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 30 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 $< \Delta_x, \Delta_y, pulsante >$, di ampiezza 3 *byte*, della cui presenza informa l'elaboratore tramite interruzione dedicata. Il tempo di servizio di ciascuna di queste interruzioni, comprensivo dell'aggiornamento dello schermo e dell'attivazione di eventuali comandi associati alla pressione sui pulsanti, è pari a 40 μs. Si stimi la percentuale di tempo impiegato dall'elaboratore al servizio del *mouse* nell'ipotesi di massima velocità di movimento, in assenza di pressione di pulsanti da parte dell'utente.

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 5 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.
A tutti gli utenti (esclusi quindi i *server*) deve anche essere garantito un traffico di accesso ad *Internet* di valore pari a 100 kbps.

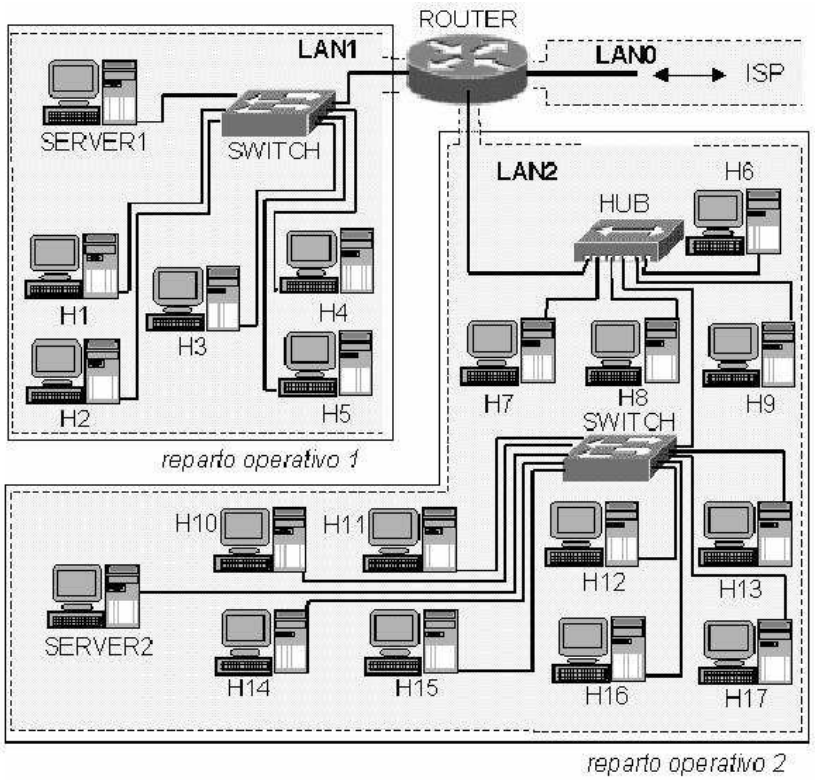

Figura 1: Schema della rete dati.

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 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 = 108.2.108.2 *subnet_mask* = 255.255.0.0. Sfruttando la tecnologia NAT (*Network Address Translation*) implementata nel *router* si vuole suddividere la rete dati aziendale in 6 sottoreti, delle quali le prime 2 utili sono LAN1 e LAN2 nella configurazione mostrata in figura, impiegando gli indirizzi privati della rete 192.168.10.0.
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)	2	2	3
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 RTC, 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 GPT, 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 GPT 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 GPT.
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 2 campi puntatori a nodi indice di primo e secondo livello di indirezione rispettivamente. Sapendo che gli indirizzi di blocco sono espressi su 32 *bit*, si vuole allocare un *file* composto da 14.500 *record* da 60 *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 indirezione. Si determini inoltre l'occupazione totale risultante in memoria secondaria.

Quesito 5. Si illustrino le caratteristiche degli indirizzi IP in classe A e si fornisca l'esempio di una maschera che realizzi *subnetting* 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 30 cm/s. (Si noti che lo spostamento elementare ammesso viene scherzosamente chiamato "*mickey*".) Ne segue che, sotto le ipotesi del quesito, il *mouse* solleverà interruzioni verso l'elaboratore con la frequenza di: $\frac{30\ cm/s}{0{,}1\ mm} = 3000/s$, con il conseguente impiego di: $3000/s \times 40\ \mu s = 120000\ \mu s/s = 12\%$ 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 5 *host*, il traffico gestibile dal *server* varrà $5X$. La rete viene gestita mediante uno *switch*, per cui ogni ramo avrà a disposizione la massima banda disponibile. Sarà quindi il ramo che collega il *server* allo *switch* (che porta un traffico da $5X$) a limitare il valore di X alla banda massima della rete, che vale 10 Mbps. In linea di principio dovremmo anche considerare il traffico di tipo *Internet* che ogni *host* effettua verso la rete esterna LAN0, ma questo non influisce sul ramo *switch-server* che limita il valore di X, e possiamo quindi trascurarlo ai fini di questo calcolo. Da $10 = 5X$ ricaviamo banalmente che $X = 2$ Mbps. I traffici utili sono riportati in figura 2.

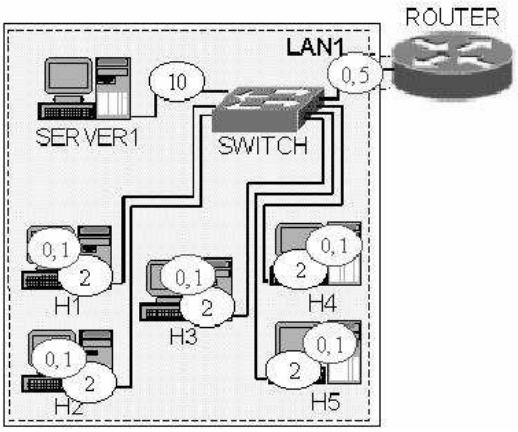

Figura 2: Stima del traffico massimo teorico nel reparto operativo 1.

LAN2:
Questa rete è caratterizzata da un *hub* ed uno *switch* che segmentano la rete in 10 domini di collisione: 1 verso il *router* e l'*hub* cui fanno capo 4 *host*; 1 verso il *server*; e gli 8 rimanenti verso gli 8 *host*. Per il calcolo del traffico possiamo far riferimento allo schema semplificato ostrato in figura 3.

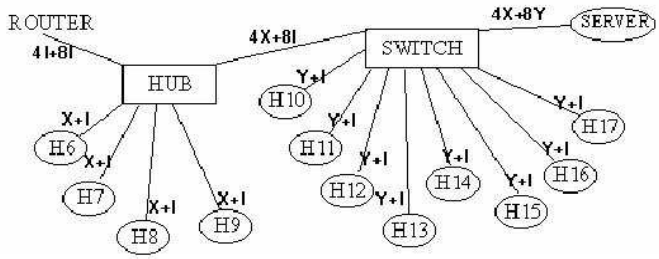

Figura 3: Schema semplificato del reparto operativo 2.

Il traffico, prevalentemente di tipo utente-servente, dei primi 4 *host* viene limitato dall'*hub*, in quanto il traffico massimo X fruibile da un *host* deve condividere la banda di una porta con i flussi analoghi provenienti

dagli altri 3 *host* collegati all'*hub*, e dal traffico che fa capo allo *switch*, di valore $4X$ diretto al *server*. La banda massima di ogni porta dell'*hub* dovrà quindi essere ripartita tra i traffici $X + 3X + 4X = 8X$. Con procedimento analogo possiamo valutare il contributo del traffico I di tipo *Internet*, proprio di ogni nodo e diretto verso il *router*. Abbiamo quindi in ogni nodo *host* un traffico totale di 8X + 8I. Dobbiamo però ricordare che il traffico *Internet* degli altri *host* che fanno capo allo *switch* (di valore complessivo di $8I$, dato che gli *host* sono 8) arriva all'*hub* attraverso il ramo che proviene dallo *switch*, ed anch'esso viene ripetuto dall'*hub* su tutte le sue porte. In definitiva, il traffico totale che attraversa ogni ramo dell'*hub* deve soddisfare l'equazione: $10 = 8X + 8I + 16I$, da cui, ricordando che $I = 0,1$ Mbps, si ricava il traffico massimo degli *host* che fanno capo all'hub: $X = (10 - 24I)/8 = (10 - 2,4)/8 = 0,95$ Mbps.
Per gli 8 *host* che fanno capo allo *switch* non vi sarebbe invece 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*, comprensivo della quota I di traffico di tipo *Internet*, potrebbe allora essere di 10 Mbps. La presenza del *server* in un ramo e la caratteristica del traffico utente-servente impongono tuttavia che sul ramo collegato al *server* si stabilisca un traffico utile massimo di $4X$, proveniente dal ramo collegato all'*hub*, più $8Y$ proveniente dagli *host* collegati ai rami dello *switch*. Conseguentemente, il traffico risultante sul ramo del *server*, che determina la banda offerta a tutti i nodi della rete, deve soddisfare l'equazione: $10 = 4X + 8Y$. Ricordando che $X = 0,95$ Mbps otteniamo: $Y = (10 - 4 \times 0,95)/8 = (10 - 3,8)/8 = 0,775$ Mbps. La componente di traffico *Internet* in questo vincolo è ininfluente, in quanto non attraversa il ramo limitante (quello tra *switch* e *server*) e quindi porta solo ad un aumento del traffico nei rami degli *host* ed, ovviamente, tra lo *switch* e l'hub.
In definitiva, i traffici teorici massimi nei vari rami, consentiti dalla configurazione di rete data, sono riportati in figura 4.

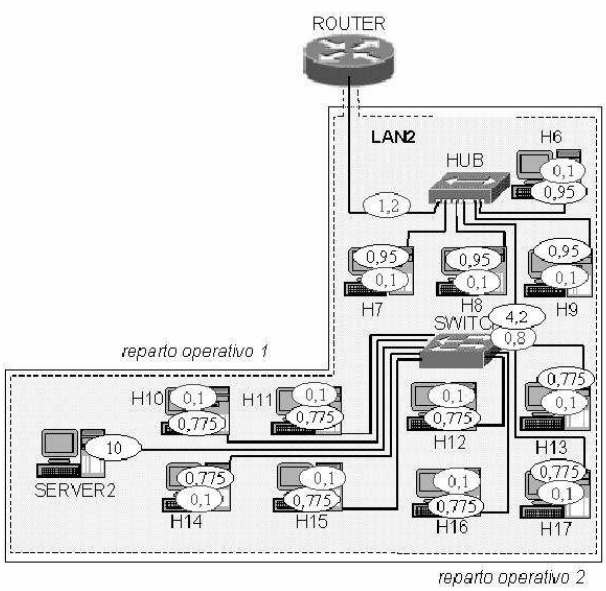

Figura 4: Stima del traffico massimo teorico nel reparto operativo 2.

Attribuzione indirizzi IP
La rete di indirizzo di classe C 192.168.10.0, di cui si vogliono ripartire gli indirizzi IP nella rete interna è ad uso privato ed esclusivo della rete dati aziendale. Il quesito specifica che la rete aziendale deve essere suddivisa in 6 sottoreti. La *subnet mask* da utilizzare deve pertanto impiegare 3 degli 8 *bit* previsti dalla classe C per il campo *host* originale. Impiegando 3 *bit* e tenendo conto degli indirizzi inutilizzabili, otteniamo in effetti $2^3 - 2 = 6$ sottoreti distinte, raggiungendo proprio il valore desiderato.
I *bit* rimanenti per l'indirizzamento di nodo sono pertanto i 5 meno significativi, il che consente, all'interno di ogni sottorete, di indirizzare fino a $2^5 - 2 = 30$ nodi distinti, che è ampiamente sufficiente a soddisfare le nostre esigenze, dato che LAN1 necessita di 7 indirizzi e LAN2 di 14.
Le sottoreti individuabili con la *subnet mask* così determinata sono:

11111111 . 11111111 . 11111111 . 111 00000	*subnet mask*
11000000 . 10101000 . 00001010 . 000 xxxxx	inutilizzabile (indirizzo della sottorete)
11000000 . 10101000 . 00001010 . 001 xxxxx	1^a sottorete utilizzabile (LAN1)
11000000 . 10101000 . 00001010 . 010 xxxxx	2^a sottorete utilizzabile (LAN2)
11000000 . 10101000 . 00001010 . 011 xxxxx	3^a sottorete utilizzabile (non necessaria)
11000000 . 10101000 . 00001010 . 100 xxxxx	4^a sottorete utilizzabile (non necessaria)
11000000 . 10101000 . 00001010 . 101 xxxxx	5^a sottorete utilizzabile (non necessaria)
11000000 . 10101000 . 00001010 . 110 xxxxx	6^a sottorete utilizzabile (non necessaria)
11000000 . 10101000 . 00001010 . 111 xxxxx	inutilizzabile (*broadcast* nella sottorete)

La composizione delle tabelle con la ripartizione degli indirizzi IP attribuiti alle sottoreti LAN0-2 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:

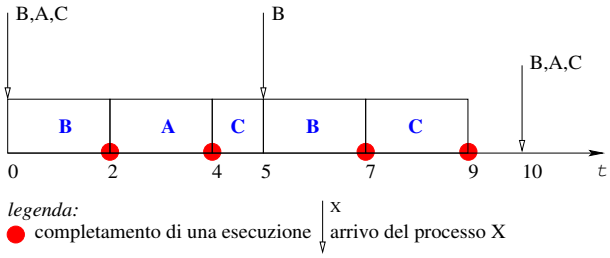

Figura 5: Primo ciclo di esecuzione dei processi del sistema.

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	$t = 0$	$t = 2$	$t = 4$	$t = 5$	$t = 7$	$t = 9$
Contenuto della lista	$\oslash$	3_B	$1_B, 5_A$	$0_B(\uparrow sveglia_B), 5_A$	$3_A, 0_B$	$1_A, 0_B, 0_C$

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* = 14.500
D_R = dimensione di un *record* = 60 *byte*
D_I = dimensione di un indirizzo = 4 *byte*
D_B = dimensione di un blocco = 256 *byte*

N_{RB} = numero di *record* per blocco = $\texttt{int}(D_B/D_R) = \texttt{int}(256/60) = \texttt{int}(4,26) = 4$
N_{BF} = numero di blocchi occupati dal *file* = $1+\texttt{int}(N_R/N_{RB}) = 1+\texttt{int}(14500/4) = 3.625$

N_{IB} = numero di indirizzi in un blocco = D_B/D_I = 256/4 = 64
N_{ID} = numero di indirizzi in X blocchi indiretti = X^2.

I blocchi da indirizzare sono N_{BF} = 3.625

- di questi, 12 possono essere indirizzati direttamente dal nodo indice principale
- dei rimanenti $3.625 - 12 = 3.613$, N_{IB} (cioè 64) sono indirizzabili ad indirezione singola tramite l'indirizzo ad indirezione semplice del nodo indice principale
- per i rimanenti $3.613 - 64 = 3.549$, si possono utilizzare $1 + \texttt{int}(3549/64) = 56$ blocchi indiretti di 64 indirizzi ad indirezione doppia, dei quali i primi 55 saranno utilizzati completamente (cioè per $64 \times 55 = 3.520$ indirizzi), mentre i residui $3.549 - 3.520 = 29$ indirizzi andranno posizionati nel 56^o blocco, come mostrato in figura 6.

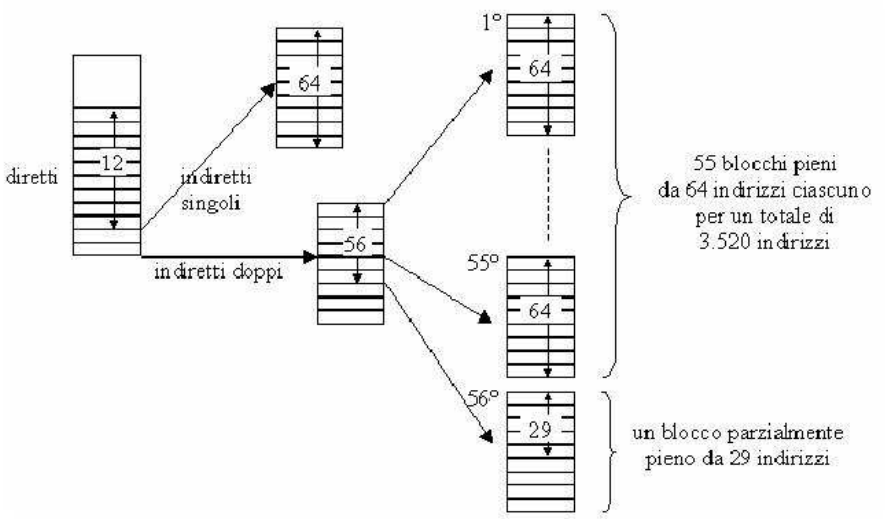

Figura 6: Allocazione del *file*.

Con le indicazioni riportate in figura possiamo concludere che:

- per allocare il *file* dati sono necessari N_{BF} blocchi, cioè 3.625 blocchi
- per gestire l'allocazione del *file* sono necessari: 1 blocco per i nodo indice principale; 1 blocco di indirizzi ad indirezione singola; 1+56 blocchi per l'indirezione doppia; per un totale di $1 + 1 + 1 + 56 =$ 59 blocchi
- l'occupazione totale in memoria secondaria vale $3.625 + 59 = 3.684$ blocchi da 256 *byte* ciascuno, per un totale di $3.684 \times 256 = 943.104$ *byte* = 921 kB.

Soluzione 5 (punti 5). Gli indirizzi in classe A riservano solo 1 *byte* per la parte di rete e 3 *byte* per la parte di nodo. Alla classe A appartengono tutti gli indirizzi IP il cui primo *bit* da sinistra vale 0, includendo pertanto i valori compresi tra **0**.0.0.0 e **127**.255.255.255, espressi in notazione decimale, con gli indirizzi 0.0.0.0 e 127.0.0.1 usati da ciascun nodo per auto-identificarsi prima della propria connessione alla rete. Ciò comporta che non vi possano essere più di $128 - 2$ indirizzi canonici distinti di reti di classe A.
Il meccanismo detto del *subnetting* è inteso estendere l'ampiezza della parte di rete dell'indirizzo oltre il limite fissato dalla classe di appartenenza, sottraendo *bit* alla parte di nodo. Nel caso specifico, una qualunque maschera di tipo $/N$ con $N > 8$ consentirebbe ad un indirizzo IP in classe A di suddividere la rete logica designata dall'indirizzo in $2^{N-8} - 2$ sottoreti interne, ciascuna capace di ospitare $2^{24-(N-8)} - 2$ nodi distinti. Per esempio, la maschera $/12$, corrispondente a 255.240.0.0, applicata ad un indirizzo di classe A consente alla rete logica a quell'indirizzo di ospitare al suo interno fino a $2^{12-8} - 2 = 14$ sottoreti distinte, ciascuna capace di $2^{24-(12-8)} - 2 = 2^{20} - 2$ nodi distinti. Si noti, infine, che questo procedimento di *subnetting* può essere liberamente ripetuto all'interno di ogni sottorete.