

Quesito 1. Cinque processi “a lotti”, identificati dalle lettere $A, B, \dots, E$, arrivano all’elaboratore rispettivamente agli istanti 0, 3, 5, 9, 11. Tali processi hanno un tempo di esecuzione atteso di 6, 2, 6, 3 e 4 unità di tempo rispettivamente. Per ognuna delle politiche di ordinamento sotto indicate, si determini il tempo medio di: risposta, turnaround ed attesa, trascurando i ritardi dovuti allo scambio di contesto. Ove la politica di ordinamento in esame consideri la priorità, i relativi valori, mantenuti staticamente per l’intera durata dell’esecuzione, siano rispettivamente: 1, 5, 3, 4 e 2 (con 5 valore di priorità massima).

<i>First Come First Served (FCFS)</i>	un processo per volta, fino al completamento
<i>Round Robin</i>	a divisione di tempo, con quanto di ampiezza 2, <u>senza</u> priorità
<i>Round Robin</i>	a divisione di tempo, con quanto di ampiezza 2, <u>con</u> priorità e prerilascio
<i>Shortest Job First (SJF)</i>	<u>senza</u> priorità esplicita, <u>con</u> prerilascio

Nell’applicare le politiche sopra indicate si dia sempre precedenza ai processi già attivati rispetto a quelli appena arrivati.

Quesito 2. Un progettista di sistema operativo desidera confrontare le prestazioni di 2 tecniche di gestione dei blocchi liberi, applicandole ad un disco fisso di ampiezza 20 GB, con blocchi ampi 512 B. Le due tecniche in questione sono: (1) a “mappa di *bit*” e (2) a lista concatenata. Il progettista ha fissato i seguenti criteri di confronto:

- costo spaziale delle rispettive strutture dati
- costo temporale di accesso al 120° blocco libero, a fronte di una occupazione di disco del 50% e massimamente sparsa (ossia con alternanza tra blocchi liberi ed occupati)

ed intende applicarli al caso di un elaboratore dotato di istruzioni macchina capaci di operare su singoli *bit* all’interno di parole di 32 *bit*, tutte al costo unitario di 1 ns. (nanosecondi) alla frequenza di *clock* fissata per il sistema. Tali istruzioni consentono di: (a) rilevare la 1^a posizione non ‘0’ entro una singola parola; (b) localizzare la posizione del primo ‘1’ successivo ad una posizione data entro una singola parola; (c) determinare il numero di occorrenze di ‘1’ in una singola parola. Il costo di accesso ad un singolo blocco di disco (comprensivo dei tempi di *seek*, *rotation* e *transfer*) è pari a 10 ms. (millisecondi).

Quesito 3. Dato il sistema descritto dalla seguente rappresentazione insiemistica delle assegnazione delle risorse:

$$\begin{aligned}
 P &= \{P_1, P_2, P_3, P_4, P_5\} \\
 R &= \{R_1^2, R_2^2, R_3^1, R_4^1\} \\
 E &= \{P_2 \rightarrow R_2, P_3 \rightarrow R_4, P_5 \rightarrow R_1, P_1 \rightarrow R_3, P_4 \rightarrow R_3, \\
 &\quad R_1 \rightarrow P_1, R_1 \rightarrow P_2, R_2 \rightarrow P_3, R_2 \rightarrow P_4, R_3 \rightarrow P_2, R_4 \rightarrow P_5\}
 \end{aligned}$$

si rappresenti il corrispondente grafo di allocazione delle risorse, verificando se il sistema si trovi attualmente in condizione di stallo. Successivamente, si verifichi se e come la situazione cambi prevedendo il rilascio della risorsa R_1 da parte del processo P_1 .

Quesito 4. In una rete *Internet*, un nodo A invia al nodo B un messaggio TCP con campo dati ampio 900 B ed intestazione (*header*) ampia 20 B . In effetti, il nodo mittente inoltrerà tale messaggio al *router* R_1 , che lo trasferirà al *router* R_2 , il quale infine lo consegnerà al nodo destinazione. Come sappiamo, i *datagram* di livello IP hanno intestazione ampia 20 B e dimensione totale massima 2^{16} B , la quale, in presenza di frammentazione, viene ripartita su frammenti di dimensione elementare 8 B o multipli interi di essa, ossia fino ad un massimo di 2^{13} frammenti. Le linee di collegamento tra i rami del percorso $A - R_1 - R_2 - B$ prevedono trame delle seguenti dimensioni (trascurando gli eventuali *trailer* ed i *flag* di separazione tra trame):

ramo	header + campo dati
$A - R_1$	$(14 + 1014) B$
$R_1 - R_2$	$(8 + 504) B$
$R_2 - B$	$(12 + 500) B$

Noti questi dati, vogliamo determinare i valori dei seguenti campi *header* di ciascun *datagram* od eventuale frammento di esso inviato su ogni ramo del percorso dato:

campo	annotazioni
<i>TL (Total Length)</i>	ampiezza totale dell'invio, incluso l' <i>header</i>
<i>DF (Don't Fragment)</i>	vale 0 se l'invio non prevede frammentazione
<i>MF (More Fragments)</i>	vale 1 per tutti i frammenti di un <i>datagram</i> frammentato tranne l'ultimo
<i>FO (Fragmentation Offset)</i>	indica la posizione relativa del frammento nel <i>datagram</i>
<i>Identification</i>	tramite il quale il nodo destinazione determina a quale <i>datagram</i> appartenga il frammento

Quesito 5. Lo schema logico riportato in figura 1 rappresenta la rete dati di una piccola Azienda composta da due reparti operativi ed una stanza per i gestori della rete.

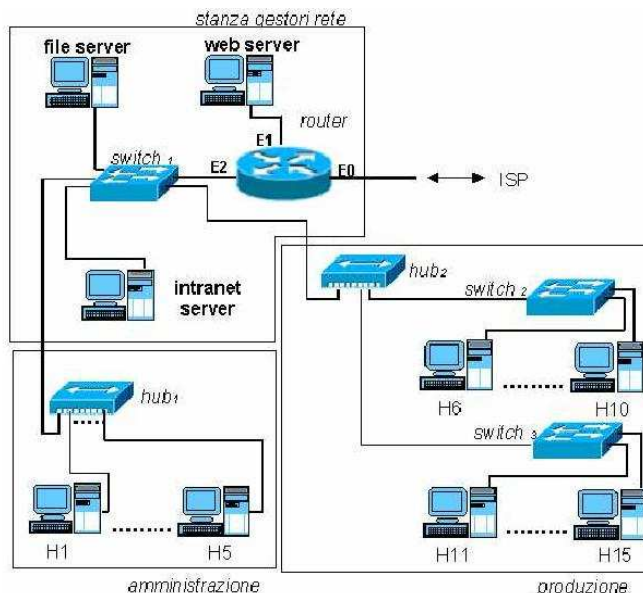


Figura 1: Schema logico della rete dati dell'Azienda.

Nella sua struttura logica sono presenti tre server, posizionati nella stanza gestori rete:

- quello denominato *web server* offre servizi *Web* accessibili prevalentemente da utenti esterni all'Azienda
- quello denominato *intranet server* offre servizi di *application server* e di accreditamento, prevalentemente al solo reparto amministrazione
- quello denominato *file server* offre servizi di condivisione di *file* a tutta l'Azienda, e gli utenti sono suddivisi tra:
 - un reparto amministrazione composto da 5 postazioni di lavoro, caratterizzate da un traffico al 50% di tipo utente-server e facente capo al file server e, per il rimanente 50%, diretto verso l'intranet server
 - un reparto produzione composto da 10 postazioni di lavoro, caratterizzate da utenti con traffico di tipo utente-server facente capo al file server.

Tutti i dispositivi di rete operano a 100 Mbps. L'Azienda accede ad *Internet* mediante un unico indirizzo IP statico fornito direttamente dall'ISP. Al suo interno, invece, decide di condividere gli indirizzi privati di una intera classe C (192.168.1.0), sfruttando la funzione di traduzione degli indirizzi (NAT) realizzata all'interno del router.

Proporre una suddivisione degli indirizzi utili in sottoreti della stessa dimensione (pertanto dotati della medesima *subnet mask*), la più grande possibile, e compilare una tabella che contenga, per tutti i dispositivi di rete, l'indirizzo IP associato, la *subnet mask* ed il *default gateway*.

Quesito 6. È nota la funzione svolta dal cosiddetto "*vacation daemon*", il quale, una volta installato ed attivato presso un utente, invia una notifica di assenza ai mittenti di eventuali messaggi. Si spieghi in che modo la realizzazione di tale servizio impedisca l'innescio di un ciclo infinito di risposte automatiche ove qualche mittente abbia installato una propria copia di esso immediatamente dopo aver inviato un messaggio genuino al nodo incriminato.

Soluzione 1 (punti 5).

- FCFS, un processo alla volta, fino al completamento

processo A AAAAAA LEGENDA DEI SIMBOLI
 processo B ---bbbBB - non ancora arrivato
 processo C -----cccCCCCC x (minuscolo) attesa
 processo D -----dddddDDD X (maiuscolo) esecuzione
 processo E -----eeeeeeEEEE

CPU AAAAAABCCCCCDDDEEEE
 coda ...bbbcc.dddddeee....
 C.....eee.....

processo	risposta	tempo di	
		attesa	turnaround
A	0	0	6
B	3	3	$3+2=5$
C	3	3	$3+6=9$
D	5	5	$5+3=8$
E	6	6	$6+4=10$
medie	3,40	3,40	7,60

- Round Robin, con quanto di ampiezza 2, senza priorità

processo A AAAAaaAA LEGENDA DEI SIMBOLI
 processo B ---bBB - non ancora arrivato
 processo C -----cccCCccCCccCC x (minuscolo) attesa
 processo D -----dDDdddD X (maiuscolo) esecuzione
 processo E -----eeeEEeeeEE . coda vuota

CPU AAAABBAACDDCCEEDCCEE
 coda ...baacc.dcceeddcee..
 C.....eddcce....

processo	risposta	tempo di	
		attesa	turnaround
A	0	2	8
B	1	1	$1+2=3$
C	3	$3+2+3=8$	$8+6=14$
D	1	$1+4=5$	$5+3=8$
E	3	$3+3=6$	$6+4=10$
medie	1,60	4,40	8,60

- Round Robin, con quanto di ampiezza 2, con priorità e prerilascio

processo A AAAaaaaaaaaaaaaaaaaAA LEGENDA DEI SIMBOLI
 processo B ---BB - non ancora arrivato
 processo C -----CCCCcccccC x (minuscolo) attesa
 processo D -----DDD X (maiuscolo) esecuzione
 processo E -----eeeEEEE . coda vuota

CPU AAABCCCCDDDCCEEEEA
 coda ...aaaaaacceeeaaa..
 aaeee.....
 a.....

processo	risposta	tempo di	
		attesa	turnaround
A	0	15	$15 + 6 = 21$
B	0	0	2
C	0	3	$3 + 6 = 9$
D	0	0	3
E	3	3	$3 + 4 = 7$
medie	0,60	4,20	8,40

- SJF, senza priorità esplicita, con prerilascio

processo A AAAaAAAA LEGENDA DEI SIMBOLI
processo B ---BB - non ancora arrivato
processo C -----cccCcccccccCCCC x (minuscolo) attesa
processo D -----DDD X (maiuscolo) esecuzione
processo E -----eEEEE

CPU AAABBAACDDDEEECCCCC
coda ...aaccC.ccecccc.....
 C.....

processo	risposta	tempo di	
		attesa	turnaround
A	0	2	$2 + 6 = 8$
B	0	0	2
C	3	10	$10 + 6 = 16$
D	0	0	3
E	1	1	$1 + 4 = 5$
medie	0,80	2,60	6,80

Soluzione 2 (punti 5). Il disco (ampio 20 GB) sarà ripartito in: $\frac{20 \times 2^{30} B}{2^9 B} = 20 \times 2^{21} = 40 M$ blocchi. Su questa base possiamo operare il raffronto.

Costo spaziale.

Mappa di bit: La mappa, a dimensione lineare nel numero di blocchi, occuperà 40 Mbit, pari a 5 MB, e verrà ripartita su: $\frac{5 \times 2^{20} B}{2^2 B/W} = 5 \times 2^{18} = 5 \times 256 \times kW = 1280 kW$, ovvero 10 k blocchi. Stante la sua non trascurabile dimensione, la mappa sarà senz'altro residente su disco, a partire da un blocco noto e fissato.

Lista concatenata: La lista concatenata dovrà semplicemente prevedere 1 campo puntatore al 1° blocco libero. Questo blocco, che si trova ovviamente su disco, conterrà l'indirizzo del blocco libero successivo (anch'esso su disco) e così via. L'indice di blocco dovrà avere ampiezza pari alla prima potenza di 2 maggiore di 40 M, ossia $64 M = 2^6 \times 2^{20} = 2^{26}$, pertanto occupando 26 bit, così minuscola da poter essere permanentemente ospitata in memoria principale e pari ad un singolo blocco se mai dovesse essere riposta su disco, 10 k volte meno di quanto richiesta dall'altra tecnica.

Costo temporale.

Mappa di bit: Il 120° blocco libero verrà localizzato con il seguente algoritmo:

1. carica in memoria principale il 1° blocco della mappa, il quale conterrà: $\frac{512 B}{4 B/W} = 128 W$ [costo: 10 ms.]
2. sapendo che, sotto le ipotesi date, ogni parola della mappa avrà il 50% dei bit ad '1', ossia 16 per parola, per $\lfloor \frac{120}{16} \rfloor = 7$ volte, conta il numero di occorrenze di '1', totalizzandone $7 \times 16 = 112$ [costo: $7 \times 1 = 7$ ns.]
3. rileva la 1ª occorrenza di '1' nell'8ª parola [costo: 1 ns.]
4. scandisci la parola fino all'occorrenza dell'8ª occorrenza di '1' in essa [costo: $(8 - 1) \times 1 = 7$ ns.]

per un tempo totale di: $10 \text{ ms.} + (7 + 1 + 7) \text{ ns.} = 10,015 \text{ ms.}$

Lista concatenata: Il 120° blocco libero verrà localizzato al 120° accesso a disco, per un costo temporale complessivo di: $120 \times 10 \text{ ms.} = 1,2 \text{ s.}$, quasi 120 volte superiore al tempo impiegato con l'altra tecnica.

Osservando che il tempo di accesso al 1° blocco libero è lo stesso per entrambe le tecniche e che il sistema operativo richiede normalmente accesso al 1° blocco od a piccoli gruppi di essi, è immediato concludere come la lista concatenata sia, per grandi dimensioni di disco, largamente da preferire alla mappa di *bit*.

Soluzione 3 (punti 4). La figura 2 illustra la rappresentazione insiemistica data:

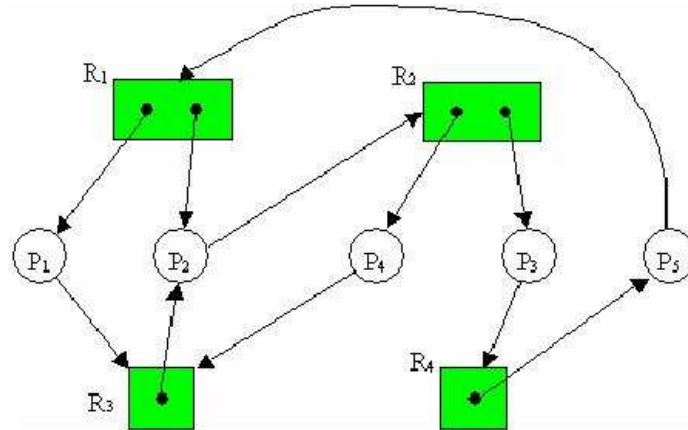


Figura 2: Grafo di allocazione delle risorse per il sistema in esame.

In essa si distinguono tre percorsi chiusi:

- percorso 1: $\{P_2 \rightarrow R_2 \rightarrow P_4 \rightarrow R_3 \rightarrow P_2\}$
- percorso 2: $\{P_5 \rightarrow R_1 \rightarrow P_2 \rightarrow R_2 \rightarrow P_3 \rightarrow R_4 \rightarrow P_5\}$
- percorso 3: $\{P_1 \rightarrow R_3 \rightarrow P_2 \rightarrow R_2 \rightarrow P_3 \rightarrow R_4 \rightarrow P_5 \rightarrow R_1 \rightarrow P_1\}$

in tutti i quali risulta presente almeno una risorsa a molteplicità multipla (i.e. > 1), il che ci impedisce di affermare alcunché a priori, imponendo l'analisi della situazione specifica.

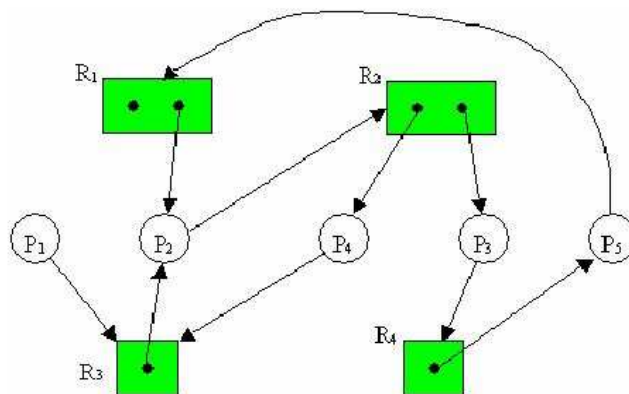
Tutti i percorsi condividono la risorsa R_2 e pertanto anche la stessa caratteristica: o sono tutti bloccati o si potranno liberare in qualche istante futuro. È immediato vedere che tutti i processi e tutte le risorse appartengono ad almeno uno dei percorsi chiusi identificati. Ciò rende impossibile che alcuno dei processi possa portare regolarmente a termine il proprio lavoro o che alcuna risorsa si liberi. Conseguentemente possiamo affermare che il sistema è certamente bloccato in una situazione di stallo.

Prendiamo ora in esame quanto si verificherebbe qualora il processo P_1 rilasciasse la risorsa R_1 . Normalmente, tale situazione non potrebbe verificarsi per volontà del processo P_1 che, trovandosi in situazione di stallo, non ha modo di far avanzare la propria esecuzione. Nondimeno, il rilascio della risorsa potrebbe essere determinato da fattori esterni, per esempio essere imposto da meccanismi di prevenzione messi in atto dal Sistema Operativo.

La situazione conseguente è illustrata in figura 3:

La risorsa R_1 ha ora un'istanza disponibile, e può pertanto soddisfare la richiesta di P_5 , il quale processo può riprendere ad avanzare, mentre tutti gli altri rimangono bloccati. Al termine della sua esecuzione P_5 rilascia R_1 (il che non ha effetti rilevanti sul sistema) ed R_4 . Quest'ultimo evento consente di soddisfare la richiesta di P_3 , che può così riprendere ad avanzare. Al termine dell'esecuzione di P_3 , anche R_2 viene liberata, sbloccando P_2 , che, terminando a sua volta, libererà R_3 . Quest'ultima risorsa è richiesta da P_1 e P_4 . Sappiamo che P_1 è stato forzato al rilascio di R_1 e dunque potrebbe essere in stato anomalo. Indipendentemente da ciò, P_4 potrà certamente accedere ad R_3 e terminare regolarmente.

In definitiva, possiamo concludere che il rilascio di R_1 determina in effetti la risoluzione dello stallo.

Figura 3: Grafo di allocazione delle risorse al rilascio di R_1 da parte di P_1 .

Soluzione 4 (punti 5+2). Si noti che il quesito descrive erroneamente il valore del campo DF . Il valore corretto è: 1 per assenza di frammentazione e 0 altrimenti. (L'uso invertito non è stato naturalmente considerato come errore.)

Ramo A – R1

Capienza dati di trama	$T1 = 1014 B$
Ampiezza richiesta	$(20 + 920) = 940 B < T1$
Invio <u>senza</u> frammentazione	
TL	940
DF	1
MF	0
FO	non significativo
ID	non significativo

Ramo R1 – R2

Capienza dati di trama	$T2 = 504 B$
Ampiezza richiesta	$(20 + 920) = 940 B > T2$

L'invio avviene in frammenti di ampiezza $n \times 8 B \leq T2$ per $n \in \mathbb{N}$ (con l'eccezione dell'ultimo frammento, che può essere più piccolo). Per quanto $T2$ sia già un multiplo di 8, scegliamo un'ampiezza di frammento inferiore in vista del fatto che il ramo successivo ha portanza $T3 = 500 < T2$. Converrà allora scegliere l'ampiezza: $(20 + 476) = 496 B < T3 < T2$. Conseguentemente, il 2° frammento avrà dimensione: $20 + (920 - 476) = 20 + 444 = 464 B < T2$.

frammento 1 di 2	
TL	496
DF	0
MF	1
FO	0 (espresso in frammenti!)
ID	$a12b$ (per esempio)
frammento 2 di 2	
TL	464
DF	0
MF	0
FO	1
ID	$a12b$ (per esempio)

Ramo R2 – R

Capienza dati di trama	$T3 = 500 B$
Ampiezza richiesta	$[496, 464] B < T3$
Invio di ciascun frammento <u>senza</u> ulteriore frammentazione	
frammento 1 di 2	(come nel ramo precedente)
frammento 2 di 2	(come nel ramo precedente)

Si noti che la deframmentazione ha luogo solo a destinazione.

Soluzione 5 (punti 5+2). Come mostrato in figura 4, le tre porte del router1 (E0, E1 ed E2) identificano altrettanti reti locali:

- LAN-0 rete esterna, di collegamento tra ISP e rete Aziendale
- LAN-1 settore non protetto della rete aziendale (detto in gergo “*demilitarized zone*”, DMZ), contenente i servizi Web resi pubblici all'esterno
- LAN-2 rete aziendale interna, composta da un unico dominio di diffusione.

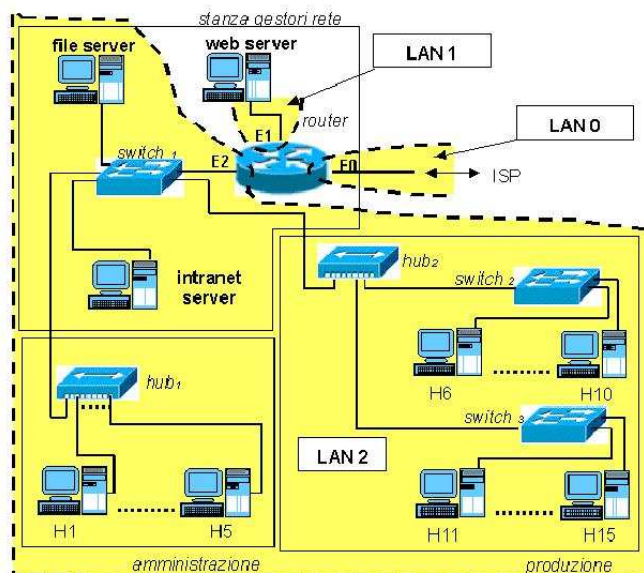


Figura 4: Individuazione delle sottoreti presenti nella rete Aziendale.

Il calcolo dei flussi nel caso peggiore può essere pertanto affrontato analizzando la situazione della sola rete interna LAN-2. Per prima cosa occorre individuare i flussi utili, ossia quelli descritti dal testo. Detti:

- A flusso dati prodotto da un generico utente del reparto amministrazione (H1 ... H5), composto al 50% da un flusso X scambiato con il file server e, per il rimanente 50% da un flusso Y scambiato con l'intranet server
- Z flusso dati prodotto dal generico utente del reparto produzione (H6 ... H15) scambiato con il file server

si ottiene facilmente la distribuzione mostrata in figura 5, ove il traffico utile è indicato in grassetto, ed il traffico effettivo è racchiuso in un ovale.

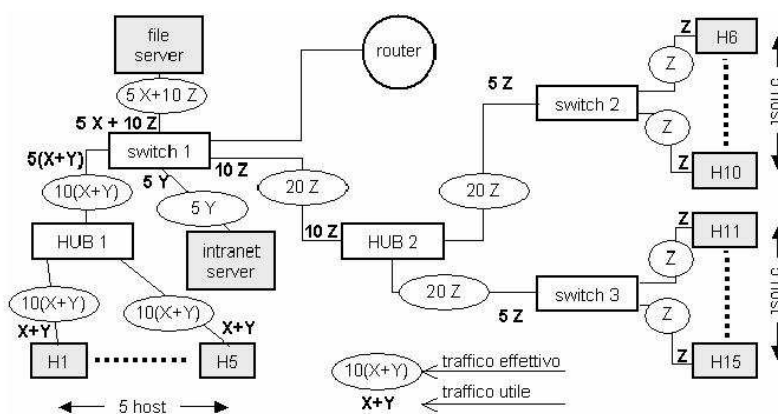


Figura 5: Determinazione dei flussi utili di caso peggiore.

Imponendo il dato di progetto che ogni ramo della rete possa trasferire al massimo 100 Mbps, otteniamo le seguenti condizioni:

$$10A = 10(X + Y) \leq 100 \text{ Mbps} \quad (1)$$

$$X = Y \quad (2)$$

$$20Z \leq 100 \text{ Mbps} \quad (3)$$

che, risolte a sistema, portano ai seguenti valori massimi:

$$X = 5 \text{ Mbps} \quad Y = 5 \text{ Mbps} \quad Z = 5 \text{ Mbps}$$

Per quanto concerne la ripartizione degli indirizzi IP all'interno della rete Aziendale, l'indirizzo IP statico acquistato dall'Azienda è un dato di fatto non influente, che ci consente di omettere l'analisi della configurazione della porta E0 del router. Per gli altri dispositivi abbiamo invece a disposizione un'intera classe C di indirizzi riservati 192.168.1.0. Ricordando quanto esposto in precedenza, dobbiamo considerare 2 sole sottoreti, LAN-1 e LAN-2.

Il dato di progetto da cui partire per l'individuazione del *subnetting* è costituito dai vincoli che tutte le sottoreti abbiano la medesima dimensione (e dunque la medesima *subnet mask*) ed anche la massima dimensione consentita.

Dovendo creare 2 sottoreti utili e sapendo che il numero di sottoreti utili S è dato dall'equazione $S = 2^N - 2$, dove N è il numero di *bit* sottratti alla parte nodo dell'indirizzo ed utilizzati per designare la sottorete interna, fissando $S = 2$ otteniamo $N = 2$. In definitiva, oltre alla parte rete (esterna), ampia 24 *bit*, come sancito per gli indirizzi di classe C, avremo una parte sottorete (interna) ampia 2 *bit* ed una parte nodo ampia 6 *bit*.

Con 6 *bit* di parte nodo possiamo identificare fino a $2^6 - 2 = 62$ indirizzi IP utili, più che sufficienti per trattare il problema proposto.

192	168	1	0	
11000000	10101000	00000001	00	000000
rete			sottorete	nodo

Tabella 1: Base dell'insieme di indirizzi di classe C riservati ad uso interno dell'Azienda.

La tabella 2 riporta le caratteristiche degli indirizzi IP delle due sottoreti utili.

parte di rete	subnetting	parte di nodo		destinazione indirizzo
11000000 . 10101000 . 00000001.	01	000000	192.168.1.64	riservato per la I sottorete
11000000 . 10101000 . 00000001.	01	000001	192.168.1.65	utilizzabile per 1° nodo nella I sottorete
...				
11000000 . 10101000 . 00000001.	01	111110	192.168.1.126	utilizzabile per ultimo nodo nella I sottorete
11000000 . 10101000 . 00000001.	01	111111	192.168.1.127	riservato al <i>broadcast</i> nella I sottorete
11000000 . 10101000 . 00000001.	10	000000	192.168.1.128	riservato per la II sottorete
11000000 . 10101000 . 00000001.	10	000001	192.168.1.129	utilizzabile per 1° nodo nella II sottorete
...				
11000000 . 10101000 . 00000001.	10	111110	192.168.1.190	utilizzabile per ultimo nodo nella II sottorete
11000000 . 10101000 . 00000001.	10	111111	192.168.1.191	riservato al <i>broadcast</i> nella II sottorete

Tabella 2: Ripartizione degli indirizzi nelle 2 sottoreti interne utili.

Da questa ripartizione possiamo determinare una possibile attribuzione di indirizzi IP alle varie postazioni della rete aziendale, riportata in tabella 3.

rete	dispositivo	IP address	subnet mask	default gateway
LAN-1	Web server	192.168.1.125	255.255.255.192	192.168.1.126
	router E1	192.168.1.126	= /26	non definibile
LAN-2	H1	192.168.1.129	/26	192.168.1.190
	H2	192.168.1.130	/26	=
	...			
	H14	192.168.1.142	/26	=
	H15	192.168.1.143	/26	=
	intranet server	192.168.1.188	/26	=
	file server	192.168.1.189	/26	=
	router E2	192.168.1.190	/26	non definibile

Tabella 3: Una possibile attribuzione di indirizzi IP interni alle postazioni della rete Aziendale.

Soluzione 6 (punti 4). Per ottenere l'effetto preventivo desiderato è sufficiente che il “*vacation daemon*” tenga traccia del nome (o dell'indirizzo) di chiunque sia stato la prima volta destinatario di una notifica automatica ed invii ulteriori notifiche solo a destinatari non ancora incontrati.