

Quesito 1. Il progettista di un elaboratore dotato di un dispositivo puntatore di tipo *mouse* ha previsto di supportarne una velocità massima di spostamento pari a 20 cm/s. Tale *mouse* rileva ogni spostamento di 0,1 mm su uno qualunque degli assi del suo sistema di riferimento (x,y) ed ogni pressione di uno dei suoi 3 pulsanti da parte dell'utente, costruendo un messaggio di tipo $\langle \Delta_x, \Delta_y, \text{pulsante} \rangle$, di ampiezza 3 byte, ad ogni interrogazione da parte dell'elaboratore, nel quale si rappresenta la variazione intercorsa dalla precedente interrogazione. Il progettista desidera effettuare rilevazioni periodiche (*polling*) per prelevare messaggi da parte del *mouse* nei quali le componenti Δ_x e Δ_y non siano mai > 1 . Sapendo che il tempo di esecuzione di ogni rilevazione, comprensivo dell'aggiornamento dello schermo e dell'attivazione di eventuali comandi associati alla pressione sui pulsanti, è pari a $60 \mu s$, si stimi la percentuale di tempo impiegato dall'elaboratore al servizio del *mouse*.

Quesito 2. Lo schema logico riportato in figura 1 rappresenta una rete dati costituita da due reparti operativi. Il reparto operativo 1 è composto da 6 postazioni di lavoro, facenti capo al SERVER1 interno alla rete LAN1, il cui traffico è prevalentemente di tipo utente-servente. Le 12 postazioni del reparto operativo 2, anch'esse caratterizzate da un traffico prevalentemente di tipo utente-servente, fanno però capo al SERVER2 interno alla rete LAN2. Il SERVER3 offre servizi *web* agli utenti esterni, e viene aggiornato internamente solo dalla postazione *host* H1 con un traffico massimo previsto di 200 kbps. A tutti gli utenti della sola LAN1 (escluso quindi il SERVER1) deve essere garantito anche un traffico di accesso ad *Internet* di valore pari a 100 kbps.

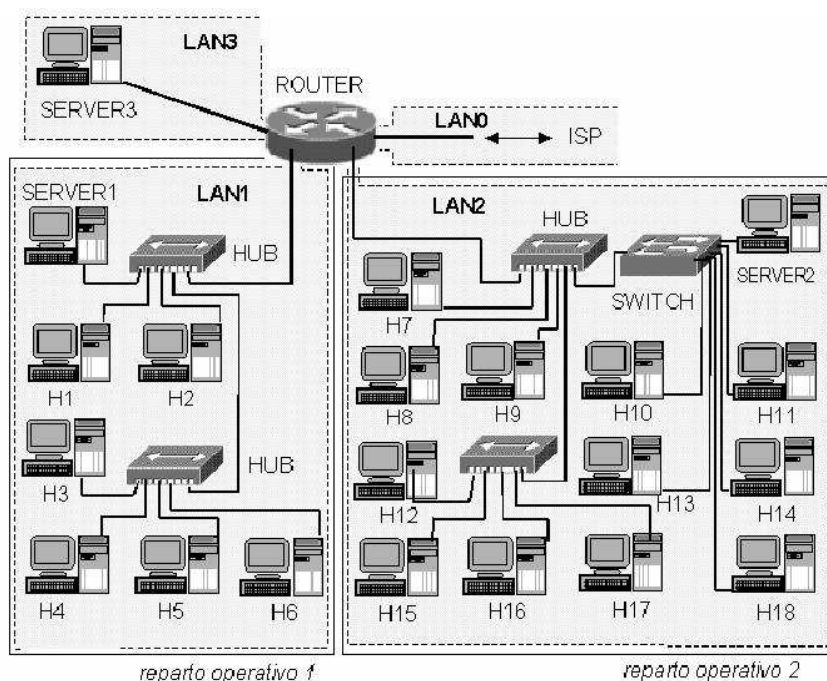


Figura 1: Schema della rete dati.

Sapendo che tutti i dispositivi di rete sono di standard 802.3 base, pertanto operanti a 10 Mbps, si specifichino i flussi di traffico massimo determinati dalla configurazione *hardware* della rete dati nel caso peggiore di traffico contemporaneo di tutti gli utenti.

Al router, visto dal lato LAN0, è stato assegnato dal proprio ISP il seguente indirizzo statico di accesso ad Internet: indirizzo IP = 98.1.10.4 subnet_mask = 255.255.255.0.

Sfruttando la tecnologia NAT (*Network Address Translation*) implementata nel router, si vuole suddividere la rete dati aziendale in 14 sottoreti interne, delle quali le prime 2 utili sono LAN1 e LAN2 nella configurazione mostrata in figura, impiegando gli indirizzi privati della rete 192.168.255.0.

Si proponga una possibile assegnazione degli indirizzi IP a tutti i dispositivi di rete utilizzando la tecnica del *subnetting*, compilando una tabella, comprensiva di indirizzo IP, maschera, ed attribuzione, per ogni sottorete.

Quesito 3. Un sistema in tempo reale, con ordinamento a priorità con prerilascio, comprende 3 processi periodici infiniti, denominati A-C, tutti pronti per l'esecuzione al tempo $t = 0$, e dotati delle caratteristiche illustrate in tabella 1 dove il valore 3 denota la priorità maggiore ed 1 quella minore.

Processi	A	B	C
Tempo di esecuzione (C)	3	1	4
Periodo (T)	10	5	10
Priorità (P)	2	3	1

Tabella 1: Caratteristiche dei processi del sistema dato.

Tale sistema è dotato di 2 orologi:

- uno, denominato GPT, opera in modalità ciclica e fornisce alla primitiva *clock* il valore del tempo corrente, espresso in unità di tempo trascorse a partire dall'istante $t = 0$
- l'altro, denominato RTC, opera in modalità non ripetitiva, con valore di inizializzazione determinato dal sistema operativo sulla base delle successive invocazioni della primitiva *suspend_until* da parte dei processi.

Ciascun processo periodico ripete infinitamente il proprio ciclo di lavoro secondo il seguente schema:

```
while(true){
    ...                // esegui lavoro (1)
    tempo = tempo + periodo; // calcola il tempo del prossimo risveglio (2)
    suspend_until(tempo);    // prenota il risveglio (3)
}
```

dove l'azione (1) ha durata C e le azioni (2-3) hanno durata nulla e dove la variabile `periodo` vale il proprio T per ogni processo e la variabile `tempo` viene inizializzata dal sistema operativo allo stesso valore 0 per tutti i processi. Il sistema operativo mantiene una lista ordinata di prenotazioni di risveglio, utilizzando l'orologio RTC per farle avvenire al momento richiesto. Ad ogni istante, il primo valore in tale lista ordinata è anche quello che il sistema operativo userà per l'inizializzazione dell'orologio RTC.

Si mostri il contenuto di tale lista ordinata durante un intero ciclo di esecuzione di tutti e 3 i processi del sistema, trascurando il tempo di commutazione del contesto.

Quesito 4. Il progettista di un sistema operativo ha deciso di usare nodi indice (*i-node*) per la realizzazione del proprio *file system*, stabilendo che essi abbiano la stessa dimensione di un blocco, fissata a 256 *byte*. Lo stesso progettista ha poi deciso che un nodo indice primario contenga 12 campi di indirizzo di blocchi di disco e 3 campi puntatori a nodi indice rispettivamente di primo, secondo e terzo livello di indirizzione. Sapendo che gli indirizzi di blocco sono espressi su 32 *bit*, si vuole allocare un *file* composto da 8.500 *record* da 75 *byte* ciascuno, imponendo che un *record* non possa essere suddiviso su due blocchi. Si determini quanti blocchi verranno utilizzati per allocare il *file* e quanti per gestirne l'allocazione tramite nodi indice e per ciascun livello di indirizzione. Si determini inoltre l'occupazione totale risultante in memoria secondaria.

Quesito 5. Si illustrino le caratteristiche degli indirizzi IP in classe C e si fornisca l'esempio di una maschera che realizzi *supernetting* su un indirizzo in quella classe.

Soluzione 1 (punti 4). L'impegno del progettista è di rilevare spostamenti elementari del *mouse*, pari a 0,1 mm su uno qualunque degli assi di riferimento, fino alla velocità massima di 20 cm/s. (Si noti che lo spostamento elementare ammesso viene scherzosamente chiamato "*mickey*".) Ne segue che la frequenza di interrogazione verso il *mouse* è data da: $\frac{20 \text{ cm/s}}{0,1 \text{ mm}} = 2000/s$, con il conseguente impiego di: $2000/s \times 60 \mu s = 120000 \mu s/s = 12\%$ del tempo di elaborazione per il servizio del *mouse*.

Soluzione 2 (punti 10). Il *router* separa le reti locali, e poiché non abbiamo ingerenze reciproche di traffico tra le due reti locali LAN1 e LAN2, possiamo stabilirne il traffico separatamente.

LAN1

Sia X il traffico massimo fruibile da un *host*. Dato che il traffico predominante è di tipo *host-server* e vi sono 6 *host*, il traffico gestibile dal *server* varrà $6X$. La rete viene gestita mediante 2 *hub* in cascata, il che, funzionalmente equivale ad averne uno solo con la somma delle porte a disposizione. Il traffico prodotto da ogni dispositivo viene inoltrato su tutte le altre porte, per cui ogni porta degli *hub* gestirà un traffico complessivo dato da $6X + 6X = 12X$.

Per il traffico secondario di tipo *Internet*, cioè diretto da ogni *host* alla porta di questa LAN del *router*, detto I il suo valore, con ragionamenti del tutto analoghi possiamo osservare che il suo contributo in ogni porta varrà ancora $6I + 6I = 12I$.

Il traffico totale su ogni porta dell'*hub* varrà allora $12X + 12I$. Questa espressione deve avere come limite massimo teorico il valore della banda del sistema, cioè 10 Mbps, mentre dal testo del quesito sappiamo che I vale 0,1 Mbps. Occorre ricordare infine che l'*host* H1 genera anche un traffico di 200 kbps diretto al *web server*, il cui contributo si ripartirà tra tutte le porte degli *hub*.

Pertanto, da: $10 \text{ Mbps} = 12X + 12I + 2 \times 0,2 = 12X + 12 \times 0,1 + 0,4 = 12X + 1,6$ ricaviamo che $X = (10 - 1,6)/12 = 0,7 \text{ Mbps}$.

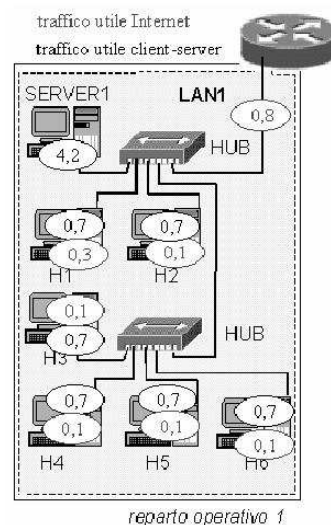


Figura 2: Stima del traffico massimo teorico nel reparto operativo 1.

LAN2:

Questa rete è caratterizzata da due *hub* ed uno *switch* che la segmentano in 7 domini di collisione: 1 verso il *router* ed i due *hub* in cascata cui fanno capo 7 *host*; 1 verso il *server*2; ed i 5 rimanenti verso i 5 *host* connessi con lo *switch*.

Per il calcolo del traffico, come già fatto in precedenza, osserviamo che due *hub* in cascata equivalgono funzionalmente ad un solo *hub*, per cui conviene far riferimento allo schema semplificato mostrato in figura 3. Il traffico di tipo utente-servente viene limitato dagli *hub*, in quanto il traffico massimo X fruibile da un *host* deve condividere la banda di una porta con gli altri $6X$ determinati dai flussi analoghi degli altri 6 *host* collegati ai due *hub*, e dal traffico che fa capo al *server*, di valore $7X$ proveniente da un ramo dello *switch*. La banda massima di ogni porta dell'*hub* dovrà quindi essere ripartita tra i traffici $X + 6X + 7X = 14X$. Il traffico risultante deve quindi soddisfare l'equazione: $10 \text{ Mbps} = 14X$, da cui $X = 10/14 = 0,71 \text{ Mbps}$.

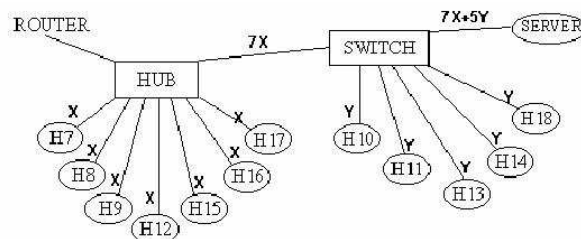


Figura 3: Schema semplificato del reparto operativo 2.

Per i 5 *host* che fanno capo allo *switch* non vi sarebbe alcun limite teorico, in quanto lo *switch* è in grado di garantire banda massima ad ogni segmento di rete. Il valore Y di traffico utile usufruibile da ciascuno di questi *host* potrebbe allora essere di 10 Mbps. La presenza del *server* in un ramo dello *switch* e la caratteristica del traffico utente-servente verso l'altro segmento di LAN2 impongono tuttavia che sul ramo collegato al *server* si stabilisca un traffico utile massimo di $7X$ proveniente dal ramo diretto verso gli *hub* più $5Y$ proveniente dagli *host* collegati agli altri rami dello *switch*).

La condizione più restrittiva da rispettare, che determina la banda offerta a tutti i nodi della rete, è quella che limita il traffico passante sul ramo del *server*, la quale deve soddisfare l'equazione: $10 \text{ Mbps} = 7X + 5Y$. Ricordando che $X = 0,71 \text{ Mbps}$, otteniamo $Y = (10 - 7 \times 0,71) / 5 = (10 - 5) / 5 = 1 \text{ Mbps}$.

In definitiva, i traffici teorici massimi nei vari rami, consentiti dalla configurazione di rete data, sono riportati in figura 4.

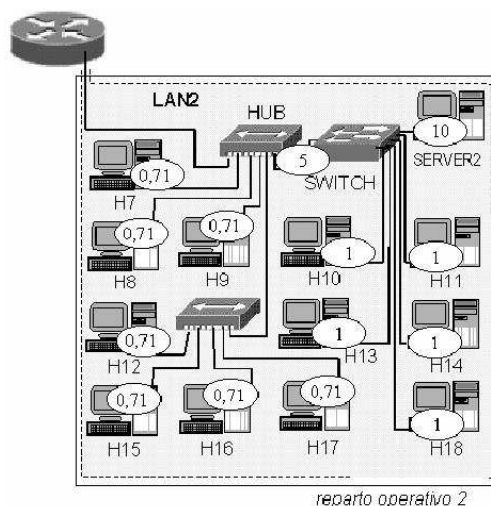


Figura 4: Stima del traffico massimo teorico nel reparto operativo 2.

LAN3

Per la LAN3 il calcolo del traffico è banale, in quanto esso si riduce al bilanciamento tra il traffico di accesso interno al *web server* generato dall'*host* H1, che l'esercizio fissa in 200 kbps, e quello proveniente dall'esterno, che potrebbe valere fino ad un massimo di $10 - 0,2 = 9,8 \text{ Mbps}$. Questo valore va però lievemente ritoccato in quanto il traffico *Internet* totale garantito agli utenti interni vale 6 (pari al numero di utenti abilitati ad accedere ad *Internet*) per 0,1 Mbps garantiti ad ogni utente, per un valore complessivo di 0,6 Mbps, che attraversa il ramo della LAN0 insieme al traffico massimo proveniente dall'esterno, che quindi deve valere: $10 - 0,6 = 9,4 \text{ Mbps}$, invece dei 9,8 calcolati in precedenza.

Attribuzione indirizzi IP

La rete di indirizzo di classe C 192.168.255.0 di cui si vogliono ripartire gli indirizzi IP nella rete interna è ad uso esclusivo e privato della rete dati. Il testo richiede di suddividere la rete in 14 sottoreti. La *subnet mask* da utilizzare deve perciò impiegare 4 degli 8 *bit* previsti dalla classe C per il campo *host* originale.

Infatti, usando 4 *bit* e tenendo conto degli indirizzi inutilizzabili, possiamo rappresentare $2^4 - 2 = 14$ sottoreti distinte, esattamente come richiesto dall'esercizio.

I *bit* rimanenti per l'indirizzamento di nodo sono pertanto i 4 meno significativi, il che consente, all'interno di ogni sottorete, di indirizzare fino a $2^4 - 2 = 14$ nodi distinti, che è sufficiente a soddisfare le nostre esigenze, poiché LAN1 necessita di 8 indirizzi, LAN2 di 14, e LAN3 solo di 2.

Le sottoreti individuabili con la *subnet mask* così determinata sono:

```

11000000 . 10101000 . 11111111 . 0000 xxxx  inutilizzabile (indirizzo della sottorete)
11000000 . 10101000 . 11111111 . 0001 xxxx  1a sottorete utilizzabile (LAN1)
11000000 . 10101000 . 11111111 . 0010 xxxx  2a sottorete utilizzabile (LAN2)
11000000 . 10101000 . 11111111 . 0011 xxxx  3a sottorete utilizzabile (LAN3)
11000000 . 10101000 . 11111111 . 0100 xxxx  4a sottorete utilizzabile (non necessaria)
11000000 . 10101000 . 11111111 . 0101 xxxx  5a sottorete utilizzabile (non necessaria)
...
11000000 . 10101000 . 11111111 . 1110 xxxx  14a sottorete utilizzabile (non necessaria)
11000000 . 10101000 . 11111111 . 1111 xxxx  inutilizzabile (broadcast nella sottorete)

```

La composizione delle tabelle con la ripartizione degli indirizzi IP attribuiti alle sottoreti LAN0-3 è lasciata al lettore per esercizio.

Soluzione 3 (punti 7). Prima di procedere con l'analisi del problema occorre accertarsi che il sistema dato sia ammissibile. Per essere tale, occorre che sia verificata almeno la condizione necessaria, ma non sufficiente: $U = \sum_{i=(A,B,C)} \frac{C_i}{T_i} \leq 1$. È facile vedere che il sistema dato è ammissibile in quanto $U = 0,9$. Osserviamo ora in figura 5 il 1^o ciclo completo di esecuzione dei 3 processi dati, tenendo conto della politica di ordinamento a priorità con prerilascio:

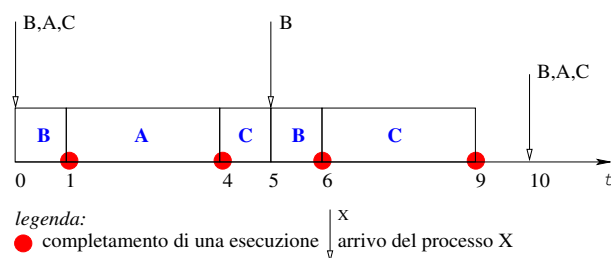


Figura 5: Primo ciclo di esecuzione dei processi del sistema.

Su questa base, e tenendo conto della struttura del codice dei processi descritta nel quesito, possiamo ricostruire la sequenza di chiamate alla primitiva *suspend_until* effettuate dai processi durante l'intervallo d'esecuzione mostrato in figura 5 ed il contenuto della corrispondente lista di prenotazioni di sveglia, la quale è ovviamente ordinata in ordine di *crescente* distanza dal tempo corrente:

Tempo di richiesta	$t = 0$	$t = 1$	$t = 4$	$t = 5$	$t = 6$	$t = 9$
Contenuto della lista	$\emptyset$	4_B	$1_B, 5_A$	$0_B(\uparrow \text{sveglia}_B), 5_A$	$4_A, 0_B$	$1_A, 0_B, 0_C$

Tabella 2: Evoluzione del contenuto della lista delle prenotazioni di sveglia.

Si noti come la lista venga costruita facendo sì che ogni prenotazione sia rappresentata dalla distanza temporale che la separa dall'occorrenza dell'evento che la precede, con il primo valore in lista che vale *sempre*:

tempo - clock

dove tempo è la variabile immessa come parametro di ingresso alla primitiva *suspend_until* invocata dal corrispondente processo. Si noti altresì come non occorra ordinare per valore di priorità decrescente le prenotazioni con scadenza allo stesso istante, poichè tutti i corrispondenti processi dovranno comunque passare in stato di pronto prima che possa avvenire la selezione per l'esecuzione.

Soluzione 4 (punti 6). Richiamiamo i dati del problema:

N_R = numero di *record* che compongono il *file* = 8.500

D_R = dimensione di un *record* = 75 byte

D_I = dimensione di un indirizzo = 4 byte

D_B = dimensione di un blocco = 256 byte

N_{RB} = numero di *record* per blocco = $\text{int}(D_B/D_R) = \text{int}(256/75) = \text{int}(3,41) = 3$

N_{BF} = numero di blocchi occupati dal *file* = $1 + \text{int}(N_R/N_{RB}) = 1 + \text{int}(8500/3) = 2.834$

N_{IB} = numero di indirizzi in un blocco = $D_B/D_I = 256/4 = 64$

I blocchi da indirizzare sono $N_{BF} = 2.834$

- di questi, 12 possono essere indirizzati direttamente dal nodo indice principale
- dei rimanenti $2.834 - 12 = 2.822$, N_{IB} (cioè 64) sono indirizzabili ad indirezione singola tramite l'indirizzo ad indirezione semplice dell'i-node principale * dei rimanenti $2.822 - 64 = 2.758$, si possono utilizzare $1 + \text{int}(2.758/64) = 44$ blocchi indiretti di 64 con indirezione doppia, di cui i primi 43 saranno utilizzati completamente (cioè per $64 \times 43 = 2.752$ indirizzi) mentre i rimanenti $(2.758 - 2.752) = 6$ indirizzi andranno posizionati nel 44º blocco, come mostrato in figura 6.

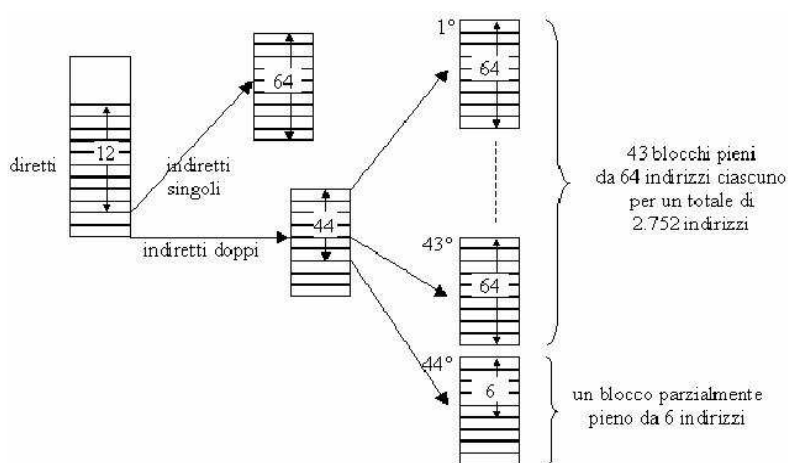


Figura 6: Allocazione del *file*.

Con le indicazioni sopra riportate possiamo concludere che:

- per allocare il *file* dati sono necessari N_{BF} blocchi, cioè 2.834 blocchi
- per gestire l'allocazione del *file* sono necessari: 1 blocco per il nodo indice principale; 1 blocco di indirizzi ad indirezione singola; e 1+44 blocchi per l'indirezione doppia; per un totale di $1+1+1+44 = 47$ blocchi
- l'occupazione totale in memoria secondaria vale $2.834 + 47 = 2.881$ blocchi da 256 byte, per un totale di $2.881 \times 256 = 737.536$ byte = circa 720 kB.

Soluzione 5 (punti 5). Gli indirizzi in classe C riservano 3 byte per la parte di rete ed 1 solo byte per la parte di nodo. Ciò comporta che reti canoniche di classe C non possano ospitare più di $256-2$ nodi distinti. Alla classe C appartengono tutti gli indirizzi IP i cui primi 3 bit da sinistra valgono 110, includendo pertanto i valori compresi tra **192.0.0.0** e **223.255.255.255**, espressi in notazione decimale.

Il meccanismo detto del *supernetting* è inteso consentire ad una rete l'utilizzo di più nodi di quanto le consenta la classe di appartenenza. Nel caso specifico, una qualunque maschera di tipo $/N$ con $N < 24$ consentirebbe ad una rete con indirizzo IP in classe C di ospitare $2^{8+(24-N)} - 254$ nodi aggiuntivi rispetto a quanto previsto dalla classe. Per esempio, la maschera $/20$, corrispondente a 255.255.240.0, applicata ad un indirizzo di classe C consente alla rete a quell'indirizzo di ospitare fino a $2^{8+4} - 2 = 4094$ nodi distinti, invece che solo $2^8 - 2 = 254$.