50 (+20%) = 60	/26	172.25.1.0	172.25.1.1-62	172.25.1.63

net4	2	/30	172.25.1.64	172.25.1.65-66	172.25.1.67
net5	2	/30	172.25.1.68	172.25.1.69-70	172.25.1.71
net6	2	/30	172.25.1.72	172.25.1.73-74	172.25.1.75

Da questi vediamo che:

- net3 avrebbe 90-108 host assegnati su 126 possibili (uguale a FLSM)
- net2 avrebbe 70-84 host assegnati su 126 possibili (uguale a FLSM)
- net1 avrebbe 50-60 host assegnati su 62 possibili (contro FLSM che ne avrebbe 50-60 su 126)
- net4 avrebbe 2 host assegnati 2 (contro FLSM che ne avrebbe 2 su 126)
- net5 avrebbe 2 host assegnati 2 (contro FLSM che ne avrebbe 2 su 126)
- net6 avrebbe 2 host assegnati 2 (contro FLSM che ne avrebbe 2 su 126)

Ciò denota quanto VLSM permetta di ottimizzare l'assegnamento degli IP rispetto a FLSM, soprattutto in casi dove la rete prevede una o più connessioni punto-punto (FLSM riserva molti IP per ogni sottorete punto-punto).

**Esercizio 2 (5 punti)** Supponendo che la PDU di un protocollo IDLE RQ sia 1000 bit, qual è il tasso di utilizzo di un collegamento punto-punto a 150 Mbps sulla distanza di 7000 km? Qual è il valore ottimale della finestra?

DimPDU = 1000 bit

BitRate = 150'000'000 bps

Dist = 7'000'000 m

Velocità Propagazione su rame VelPr =  $2 \cdot 10^8$  m/s

L'efficienza (o tasso di utilizzo del collegamento) di Idle-RQ si calcola come segue:

$$1 / (1 + 2 \cdot T_p / T_{ix}) = 1 / (1 + 2 \cdot (\text{Dist}/\text{VelPr}) / (\text{DimPDU}/\text{BitRate}))$$

E quindi

$$1 / (1 + 2 \cdot (7000000/(2 \cdot 10^8)) / (1000 / 150000000)) = 0.000095229$$

Dunque la percentuale di utilizzo del canale è di ~0.0095%

Considerando la struttura di Idle RQ, la finestra è sempre uguale a 1 (Idle RQ prevede sempre l'invio di un singolo PDU e dell'attesa di un ack / scadenza timeout prima di trasmettere un altro PDU).

Tuttavia, applicando Continuous RQ allo scenario richiesto, la finestra ottimale consiste nel quantitativo di PDU inviabili prima di ricevere il primo acknowledgement

Ed esso si calcola come segue:

$$(T_{ix} + 2 \cdot T_p) / T_{ix} = ((\text{DimPDU}/\text{BitRate}) + 2 \cdot (\text{Dist}/\text{VelPr})) / (\text{DimPDU}/\text{BitRate})$$

$$\text{E quindi: } ((1000 / 150000000) + 2 \cdot (7000000/(2 \cdot 10^8))) / (1000 / 150000000) = 10501 \text{ PDU}$$

Dunque la finestra sarà di 10501000 bit, che in byte sono 1312625.

### Esercizio 3 (9 punti)

- **Domanda 1 (3 punti)** Sapendo che il valore iniziale di Round Trip Sampled per una connessione TCP e' RTS = 45 ms, e che gli ACK successivi arrivano con ritardi di 15, 24 e 33 ms, calcolate i valori del timer di ritrasmissione di TCP. Fate le vostre ipotesi sui valori dei parametri.
- **Domanda 2 (3 punti)** Ripetete il calcolo usando l'algoritmo di Jakobson e con i pesi  $g = 2/3$ ,  $h = 1/3$ . Fate tutte le ipotesi necessarie.

Domanda 1)

In questo caso la domanda è ambigua, e sarebbe da chiedere al prof in sede d'esame se il ritardo per gli ack successivi si riferisce al tempo intercorso tra l'invio del PDU relativo e il loro arrivo (e quindi ad esempio RTS = 15 per il primo ACK successivo ecc.) - caso 1- oppure il ritardo si riferisce al RTS, indicando i ms in più che ci mette a tornare l'ack rispetto al RTS (e quindi ad esempio RTS = 60 nel primo ACK successivo ecc.) - caso 2 -.

Qua svolgo entrambe le casistiche:

Ipotesi: pongo il primo EstimatedRTT = primo RTS (45ms) non avendo valori precedenti di EstimatedRTT

Ipotesi: utilizzo il valore raccomandato 2 come fattore di varianza (DelayVarianceFactor)

Caso 1: Il ritardo viene considerato come SampleRTT

- Primo ACK:  
$$\text{EstimatedRTT} = (1-x) * \text{EstimatedRTT} + x * \text{SampleRTT} = (0.9 * 45) + (0.1 * 15) = 42$$
$$\text{Timeout} = \text{EstimatedRTT} * \text{DelayVarianceFactor} = 42 * 2 = 84 \text{ ms}$$
- Secondo ACK:  
$$\text{EstimatedRTT} = (0.9 * 42) + (0.1 * 24) = 40.2$$
$$\text{Timeout} = 40.2 * 2 = 80.4 \text{ ms}$$
- Terzo ACK:  
$$\text{EstimatedRTT} = (0.9 * 40.2) + (0.1 * 33) = 39.48$$
$$\text{Timeout} = 39.48 * 2 = 78.96 \text{ ms}$$

Caso 2: Il ritardo viene sommato agli EstimatedRTT per calcolare il SampleRTT

- Primo ACK:  
$$\text{EstimatedRTT} = (0.9 * 45) + (0.1 * (45+15)) = 46.5$$
$$\text{Timeout} = 46.5 * 2 = 93 \text{ ms}$$
- Secondo ACK:  
$$\text{EstimatedRTT} = (0.9 * 46.5) + (0.1 * (46.5 + 24)) = 48.9$$
$$\text{Timeout} = 48.9 * 2 = 97.8 \text{ ms}$$
- Terzo ACK:  
$$\text{EstimatedRTT} = (0.9 * 48.9) + (0.1 * (48.9 + 33)) = 52.2$$
$$\text{Timeout} = 52.2 * 2 = 104.4 \text{ ms}$$

Domanda 2) Applico Jakobson (per far prima lo applico solo al caso 1)

Jakobson:

**nb:  $x = g$**

$$\text{EstimatedRTT} = (1-g) * \text{EstimatedRTT} + (g) * \text{SampleRTT}$$

$$\text{Error} = |\text{SampleRTT} - \text{EstimatedRTT}|$$

$$\text{Deviation} = \text{Deviation} + h * (\text{Error} - \text{Deviation})$$

$$\text{Timeout} = \text{EstimatedRTT} + 4 * \text{Deviation}$$

Ipotesi: stabilisco un valore di deviazione iniziale in base al valore di Timeout iniziale consigliato dalla RFC 6298 di 1 sec:

- $\text{Timeout} = \text{EstimatedRTT} + 4 * \text{Deviation} \rightarrow 1000 = 45 + 4 * \text{Deviation} \rightarrow \text{Deviation} = 238$

- Primo ACK:

$$\text{EstimatedRTT} = (0.3 * 45) + (0.6 * 15) = 22.5$$

$$\text{Deviation} = (0.6 * 238) + (0.3 * |22.5 - 15|) = 145.05$$

$$\text{Timeout} = 22.5 + 4 * 145.05 = 602.7 \text{ ms}$$

- Secondo ACK

$$\text{EstimatedRTT} = (0.3 * 22.5) + (0.6 * 24) = 21.15$$

$$\text{Deviation} = (0.6 * 145.05) + (0.3 * |21.15 - 24|) = 87.89$$

$$\text{Timeout} = 21.15 + 4 * 87.89 = 372.71 \text{ ms}$$

- Terzo ACK

$$\text{EstimatedRTT} = (0.3 * 21.15) + (0.6 * 33) = 26.15$$

$$\text{Deviation} = (0.6 * 87.89) + (0.3 * |26.15 - 33|) = 54.79$$

$$\text{Timeout} = 26.15 + 4 * 54.79 = 245.21 \text{ ms}$$

**Domanda 3 (3 punti)** Un'applicazione lato server riceve un parametro da un dispositivo via TCP, lo usa per eseguire un lookup su disco che richiede 50 msec e poi invia la risposta al mittente, che manda un altro parametro, e così via. Lo sviluppatore intende eseguire un controllo per disabilitare il meccanismo "delayed ack" di TCP. E' una buona idea? La vostra risposta cambierebbe se la lookup table fosse in memoria centrale? Perché sì o perché no?

Domanda 3)

Disattivare il delayed potrebbe non essere una buona idea, in quanto lo scopo di questo è quello di ritardare la trasmissione degli ack di 200ms (o 500ms, fonti diverse danno valori diversi) in modo che questo possa essere trasmesso in piggyback con una trasmissione dati. In questo caso, se venisse disattivato, l'applicativo server restituirebbe subito un ack all'applicativo client, mentre al termine del lookup dovrebbe inviare un ulteriore segmento con i dati di risposta, al quale il client dovrebbe rispondere con un ack a sua volta. Questo porta alla trasmissione di 4 PDU, mentre con il delayed ack, il server attenderebbe un quantitativo di ms indicativo che permetterebbe di accorpate in piggyback l'ack al PDU dati di risposta, limitando il numero di segmenti scambiati a 2 (con conseguente minor occupazione di banda) (a sua volta, il client se sufficientemente veloce, trasmetterebbe l'ack per la risposta del server in un PDU dati).

In caso il lookup fosse eseguito in memoria centrale la situazione cambierebbe, perché sicuramente esso richiederebbe un tempo molto minore rispetto al lookup su disco, addirittura trascurabile. In questo caso, il delayed ack risulterebbe inserirebbe una inutile attesa nella trasmissione, facendo rimanere i dispositivi in attesa di dati che molto probabilmente non arriveranno entro la scadenza del delay.