

Per ampliare il numero di nodi e host su una rete LAN, la tecnologia IEEE 802.3\* mette a disposizione una serie di apparati di interconnessione atti a risolvere varie problematiche come:

- l'attenuazione di segnale per stazioni distanti (ripetitori o repeater);
- l'attestazione dei nodi su un centro stella all'apparire della tecnologia \*BASE-T (concentratori o hub);
- la separazione dei domini di collisione e la tolleranza ai guasti (bridge e switch);
- la separazione dei domini di broadcast (managed switch/VLAN)
- la gestione di sistemi virtuali (switch virtuali).

Alcuni di questi dispositivi sono ormai obsoleti (o legacy), come repeater, hub e bridge; ciononostante la letteratura di settore li tratta ancora per comprendere meglio le dinamiche di circolazione dei pacchetti.

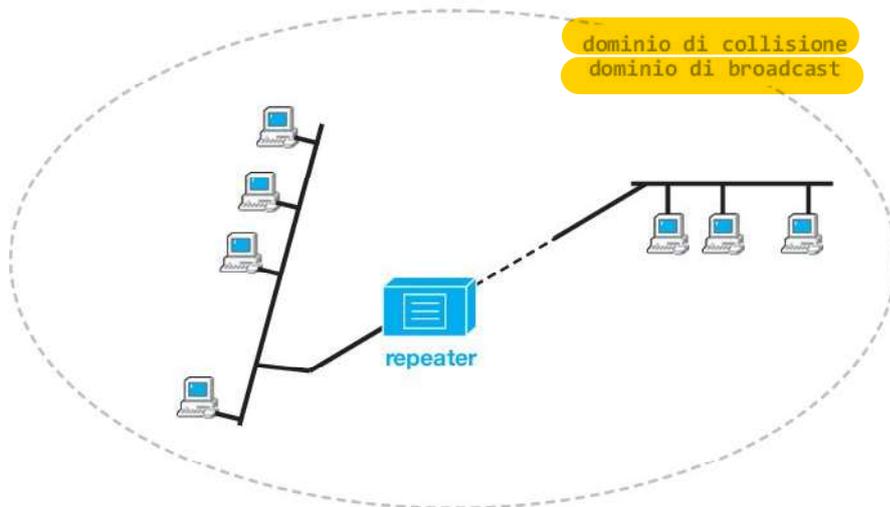
Molti di questi dispositivi, inoltre, sono detti trasparenti, ovvero non necessitano di configurazione né manutenzione da parte di un operatore tecnico. Nel gergo delle reti l'attività di configurazione di un dispositivo da parte di un operatore viene detta management.

I dispositivi elencati, soprattutto quelli più evoluti, operano a vari livelli del modello OSI, anche fino al livello 7 Applicazione; per coerenza espositiva si analizzeranno solo i dispositivi operanti fino al livello 2 Datalink, anche se qualche accenno ad apparati di livello 3 Network sarà necessario per meglio comprendere i concetti.

## 1 Ripetitore

Il ripetitore (repeater) serve per connettere tratte di rete molto distanti in cui il segnale elettrico può deperire, prolungando la distanza massima raggiungibile. Il ripetitore è un apparato obsoleto.

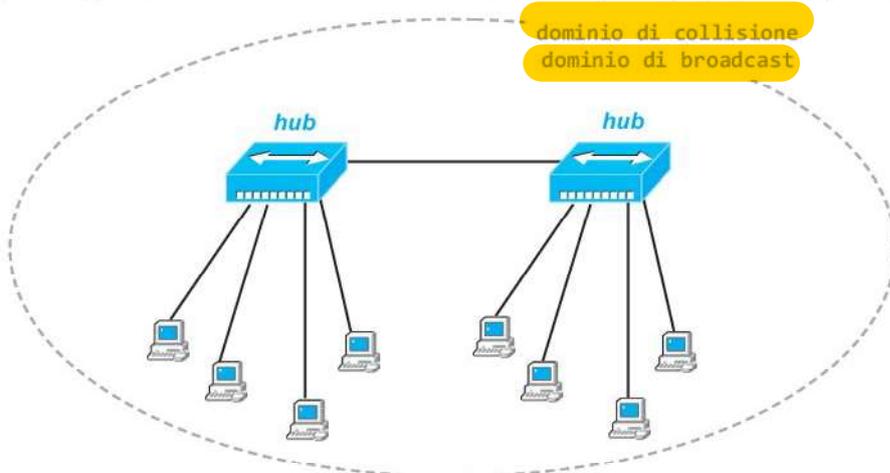
La funzione di un ripetitore è solo attiva a livello 1 Fisico; tipicamente realizzati con due porte, i ripetitori si limitano ad agire amplificando il segnale digitale in ingresso, risincronizzandolo e rigenerando i preamboli in uscita. Ricreare il preambolo significa avere una memoria di transito e creare un breve gap di tempo nella trasmissione; quindi per non superare il tempo limite di trasmissione previsto dal modello teorico è comunque necessario dare un limite massimo al numero di ripetitori utilizzabili in cascata.



Il ripetitore estende il dominio di collisione e il dominio di broadcast.

## 2 Concentratore → Hub

Quando la tecnologia 802.3 10BASE-T prese piede, si utilizzarono questi concentratori (**hub**) come centro stella della topologia fisica (la topologia logica rimane sempre il bus). L'hub quindi ha un certo numero di **porte** di input/output su cui connettere le stazioni della rete (esempio, 4, 8 o 32 porte).



L'hub è un apparato *obsoleto*.

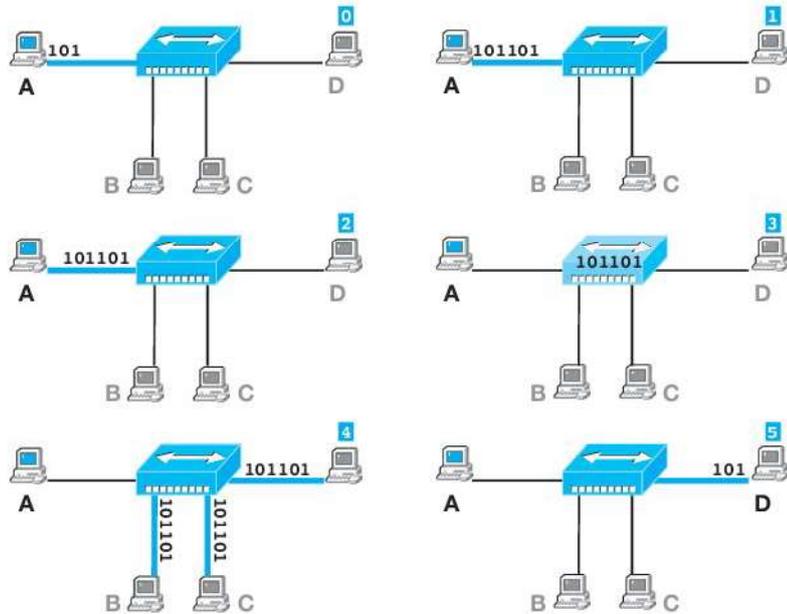
Anche l'hub lavora solo a livello 1 Fisico, e rigenera il segnale elettrico in ingresso verso la stazione destinataria, avendo quindi anche la funzione di ripetitore. Per questioni economiche e di scalabilità gli hub prevedono, oltre alle porte di I/O per i nodi collegati, un paio di porte denominate **uplink** per collegare in cascata più hub tra loro e quindi aumentare il numero di nodi totali nel centro stella, ottenendo un minimo grado di tolleranza ai guasti (*fault tolerance*).

Con la tecnologia 10BASE-T viene anche introdotto un rudimentale meccanismo di **autonegoziazione** attivata quando il NIC è in idle (inattività): periodicamente la scheda di rete emette un segnale impulsivo (di durata 100 ns) a intervalli di 16 ms detto **LIT** (*Link Integrity Test*): se l'hub

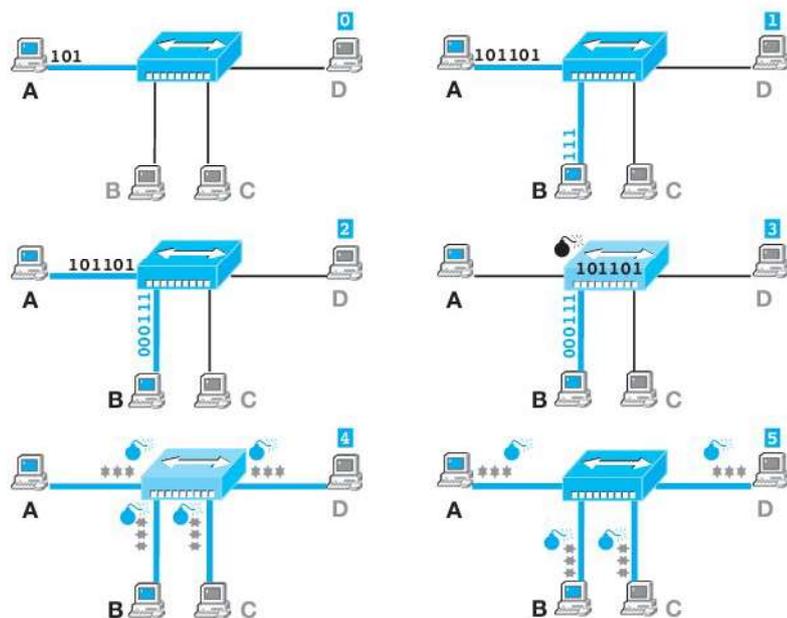
non rileva questo segnale per più di 150 ms blocca la porta ritenendo la scheda ad essa connessa spenta (o guasta).

L'hub estende il dominio di collisione e il dominio di broadcast esattamente come il repeater: quando l'apparato riceve un pacchetto su una porta, lo ritrasmette su tutte le altre.

La circolazione dei pacchetti in un hub avviene secondo il diagramma sottostante, in cui si ipotizza un trasferimento dalla stazione A alla stazione D.



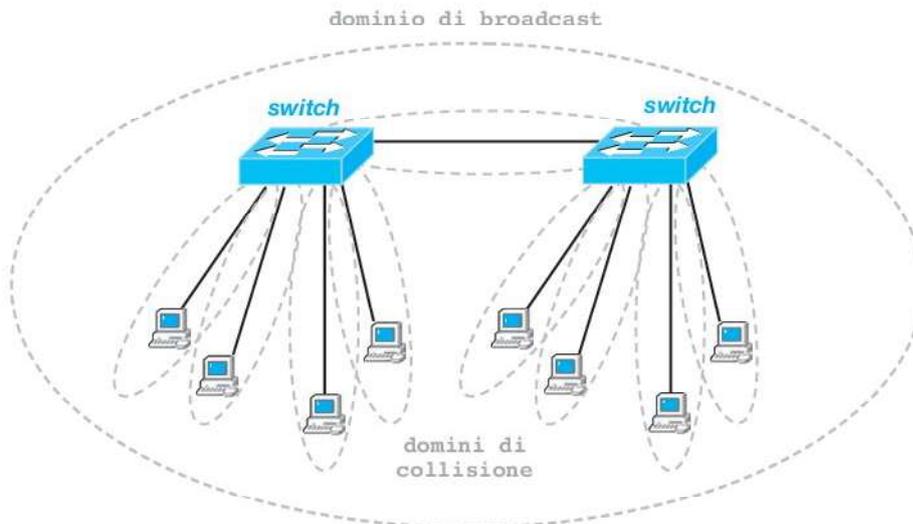
Le collisioni su un hub avvengono ogniqualvolta due pacchetti si trovano contemporaneamente al suo interno.



Se la collisione avviene sul cavo di una stazione, si propagerà all'hub che la reinverrà su tutte le porte meno quella di provenienza.

### 3 Switch

Lo **switch half duplex** sostituisce l'hub come concentratore di **centro stella** e aggiunge una preziosa capacità che opera nel sottolivello MAC dello strato 2 Datalink. **All'interno dello switch è presente una memoria dinamica volatile denominata CAM (Content Addressable Memory)** che mantiene aggiornata una semplice **tabella in cui vengono registrati tutti gli indirizzi MAC mittente dei pacchetti che attraversano le sue porte**. **Ogni riga della tabella (detta **filtering database**) è una terna che prevede l'indirizzo MAC mittente, la porta da cui il pacchetto è transitato e un valore numerico che indica da quanto tempo il valore è presente nella tabella (*ageing time*).**

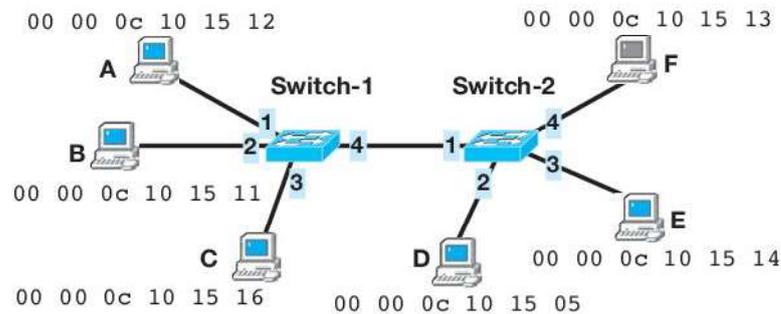


ESEMPIO

#### Filtering database

Dato il seguente schema di switched LAN, mostrare il contenuto della CAM dello Switch-1 di marca Cisco se:

1. L'host A manda un pacchetto all'host B.
2. L'host E manda un pacchetto all'host C.
3. L'host B manda un pacchetto all'host D.
4. L'host D manda un pacchetto all'host C.
5. L'host B manda un pacchetto all'host F (che è spento).



- L'attività 1. mappa l'host **A** sulla porta **1** di Switch-1.  
 L'attività 2. mappa l'host **E** sulla porta **4** di Switch-1.  
 L'attività 3. mappa l'host **B** sulla porta **2** di Switch-1.  
 L'attività 4. mappa l'host **D** sulla porta **4** di Switch-1.  
 L'attività 5. non genera nessuna informazione per Switch-1.

Nella seguente videata è riportato l'output del comando console `show cam dynamic` che mostra il filtering database dello Switch-1:

```
Switch-1#show cam dynamic
* = Static Entry. + = Permanent Entry.
# = System Entry X = Port Security Entry

Dest MAC Address   Ports Age
-----
00-00-0c-10-15-12   1    1
00-00-0c-10-15-14   4    2
00-00-0c-10-15-11   2    3
00-00-0c-10-15-05   4    3
```

Quando un pacchetto MAC si presenta su una porta subisce il **processo di learning**, che aggiorna il filtering database; quindi è sottoposto ad un **processo di forwarding**. Questa volta si analizza l'indirizzo MAC destinatario, si consulta la tabella per cercarne la presenza e, nel qual caso, lo switch inoltra il pacchetto solo sulla porta corrispondente. **Se l'indirizzo MAC destinatario invece non compare nel filtering database il pacchetto viene replicato su tutte le porte dello switch tranne quella di provenienza (flooding).**

In questo modo il numero di pacchetti che circolano sulle stazioni collegate allo switch si riduce drasticamente. Per evitare che alcune righe del filtering database possano diventare obsolete (esempio, quando si cambia l'host su una porta), allo scadere dell'*ageing time* la riga in tabella viene eliminata.

Il filtraggio dei pacchetti sulle porte d'uscita non avviene quando lo switch viene acceso (dato che la CAM è vuota), comportandosi per un certo periodo iniziale dopo l'accensione come un hub, ovvero inoltrando i pacchetti ricevuti su una porta verso tutte le altre porte. Man mano che i pacchetti transitano, la CAM dello switch si riempie e l'operazione di filtraggio si fa sempre più frequente.

Lo switch poi ha anche la proprietà di mantenere separati i domini di collisione: ogni coppia di porte su uno switch rappresenta un dominio di collisione, contrariamente all'hub.

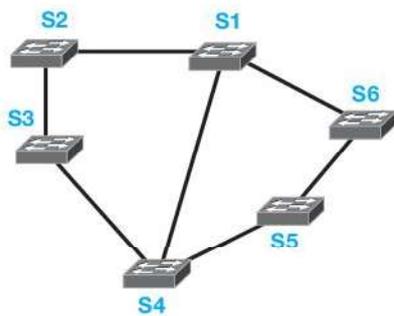
In questo modo i traffici locali alle stazioni collegate al singolo switch rimangono circoscritti al suo interno e il traffico complessivo diviene ottimizzato in maniera radicale, eliminando gran parte delle collisioni inutili e molti pacchetti che non arriverebbero mai a destinazione.

L'introduzione dello switch ha rivoluzionato la tecnologia delle LAN Ethernet, consentendo di poter gestire, su una sigola infrastruttura interconnessa da switch, anche migliaia di stazioni (**XLAN**, *Extended Lan* o anche **switched LAN**).

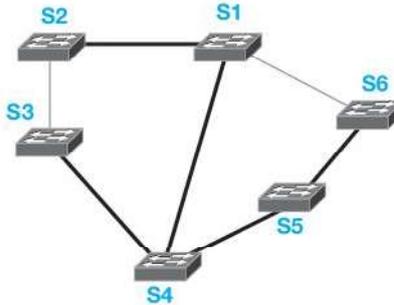
**Il collegamento di più switch è valido, però, solo se la topologia fisica che si ottiene dalla loro interconnessione viene mantenuta *ad albero*, ovvero se viene sempre garantito un unico percorso fisico possibile tra due stazioni qualsiasi della switched LAN. Se così non fosse si corre il rischio che alcuni**

pacchetti MAC (esempio, con indirizzo destinatario non ancora mappato su nessuna porta degli switch) rimangono indefinitamente in circolazione; inoltre i pacchetti di broadcast causerebbero velocemente l'intasamento della XLAN (**broadcast storm**).

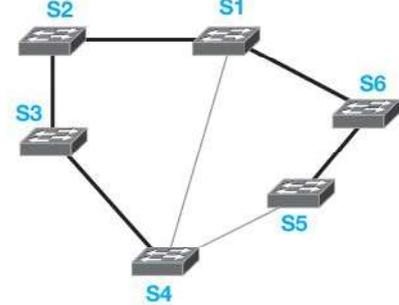
Topologia a maglia



Topologia a albero (1)

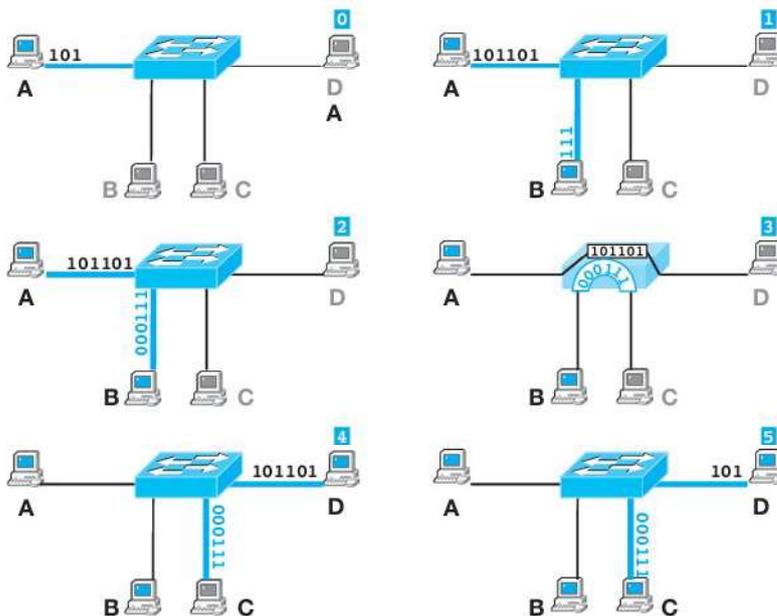


Topologia a albero (2)



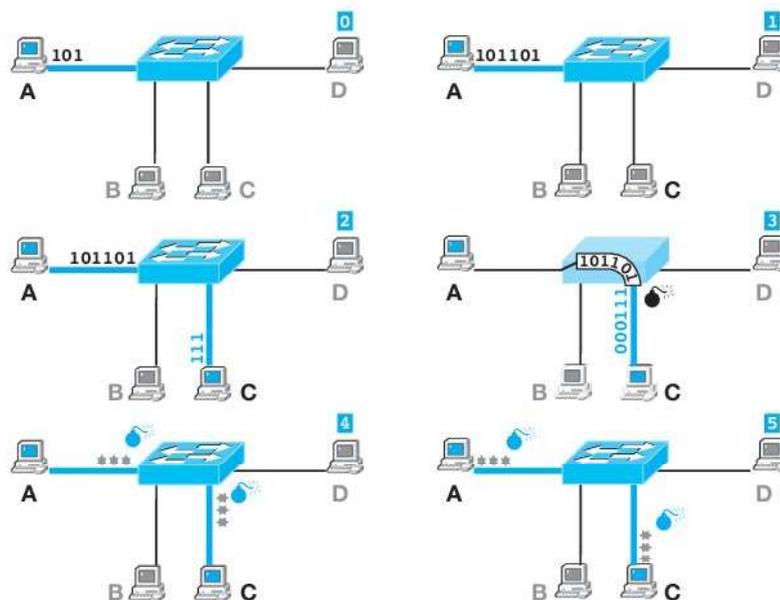
La topologia ad albero per le XLAN induce un potenziale problema di fault tolerance: l'interruzione (o il malfunzionamento) su una qualsiasi tratta di interconnessione tra switch divide in due l'intera rete rendendo le due parti incomunicanti. A questo problema si pone rimedio con switch dotati di funzionalità speciali (*Spanning Tree Protocol*, cfr. *Bridge*).

La circolazione dei pacchetti in uno switch avviene secondo il diagramma sottostante, in cui si ipotizza un trasferimento dalla stazione A alla stazione D e dalla stazione B alla stazione C.



Le collisioni su switch half duplex possono verificarsi nelle seguenti modalità:

- Quando una stazione sta trasmettendo e, prima che termini, le giungono trame in ricezione. Rispetto ad un hub, questa eventualità è decisamente più infrequente: avviene solo se la stazione che invia è la stessa che deve ricevere (mentre sull'hub la collisione avviene per qualsiasi coppia di stazioni).



- Quando due trasmissioni provenienti da stazioni differenti devono giungere alla medesima stazione «contemporaneamente» (cioè all'interno dell'intervallo di tempo RTD delle rispettive trame). Questa evenienza può essere contrastata dalla presenza di un **buffer di transito** che sequenzia i due flussi sulla medesima porta separandone l'RTD.



### Filtering database

Data la seguente attività di rete su uno switch a 8 porte, scrivere un programma in linguaggio C che, per ogni pacchetto in entrata su una porta P, indichi la porta di inoltro. Mostrare il filtering database.

| P | MAC Dest          | MAC Source        | P | MAC Dest          | MAC Source        |
|---|-------------------|-------------------|---|-------------------|-------------------|
| 1 | 00 07 0d af f4 54 | 08 00 37 15 e6 bc | 2 | ff ff ff ff ff ff | 08 00 37 15 e6 bc |
| 2 | 08 00 37 15 e6 b1 | 00 12 3f 4a 33 d2 | 8 | 00 12 3f 4a 33 d2 | 00 12 3f 76 12 1° |
| 3 | 08 00 37 15 e6 bc | 00 00 c0 9f a0 97 |   |                   |                   |

### OUTPUT

```
Traffico
.....
P MAC Dest      MAC Source
1;00 07 0d af f4 54;08 00 37 15 e6 bc
2;08 00 37 15 e6 b1;00 12 3f 4a 33 d2
3;08 00 37 15 e6 bc;00 00 c0 9f a0 97
2;ff ff ff ff ff ff;08 00 37 15 e6 bc
8;00 12 3f 4a 33 d2;00 12 3f 76 12 1a

Learning e forwarding
.....
Trama ricevuta su 1 -> broadcast su 2 3 4 5 6 7 8
Trama ricevuta su 2 -> broadcast su 1 3 4 5 6 7 8
Trama ricevuta su 3 -> 1
Trama ricevuta su 2 -> broadcast su 1 3 4 5 6 7 8
Trama ricevuta su 8 -> 2

Filtering database
.....
P: 1 MAC: 08 00 37 15 E6 BC
P: 2 MAC: 00 12 3F 4A 33 D2
P: 3 MAC: 00 00 C0 9F A0 97
P: 2 MAC: 08 00 37 15 E6 BC
P: 8 MAC: 00 12 3F 76 12 1A
```

Il codice in linguaggio C:

```

00 #include <stdio.h>
01 #include <string.h>
02 #include <stdlib.h>
03 #include "htoi.inc" // funzione htoi()
04
05 typedef struct {
06     int porta;
07     unsigned char MACd[6];
08     unsigned char MACs[6];
09 } TRAFFICO; TRAFFICO* pt0; TRAFFICO* pt;
10
11 typedef struct {
12     int porta;
13     unsigned char MAC[6];
14     int age;
15 } FILTERINGDB; FILTERINGDB* pf0; FILTERINGDB* pf;
16
17 int FromMACtoByte(char* szmac, int offs, int idx);
18 int isMACequal(unsigned char* mac1, unsigned char* mac2);
19
20 int main (void)
21 {
22     FILE *fp;
23     int i,j, nT, nF;
24     char szRiga[80];
25
26     printf("Traffico\n.....\n");
27     i = 0;
28     fp = fopen ("traffic.txt", "r");
29     fgets(szRiga, 80, fp); printf("%s",szRiga);
30     while ( fgets(szRiga, 80, fp) != NULL )
31     {
32         printf("%s",szRiga);
33         if (szRiga[0])
34         {
35             if (i==0) { pt0 = malloc(sizeof(TRAFFICO)); pt = pt0;}
36             else { pt0 = realloc(pt0, (i+1)*sizeof(*pt0)); pt = pt0+i;}
37
38             pt->porta = szRiga[0]-'0';
39             for (j=0; j<6;j++)
40             {
41                 pt->MACd[j]=(unsigned char)FromMACtoByte(szRiga, 2, j);
42                 pt->MACs[j]=(unsigned char)FromMACtoByte(szRiga, 20, j);
43             }
44             i++; nT = i;
45         }
46     }
47     fclose (fp);
48
49     printf("\nLearning e forwarding\n.....");
50     nF = 0;
51     for (i=0;i<nT;i++)
52     {
53         int trovato;
54
55         trovato = 0;
56         for (j=0;j<nF;j++)
57         {
58             if ((pf0+j)->porta==(pt0+i)->porta)
59             {
60                 if (isMACequal((pf0+j)->MAC, (pt0+i)->MACs))
61                 {
62                     trovato = 1;

```

```

63         break;
64     }
65 }
66 }
67
68 if (!trovato)
69 {
70     if (nF==0) { pf0 = malloc(sizeof(*pf0)); pf = pf0;}
71     else { pf0 = realloc(pf0, (nF+1)*sizeof(*pf0)); pf = pf0+nF;}
72     nF++;
73
74     pf->porta = (pt0+i)->porta;
75     memcpy(pf->MAC, (pt0+i)->MACs, sizeof(pf->MAC));
76 }
77
78 trovato = 0;
79 for (j=0;j<nF;j++)
80 {
81     if (isMACequal((pf0+j)->MAC, (pt0+i)->MACd))
82     {
83         printf("\nTrama ricevuta su %d -> %d", (pt0+i)->porta, (pf0+j)->porta);
84         trovato = 1;
85     }
86 }
87
88 if (!trovato)
89 {
90     printf("\nTrama ricevuta su %d -> broadcast su ", (pt0+i)->porta);
91     for (j=1; j<9; j++)
92         if (j!=(pt0+i)->porta) printf("%d ",j);
93 }
94 }
95
96 printf("\n\nFiltering database\n.....\n");
97 for (i=0;i<nF;i++)
98 {
99     printf("\nP: %d MAC: %02X %02X %02X %02X %02X %02X", (pf0+i)->porta,
100         (pf0+i)->MAC[0], (pf0+i)->MAC[1], (pf0+i)->MAC[2],
101         (pf0+i)->MAC[3], (pf0+i)->MAC[4], (pf0+i)->MAC[5]);
102 }
103
104 free(pt0);
105 free(pf0);
106 return 0;
107 }
108
109 int FromMACtoByte(char* szmac, int offs, int idx)
110 {
111     char szByte[3];
112
113     szByte[0]=szmac[offs+idx*3];szByte[1]=szmac[offs+idx*3+1];szByte[2]=0;
114     return htoi(szByte);
115 }
116
117 int isMACequal(unsigned char* mac1, unsigned char* mac2)
118 {
119     int i;
120
121     for (i=0;i<6;i++)
122     {
123         if (*mac1!=*mac2) return 0;
124         mac1++; mac2++;
125     }
126     return 1;
127 }

```

Il codice fa uso di due strutture dati per descrivere il traffico assegnato, letto da un file di testo **traffic.txt** (righe 5÷9) e il filtering database che verrà costruito in base ad esso (righe 11÷15).

Nel blocco di codice tra **riga 28** e **riga 47** viene letto il file di testo nel quale, su ogni riga, è memorizzato il traffico dato (in totale 6 righe, la prima riga è l'intestazione).

Nel secondo blocco di codice tra **riga 49** e **riga 94** vengono analizzate ordinatamente le 5 attività, costruito il filtering database (solo se l'attività fornisce una informazione valida) e decisa la porta di inoltramento.

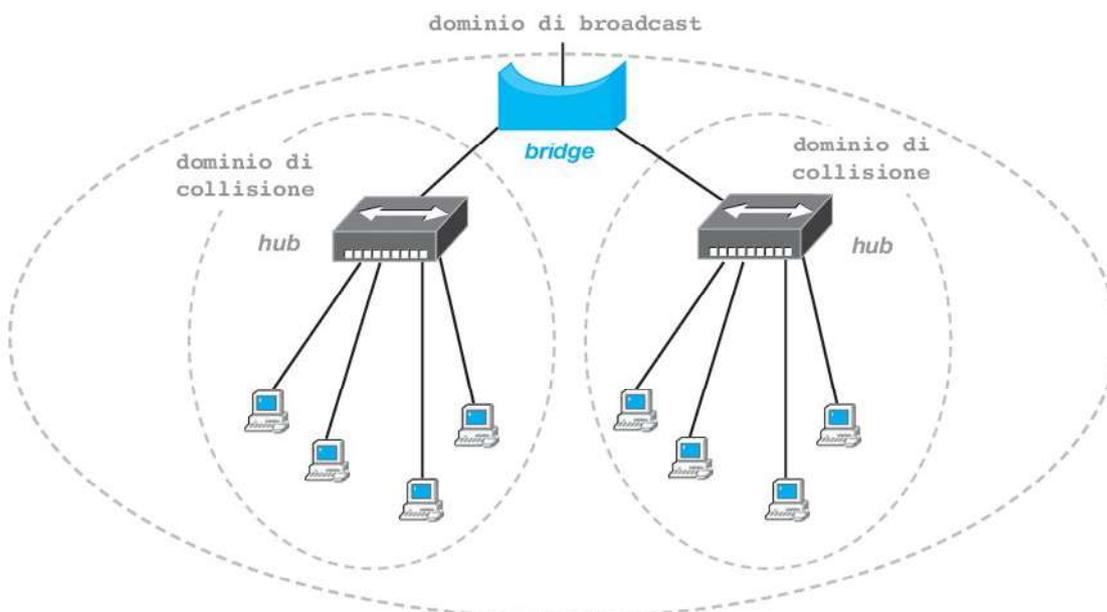
Infine nel terzo blocco di codice (**righe 96÷102**) viene stampata sullo schermo la sequenza delle regole che alla fine dell'attività si sono accumulate (filtering database).

Il programma usa la funzione *htoi()* (**riga 3**) per passare da esadecimale a decimale; una funzione *fromMAC-tobyte()* per estrarre il singolo byte da una riga di testo e una funzione *isMACequal()* per confrontare due indirizzi MAC.

Con i dati forniti si nota che un'attività non genera un'informazione già presente nel filtering database; inoltre se per un'attività non si individua la porta di inoltramento viene stampato l'inoltramento del broadcast, compresa la destinazione di broadcast classica (ff ff ff ff ff).

## 4 Bridge

Il **bridge** opera esattamente come uno switch. In realtà il bridge ha un numero limitato di porte, non dovendo sostenere host ma solo collegare altri bridge. Si tratta di un **dispositivo legacy** e non più diffuso nelle XLAN, sostituito ovunque dallo switch. Il bridge però è stato il **primo dispositivo che ha risolto il problema dell'interruzione di un ramo di XLAN introducendo fault tolerance nelle reti switched.**



A differenza dello switch classico, il bridge è in grado di gestire (aprire o chiudere) le sue porte di interconnessione utilizzando un protocollo specifico che opera in modo distribuito tra i vari bridge interconnessi. In questo modo è possibile collegare i bridge senza garantire una struttura ad albero della topologia fisica della rete, ma ricercando invece una struttura

a maglia (incompleta) che preveda più cammini possibili tra due qualsiasi stazioni.

Il protocollo operante automaticamente tra i bridge riesce infatti a ricostruire una struttura ad albero all'interno della topologia a maglia, chiudendo opportunamente le porte ridondanti. In questo modo se un collegamento tra bridge si interrompe a causa di un guasto, il protocollo agisce automaticamente e, in breve tempo, riottiene una nuova struttura ad albero funzionante, inibendo le porte coinvolte nel guasto.

L'algoritmo per la riduzione ad albero (di una rete ridondante a maglia) è detto di *Spanning Tree*, e il protocollo adottato dai bridge **STP** (*Spanning Tree Protocol*, IEEE 802.1D). In seguito STP è stato adottato dagli switch di fascia alta, migliorato e ampliato, cosicché oggi si possono trovare switch che adottano anche contemporaneamente algoritmi di riduzione ad albero come **RSTP** (*Rapid STP* o 802.1w) e **MSTP** (*Multiple STP* o 802.1s), auto-negoziando il protocollo da eseguire.

## 4.1 Spanning Tree Protocol

Il protocollo STP si colloca, a rigore, a livello 3 Network, dato che si occupa dell'instradamento di pacchetti e opera in maniera distribuita tra i vari nodi bridge tramite speciali trame di servizio denominate **BPDU** (*Bridged PDU*). Siccome però le operazioni dello STP avvengono in ambito locale e in autonomia, la letteratura pone i bridge a livello 2 Datalink, benché abbiano capacità di livello 3 Network.

Il traffico di BPDU tra i bridge è trasparente e automatico, veicolato con trame 802.3 LLC multicast aventi *Protocol type* pari a 42h.

Ogni porta del bridge possiede un indirizzo MAC e il bridge stesso adotta come MAC address quello della sua prima porta. Inoltre ogni bridge possiede un valore di **priorità**, impostabile via management o acquisito di default. Anche ogni porta assume un valore di priorità e un valore di **costo**, allo stesso modo. Con questi dati a disposizione e facendo circolare le BPDU, STP riesce a ridurre una topologia a maglia in una topologia ad albero, bloccando le porte ridondanti in poche decine di secondi.

Quando la topologia si modifica, per esempio per l'aggiunta di un nuovo bridge o a causa della rilevazione di un guasto su un collegamento, STP si riavvia automaticamente e ricalcola una nuova topologia ad albero.

L'algoritmo agisce secondo le fasi seguenti:

- 1) Elezione del **root bridge**. Ogni bridge immette la propria priorità in una BPDU e la fa circolare in multicast; ricevendo varie BPDU da altri bridge confronta la priorità ricevuta con la propria e, se minore, smette di inviare BPDU. Ben presto rimarrà un solo bridge trasmittente, quello con la priorità più bassa, il root bridge eletto.
- 2) Impostazione della **root port**. Ogni bridge (tranne il root bridge) deve selezionare una delle sue porte come quella che ora dovrà raggiungere il root bridge appena eletto. Continuando a ricevere su ogni

porta le BPDU del root bridge, ricava da esse il **root path cost** di ogni sua porta. Quella con valore minore viene impostata come root port. D'ora in avanti solo le BPDU ricevute da una root port saranno inoltrate.

- 3) Impostazione delle **designated port**. Ogni bridge (compreso il root bridge) deve infine selezionare tra le sue porte (tranne la root port) quelle che devono rimanere aperte (designated port). Come prima, seleziona quelle con root path cost minore o nullo. Tutte le porte rimanenti saranno chiuse (**blocked**).

Una volta raggiunta la topologia ad albero, il root bridge continua a emettere periodicamente BPDU e anche le porte bloccate continuano a riceverle.

Questo serve affinché possa essere autorilevato un cambiamento della topologia, per esempio quando un nuovo switch viene aggiunto, oppure uno si spegne, oppure qualche porta si guasta.

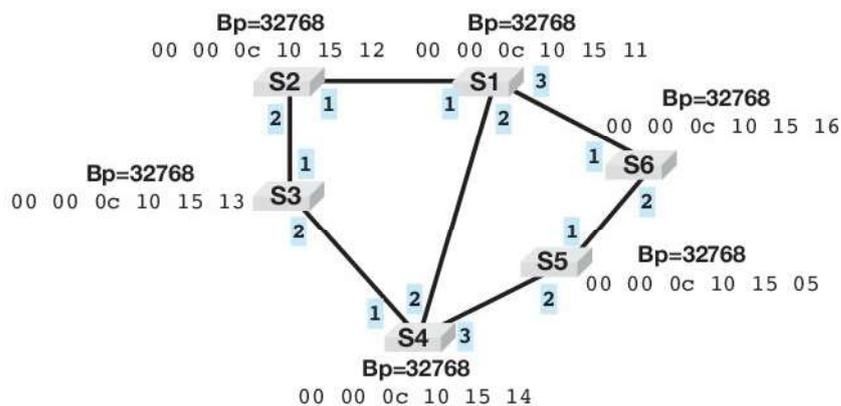
Lo spegnimento di uno switch e il malfunzionamento di un link causano un timeout sulla ricezione delle BPDU del root bridge per almeno una porta di uno switch ancora attivo. Lo switch in timeout emette in multicast una BPDU speciale detta **Topology Change**. Ciò riavvierà lo STP e ogni bridge dovrà resettare il proprio filtering database.

Se invece un nuovo switch viene aggiunto, tenterà l'elezione del root bridge emettendo una BPDU. Il root bridge, ricevendo una BPDU non emessa da lui, avvierà la fase di Topology Change e il successivo riavvio di STP.

**ESEMPIO**

**Spanning tree**

Dato il seguente schema di switched LAN a maglia, considerando pari a 10 il costo di ogni porta di ogni switch, applicare l'algoritmo STP per ottenere una topologia ad albero.



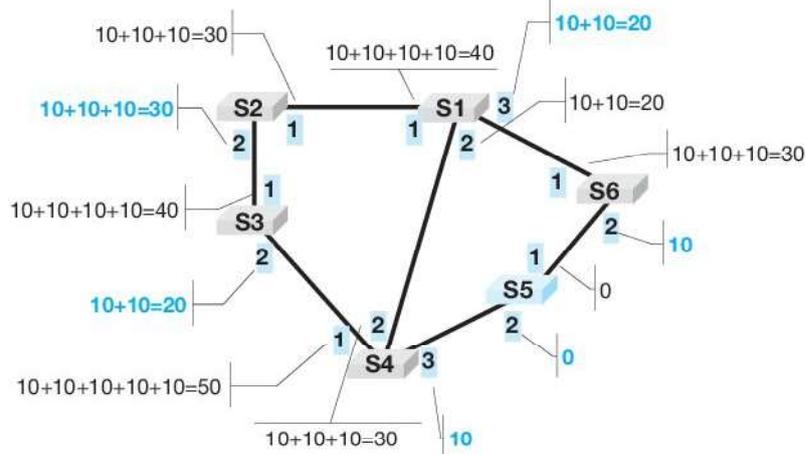
L'elezione del root bridge è abbastanza agevole.

Inizialmente ogni switch emana una BPDU con la propria priorità. Si nota che tutti gli switch hanno priorità di default (32768), ovvero quella ottenuta alla loro accensione.

Quando le priorità coincidono, si passa all'indirizzo MAC: il minore vince l'elezione. In questo caso si tratta dello switch **S5**, che assume il ruolo di **root bridge**.

Ora si passa ad analizzare, per ogni porta di ogni switch, il **root path cost**, ovvero il valore che si ottiene, partendo da una porta specifica, raggiungendo **S5** e sommando, ad ogni «salto» il costo della porta (costo 10 per tutte).

Per esempio la porta **1** di **S1** raggiunge **S5** con quattro «salti» (**S2-S3-S4-S5**), quindi il suo root path cost vale 40. Per ogni porta va ripetuto il calcolo. Le porte del root bridge hanno ovviamente root path cost uguale a zero. Ora, per ogni switch si può dedurre la **root port**: è la porta con il root path cost minore.



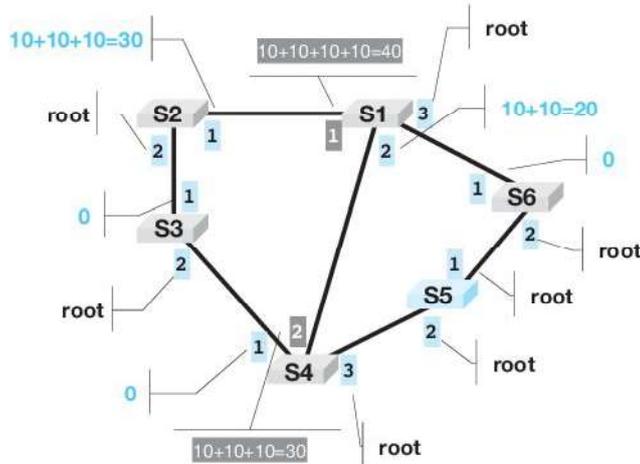
Ora bisogna scegliere le **designated port** e le **blocked port**.

Avendo a disposizione il root bridge e tutte le root port, bisogna ricordare che ora le BPDU partono solo dal root bridge e ogni switch le inoltra solo se le riceve dalla sua root port. Per esempio, alla porta **1** di **S6** nessuna BPDU potrà mai arrivare (provare, ricordando che le BPDU ora partono solo da **S5** e si propagano solo se ricevute da una root port).

Stesso discorso per la porta **1** di **S4** e la **1** di **S3**. Queste porte avranno root path cost = 0.

Le altre porte invece vengono raggiunte dalle BPDU.

Per esempio, la porta **2** di **S1** con il percorso **S5-S4-S1**: avrà root path cost = 20. Terminata l'analisi delle porte, lo schema diviene:

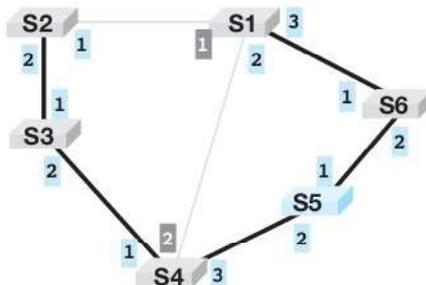


Le porte con root path cost=0 sono sicuramente designate.

Su **S1** invece va designata la porta **2**: ha il minor root path cost. La porta **1** di **S1**, di conseguenza, va bloccata.

Su **S4** la porta bloccata sarà la **2** (root path cost superiore).

L'algoritmo ha raggiunto lo scopo, la nuova topologia ad albero è la seguente:

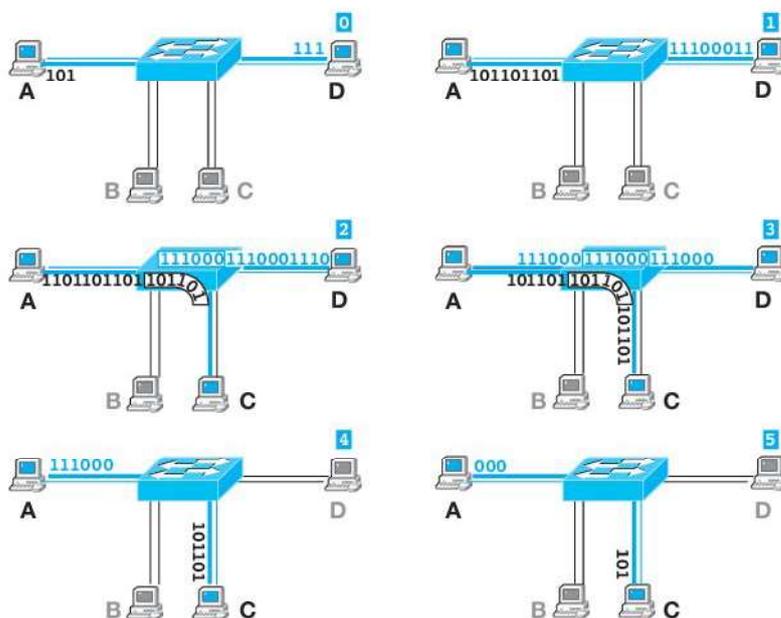


## 5 Switch full duplex

Gli **switch full duplex** sono stati introdotti nel 1997 con lo standard IEEE 802.3x. Switch e NIC conformi a questo standard **non generano più collisioni**, ovvero il meccanismo base di Ethernet, il **CSMA/CD**, viene **disabilitato**.

Infatti se le stazioni possono trasmettere e ricevere contemporaneamente e lo switch altrettanto sulla porta corrispondente, il mezzo non è più condiviso ma dedicato. A questo punto anche la distanza tra due nodi Ethernet full duplex non ha i limiti dettati da CSMA/CD e dipende solo dalle caratteristiche del mezzo trasmissivo.

Questa tecnologia è anche detta *Ethernet flow control*.



### Protocolli vendor neutral

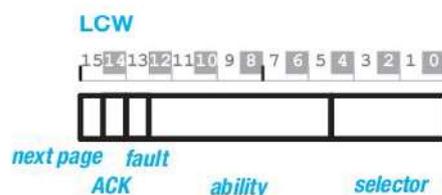
Alcuni protocolli IEEE si dicono **vendor-neutral** perché, pur descrivendo una particolare funzionalità di rete, non dettano specifiche stringenti di implementazione, ma solo linee guida. In questo modo i produttori sono liberi di realizzare una versione proprietaria del protocollo, imponendo l'uso dei propri prodotti.

LLDP (Link Layer Discovery Protocol, IEEE 802.1AB) è uno di questi. A livello Datalink ha il compito di normare proprio l'autonegoziamento degli apparati di rete.

### 5.1 Autonegoziamento

La transizione verso Ethernet flow control è graduale, pertanto gli switch full duplex devono spesso adattare qualche porta a tecnologie half duplex.

Per questo motivo gli switch adottano un meccanismo di **autonegoziamento** attivo su ogni porta che deriva dal LIT di 10BASE T e lo estende. In assenza di traffico, la scheda invia periodicamente un treno di impulsi codificati (**NLP**, *Normal Link Pulses*) che comunicano alla porta dello switch alcune informazioni su 16 bit (**LCW**, *Link Code Word*).



Il campo *ability* codifica proprio il tipo di supporto della scheda che emette il segnale:

- bit 5, 10BASE-T;
- bit 6, 10BASE-T full duplex;
- bit 7, 100BASE-TX;
- bit 8, 100BASE-TX full duplex;
- bit 9, 100 BASE-T4;
- bit 10, *pause*; bit 11, *asymmetric pause*; bit 12, *reserved*.

In questo modo lo switch può sostenere sia NIC full duplex che NIC half duplex consentendo l'interoperabilità dei dispositivi di rete.

L'LCW consiste in una serie di impulsi a frequenza regolare che occupano in totale 2 ms: durante questo tempo la scheda può emettere al minimo 17 impulsi (per 16 bit a 0) o al massimo 33 impulsi (per 16 bit a 1).

**ESEMPIO**

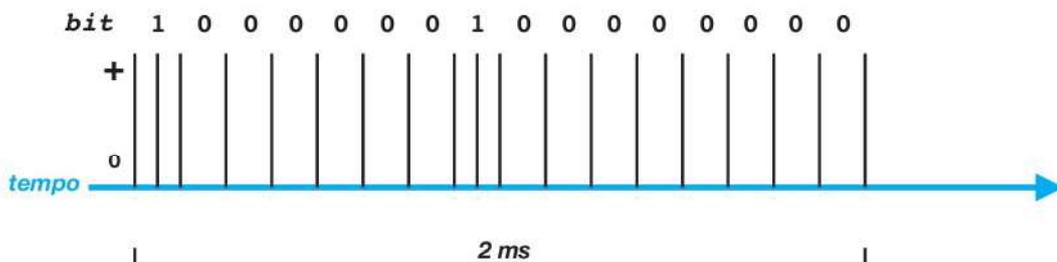
### Autonegoziazione

Mostrare la codifica impulsiva di un'autonegoziazione NLP per un NIC 100BASE-TX full duplex.

La codifica impulsiva dei 16 bit LCW avviene entro un intervallo di 2 ms (millisecondi).

Un bit a **1** viene trasmesso con un impulso a metà del bit time dedicato al bit **0**.

La codifica segue la prassi little endian (il primo bit trasmesso ha peso 0 e l'ultimo ha peso 15).



I due bit a 1 presenti nella sequenza in figura (di peso 8 e di peso 15) significano rispettivamente 100 BASE-TX e next page.



**PROGRAMMA**

### Autonegoziazione

Scrivere un programma in linguaggio C che, presa in ingresso una stringa binaria di 16 caratteri, stampi la codifica impulsiva dell'LCW corrispondente.

```
LAYOUT
Inserire LCW (16 char binari): 10000010000000
LCW in decimale (big endian): 33024
1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
| | | | | | | | | | | | | | | |
```

Un programma in linguaggio C potrebbe essere:

```
00 #include <stdio.h>
01 #include <conio.h>
02 #include <string.h>
03
04 #define LITTLEENDIAN (1)
05 #define BIGENDIAN (0)
06
07 int BinarytoDec(char* szbinary, int ile);
```

```

08
09 int main (void)
10 {
11     char szLCW[17];
12     char ch;
13     int i = 0;
14
15     printf("Inserire LCW (16 char binari): ");
16     while (i!=16)
17     {
18         ch = getche();
19         if ((ch=='0') || (ch=='1'))
20         {
21             szLCW[i] = ch; szLCW[i+1]=0;
22             i++;
23         }
24     }
25
26     i = BinarytoDec(szLCW, BIGENDIAN);
27     printf("\nLCW in decimale (big endian): %d\n",i);
28
29     for (i=strlen(szLCW)-1; i>=0; i--)
30     {
31         if (szLCW[i]=='0') printf("0 ");
32         else printf("1 ");
33     }
34     printf("\n");
35
36     for (i=strlen(szLCW)-1; i>=0; i--)
37     {
38         if (szLCW[i]=='0') printf("| ");
39         else printf("||");
40     }
41
42     return 0;
43 }
44
45 int BinarytoDec(char* szbinary, int ile)
46 {
47     int iPos, iRv = 0;
48
49     if (ile==0) strrev(szbinary); // big endian
50
51     for (iPos=strlen(szbinary)-1; iPos>=0; iPos--)
52     {
53         iRv = iRv << 1;
54         if (szbinary[iPos]=='1') iRv += 1;
55     }
56
57     return iRv;
58 }

```

Dopo la sezione di input dei 16 caratteri binari (**righe 16÷24**), viene stampato il valore dell'LCW in decimale, interpretando la sequenza di bit in big endian con la funzione *szBinarytoDec()* alla **riga 27**.

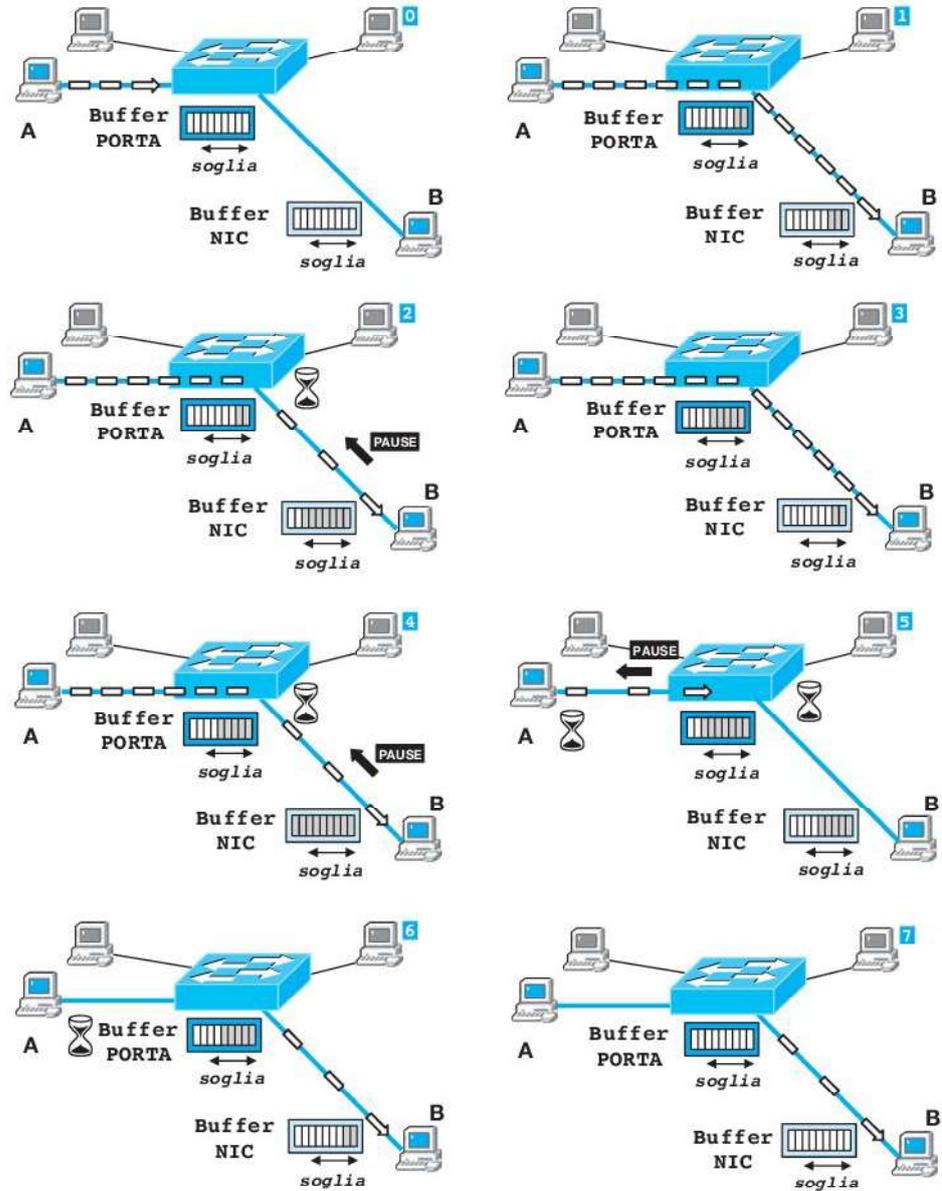
Quindi si stampa la sequenza binaria dell'LCW (**righe 29÷33**) e la relativa coppia di caratteri grafici "I " o "||" a seconda del valore del bit corrispondente (0 o 1) nel blocco tra le **righe 36÷40**.

## 5.2

### MAC control

Negli switch full duplex vengono evitate le collisioni, prevedendo due canali fisici separati per ricezione e trasmissione. Due o più flussi contemporanei diretti verso un'unica porta/stazione sono regolati da appositi buffer di memorizzazione, sia sulla scheda, sia su ogni porta dello switch. Quando i buffer sono pieni (per esempio, in presenza di un traffico di broadcast o nel caso di

porte con diverso bitrate), gli switch e le NIC full duplex utilizzano un protocollo speciale denominato **MAC control** per inviare ai nodi trasmettenti particolari pacchetti di servizio (PAUSE) che li mettono in pausa temporanea.



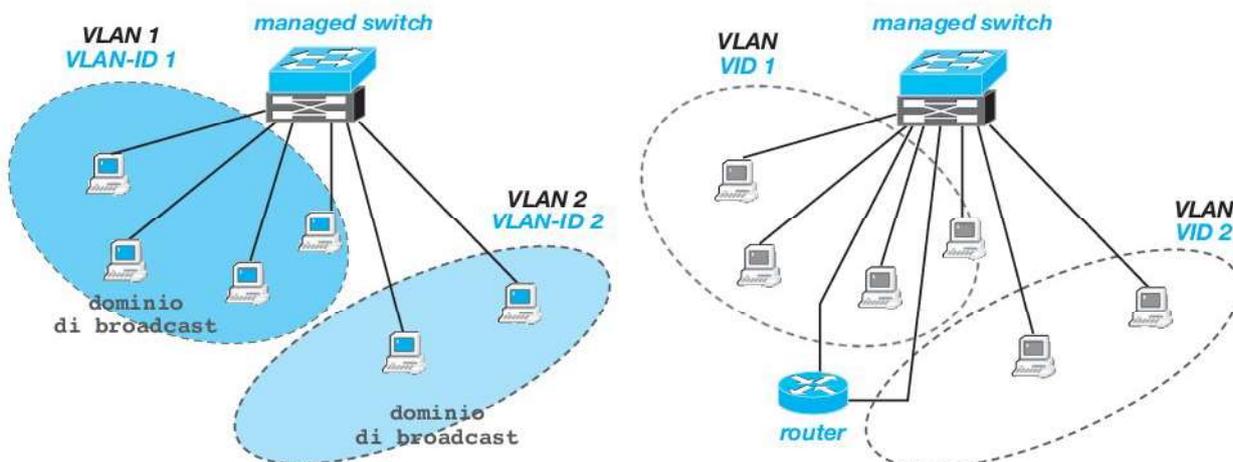
Il MAC control è standardizzato nel documento IEEE 802.3x e viene veicolato con pacchetti Ethernet aventi Ethertype pari a 8808h. È quindi un protocollo completamente trasparente.

In definitiva negli switch full duplex mancano sì le collisioni, ma può comunque originarsi una latenza a causa dei pacchetti di PAUSE.

## 6 Managed switch (VLAN)

Quando una switched LAN diventa molto ampia, benché con domini di collisione ben partizionati, rimane comunque vulnerabile all'intenso traffico di broadcast che non viene trattenuto dagli switch.

Inoltre è spesso desiderabile che insiemi (logici) di stazioni non abbiano nulla a che fare con altri insiemi di stazioni (esempio, i personal computer dell'amministrazione con i personal computer della progettazione di un'azienda), cercando di ottenere che ogni singolo pacchetto circolante tra uno di questi insiemi non raggiunga mai una stazione dell'altro insieme, se non attraverso una precisa regola impostata a priori.



In questi casi si può utilizzare una tecnica di partizionamento delle stazioni denominata **VLAN** (*Virtual LAN*, IEEE 802.1Q) basata su **managed switch**, ovvero switch che devono essere configurati specificando alcuni parametri operativi per ognuna delle porte di cui sono dotati.

Il traffico di servizio generato dalle VLAN è trasportato da frame Ethernet 2 con *Protocol type* pari a 8100h, attivo fondamentalmente allorché si connettono tra di loro più managed switch affinché gli stessi possano operare con le stesse regole condivise.

In un managed switch si possono impostare le  $n$  porte dell'apparato come facenti parte di  $m$  insiemi (in questo caso  $m$  VLAN) in modo tale che le stazioni connesse a quelle porte possano appartenere di fatto a  $m$  LAN differenti, con domini di broadcast separati.

Con un'operazione di configurazione manuale (management), prima si creano le VLAN sullo switch, definendole con un identificativo numerico progressivo (**VLAN-ID**), quindi ogni porta può essere assegnata ad una VLAN. In questo caso la porta dello switch è detta **access port**.

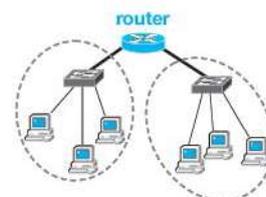
Nel caso si volesse mettere in contatto due (o più) VLAN, è necessario connettere alle VLAN un **router** che, operando a livello 3, analizza e inoltra i pacchetti tra le VLAN con regole esplicite (**inter-VLAN routing**).

Se l'estensione della switched LAN è molto ampia, può essere il caso di dover connettere tra di loro più managed switch e quindi ottenere una serie di VLAN per ogni managed switch utilizzato. Si consideri che su diversi managed switch possono, e spesso devono, essere presenti anche le stesse VLAN.

In questi casi i managed switch coinvolti devono avere configurata almeno una porta in modalità **trunk**, la porta di collegamento (**tagged port**). Attraverso questa porta transiteranno quei pacchetti che vanno da

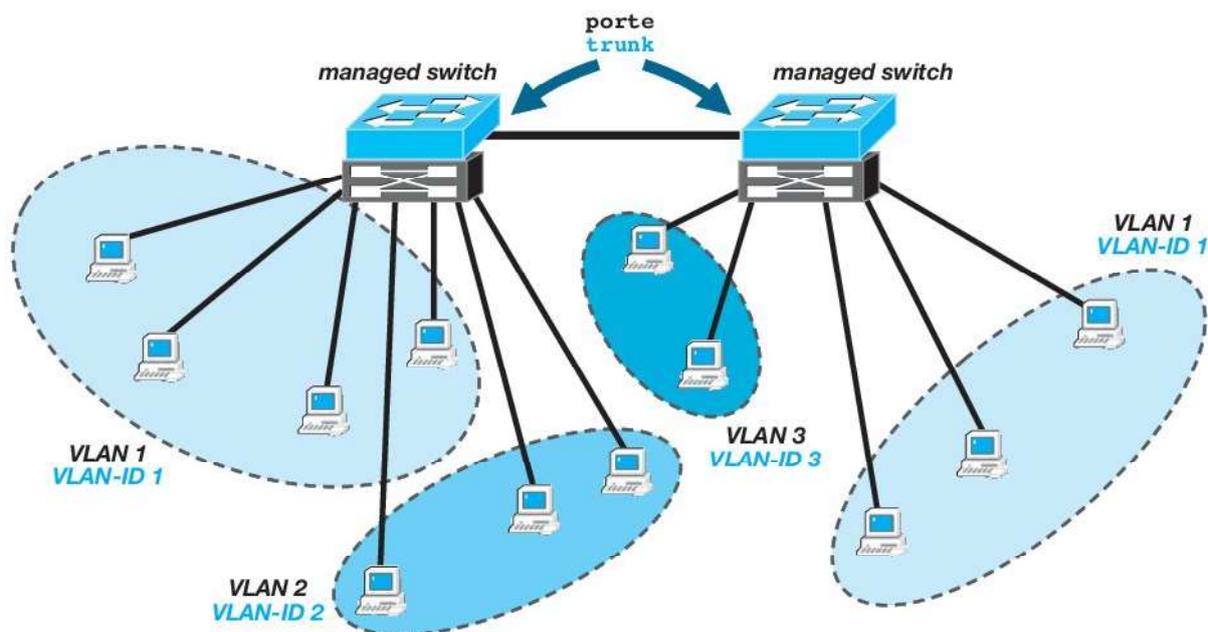
### Switch, router e broadcast

Il modo classico per separare i domini di broadcast su una LAN è utilizzare un **router** come dispositivo di connessione tra due (o più) sottoreti:

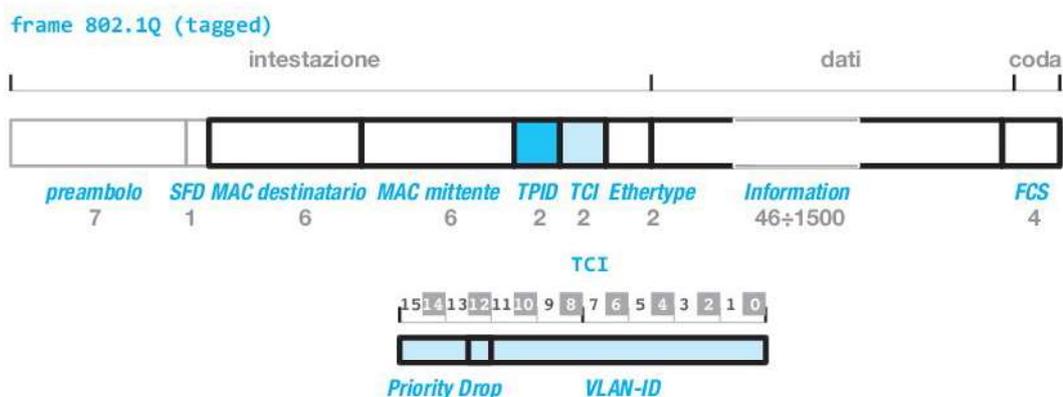


Il router però è un apparato che opera a livello 3 OSI (Network) e non a livello 2 Data link e, come tale, necessita di una configurazione esplicita.

Per separare i domini di broadcast e avere contemporaneamente funzionalità di routing, sono molto diffusi in ambienti *enterprise* (reti complesse) anche gli **switch layer 3**, normalmente più veloci nello smistamento dei pacchetti di un router classico, ma senza la possibilità di connessioni a reti WAN.



una VLAN di uno switch ad un'altra VLAN (spesso la stessa) di un altro switch. I pacchetti che transitano sulle porte trunk vengono ricreati prima dell'inoltro, aggiungendo all'header del frame Ethernet originale quattro byte<sup>(NB)</sup> che contengono il VLAN-ID della VLAN a cui appartiene quel pacchetto. In questo modo lo switch ricevente sarà in grado di inoltrare il frame originale sulla corretta VLAN di destinazione, rimuovendone l'intestazione di trunking.



Il campo *TPID* contiene sempre l'*Ethertype* 8100h in modo tale che i dispositivi in lettura della trama possano immediatamente essere in grado di decodificare il campo *TCI* che contiene l'id. numerico della VLAN a cui appartiene il pacchetto.

NB. L'aggiunta di 4 byte al frame Ethernet potrebbe causare il superamento del limite di dimensione per il pacchetto (esempio, superare 1518 bytes totali passando a 1522 bytes), situazione formalmente non consentita per il MAC di Ethernet.

## Impostazione VLAN

Impostare su un managed switch Cisco tre VLAN con 1, 2 e 18 porte, quindi veicolare la VLAN più numerosa su una porta trunk e tutte le VLAN su una seconda porta trunk.

Nella seguente videata è riportata la sequenza di comandi per creare una VLAN (VLAN-ID=10 e nome **itispr**) su uno switch di marca Cisco di nome **Switch-1**:

```
Switch-1 (config)#vlan 10
Switch-1 (config-vlan)#name itispr
```

Ora i comandi per creare altre due VLAN (VLAN-ID 20 e 30, **uffici** e **test**):

```
Switch-1 (config)#vlan 20
Switch-1 (config-vlan)#name uffici
Switch-1 (config-vlan)#exit

Switch-1 (config)#vlan 30
Switch-1 (config-vlan)#name test
```

Quindi l'assegnazione delle porte alle tre VLAN. Prima le due porte **22** e **23** sulla VLAN **test** (VLAN-ID=30):

```
Switch-1 (config)#interface fa0/22
Switch-1 (config-if)#switchport access vlan 30
Switch-1 (config-if)#exit

Switch-1 (config)#interface fa0/23
Switch-1 (config-if)#switchport access vlan 30
```

Quindi, in modo compatto, le altre porte alle VLAN **uffici** e **itispr**:

```
Switch-1 (config)#interface range fa0/19-21
Switch-1 (config-if)#switchport access vlan 20
Switch-1 (config-if)#exit

Switch-1 (config)#interface range fa0/1-18
Switch-1 (config-if)#switchport access vlan 10
```

La configurazione dello switch può essere mostrata con il comando **show**:

```
Switch-1#show vlan brief
VLAN Name          Status Ports
----
1    default         active Fa0/24, Gi0/1, Gi0/2
10   itispr          active Fa0/1, Fa0/2, Fa0/3, Fa0/4,
    Fa0/5, Fa0/6, Fa0/7, Fa0/8,
    Fa0/9, Fa0/10, Fa0/11, Fa0/12,
    Fa0/13, Fa0/14, Fa0/15, Fa0/16,
    Fa0/17, Fa0/18
20   uffici           active Fa0/19, Fa0/20, Fa0/21
30   test            active Fa0/22, Fa0/23
```

Infine si impostano le due porte Giga Ethernet come porte di **trunk** e si assegnano ad esse le VLAN che devono veicolare:

```
Switch-1 (config)#interface Gi0/1
Switch-1 (config-if)#switchport mode trunk
Switch-1 (config-if)#switchport trunk allowed vlan add 10
Switch-1 (config-if)#exit

Switch-1 (config)#interface Gi0/2
Switch-1 (config-if)#switchport mode trunk
Switch-1 (config-if)#switchport trunk allowed vlan all
```

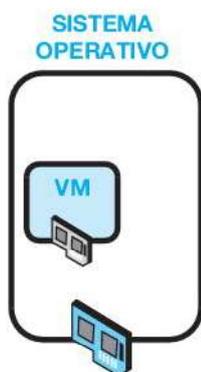
Si nota che un managed switch non ancora configurato possiede una VLAN con id=1 detta di **default** che raggruppa tutte le porte.

## 7 Switch virtuali

Lo **switch virtuale**, a differenza dello switch classico, è un oggetto software, cioè un programma che emula il funzionamento di uno switch propriamente detto. Essendo un oggetto software, possiede porte software (virtuali) a cui si «connettono» schede di rete software (virtuali).

In realtà lo switch software fa parte di un sistema di virtualizzazione che comprende un intero sistema operativo per comporre quella che in gergo informatico viene detta **macchina virtuale (VM, Virtual Machine)**.

### 7.1 Virtualizzazione



Da tempo sul mercato esistono programmi di virtualizzazione, ovvero programmi che, installati su un sistema operativo «padre», creano un ambiente isolato all'interno del quale opera un secondo sistema operativo del tutto indipendente, la *macchina virtuale*.

Nella figura a lato, il sistema operativo «padre», dotato di una interfaccia di rete, ha creato una VM con un suo proprio sistema operativo e che possiede una scheda di rete NIC virtuale.

Entrambi i sistemi accedono «contemporaneamente» alla rete, in modo indipendente e con le proprie configurazioni. La scheda di rete virtuale possiede infatti un proprio indirizzo MAC ed eventualmente anche un proprio indirizzo IP.

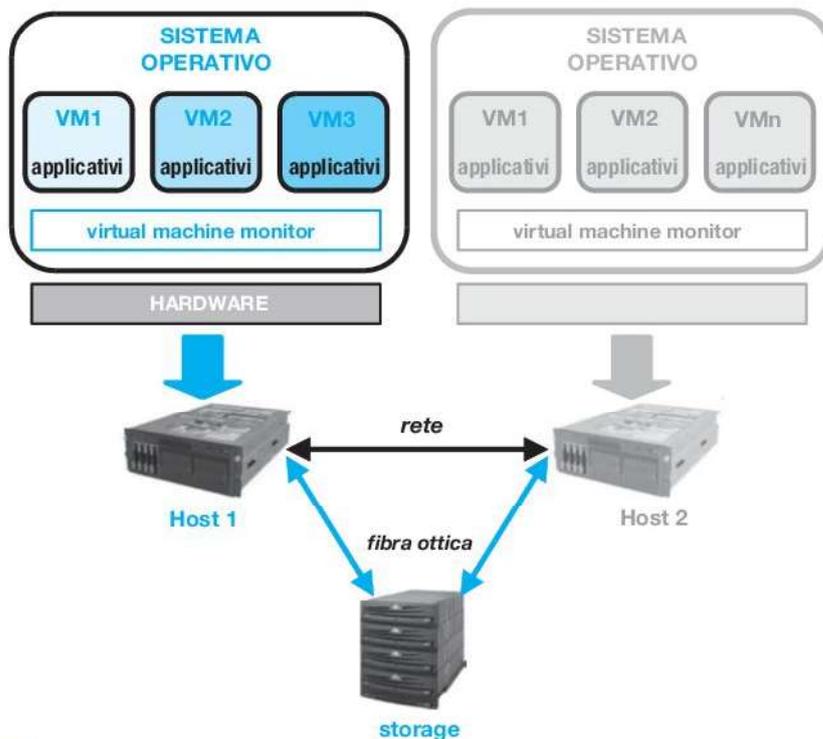
Per comunicare in rete la NIC virtuale «sfrutta» la NIC fisica: il software di virtualizzazione installato sul sistema operativo «padre» arbitra la risorsa fisica in base alle richieste dei due sistemi operativi.

Se all'inizio i programmi di virtualizzazione sono stati concepiti per poter utilizzare programmi applicativi *legacy* su macchine fisiche dotate di sistema operativo inappropriato per quegli applicativi, da qualche tempo questa funzionalità ha prepotentemente invaso anche il mondo delle reti, soprattutto per merito di un'azienda produttrice, VMware Inc. Anche altre aziende operano proficuamente nel mondo della virtualizzazione, come Microsoft e Citrix.

L'adozione di un sistema virtualizzato basato su una sola macchina fisica denominata **Host** comporta una serie di opportunità molto preziose:

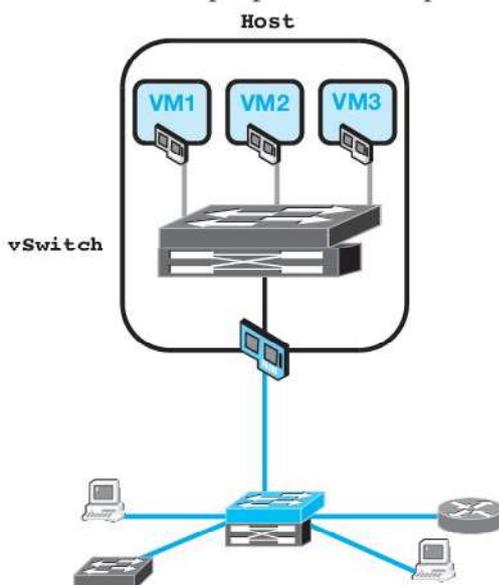
- **Consolidamento.** Su una sola macchina fisica operano più sistemi operativi (VM) e quindi più applicazioni. In questo modo si risparmiano i costi dell'hardware.
- **Centralizzazione.** Le attività di gestione (management) possono essere svolte con un solo referente fisico e in un solo luogo fisico. In questo caso si risparmia sul costo del lavoro (meno addetti).
- **Storage e backup.** I dati generati dalle applicazioni sono gestiti in modo coordinato e con tecnologie dedicate. In questo modo si risparmia sui costi di recupero dai disastri informatici.
- **Scalabilità.** L'ampliamento di un sistema virtualizzato è semplice, potendo associare più Host tra di loro.

Lo schema classico di un sistema virtualizzato è il seguente:



## 7.2 vSwitch

I sistemi virtuali hanno una diffusa applicazione anche come supporto a reti di notevoli dimensioni, sia per realizzare *cloud computing*, sia per ottimizzare grandi reti switched. Normalmente le VM ospitano programmi applicativi di tipo *server*, tipo programmi gestionali, server di database, web server, ecc., ognuno installato sul proprio sistema operativo più congeniale.



Un Host virtualizzato, contenente varie VM con altrettante NIC virtuali, può connetterle emulando un managed switch software detto **switch virtuale** (**vSwitch**, *virtual Switch*).

Nello schema della figura compaiono 3 VM con altrettante NIC virtuali, connesse ad un vSwitch; sullo stesso è attestata un'interfaccia fisica di rete (NIC) dell'Host che a sua volta comunica con il mondo esterno essendo collegata ad un managed switch classico.

Ogni vSwitch può connettersi a una sola interfaccia fisica, quindi se si volesse incrementare la banda del sistema aggiungendo interfacce fisiche all'Host, è necessario creare tanti vSwitch quante sono le schede di rete dell'Host.

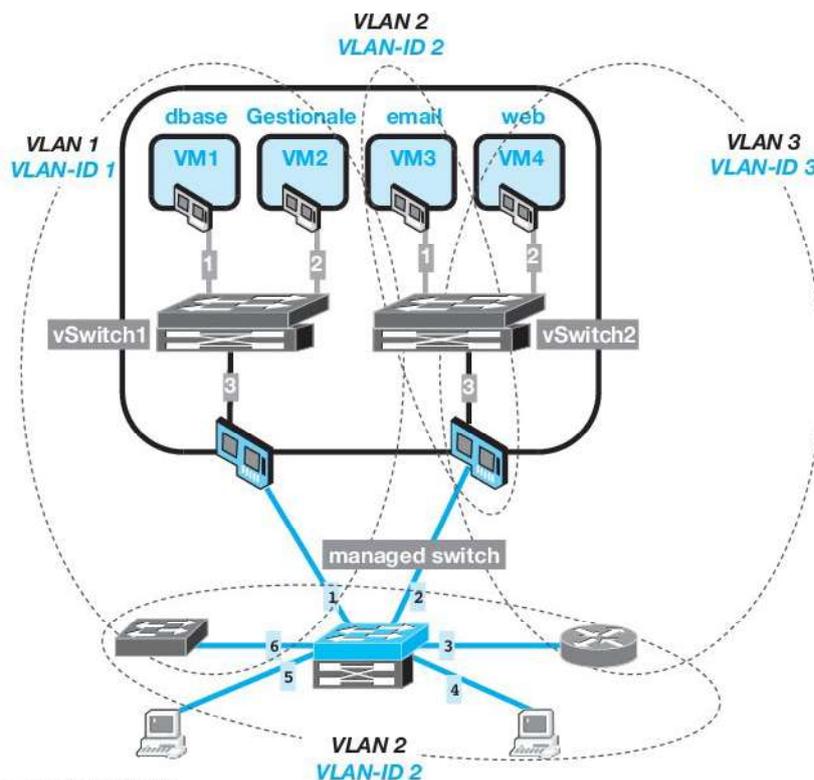
Per configurare un sistema del genere vanno fatte operazioni sia sull'Host, che sullo switch esterno, tipicamente configurazioni in modalità VLAN 802.1Q.

### Configurazione di un sistema VNWare

Disegnare lo schema di un Host VMWare che ospita quattro VM su cui operano il programma gestionale di un'azienda, un database server aziendale, un web server e un server di posta elettronica. Riportare l'elenco delle VLAN utilizzate compreso di porte access e trunk per vSwitch e managed switch.

La schematizzazione del sistema potrebbe essere quella rappresentata in figura, in cui si fa uso di tre **VLAN 1,2,3** e due **vSwitch 1,2**.

Su **VLAN1** si accumula il traffico interno per database server e programma gestionale; su **VLAN2** si accumula il traffico della posta elettronica e su **VLAN3** il traffico web.



La configurazione degli apparati:

- VLAN1** su **vSwitch1**: porte 1,2,3; trunk: porta 3.
- VLAN1** su **managed switch**: porte 1,6; trunk: porta 1.
- VLAN2** su **vSwitch2**: porte 1,3; trunk: porta 3.
- VLAN2** su **managed switch**: porte 1,2,3,4,5,6; trunk: porta 2.
- VLAN3** su **vSwitch2**: porte 2,3; trunk: porta 3.
- VLAN3** su **managed switch**: porte 2,3; trunk: porta 2.

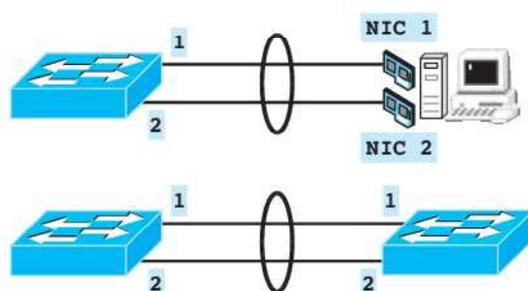
## 7.3 Link aggregation (dopo "Traffico e dimensionamento")

Già da molto tempo (1990) si è affrontata la possibilità di aumentare la larghezza di banda di una connessione Ethernet punto-punto (esempio, tra due switch), utilizzando per lo stesso flusso due o più canali fisici in parallelo.

Dal 2008 l'IEEE ha rilasciato uno standard che descrive il **LACP** (802.1AX, *Link Aggregation Control Protocol*), il protocollo che consente di aggregare un collegamento Ethernet su due o più linee fisiche.

Tipicamente l'operazione avviene tra due switch o tra uno switch e una macchina server e, oltre ad aumentare la banda della linea, aggiunge un buon grado di tolleranza ai guasti sul collegamento: quando una delle linee si guasta, il protocollo autonomamente «carica» le rimanenti.

Link aggregation



Il protocollo LACP, se previsto dai dispositivi (switch e NIC) opera autonomamente negoziando sia il collegamento, sia il tipo di bilanciamento di carico da ripartire sulle linee. Si basa, come sempre, su pacchetti LACPPDU Ethernet con uno speciale *Ethertype* che vale 8802h.

Il link aggregation è prezioso anche per aumentare la larghezza di banda delle interfacce fisiche dei vSwitch in sistemi virtualizzati.

## 8 Traffico e dimensionamento

È abbastanza interessante capire se una rete LAN switched è organizzata in modo da servire adeguatamente gli utenti in base alla quantità di traffico di cui necessitano. Per essere in grado di dimensionare correttamente le porzioni di una rete LAN è possibile effettuare il calcolo del traffico in ogni suo ramo e verificare se ogni dispositivo è ben dimensionato.

I dispositivi di interconnessione su cui è necessario effettuare il calcolo del flusso sono hub e switch (half e full duplex), pertanto si analizzano i flussi veicolati da questi dispositivi nel caso tipico in cui devono servire una certa quantità di host che scambiano dati su una o più **porte di destinazione** (di uplink o collegate a server), che è la situazione tipica.

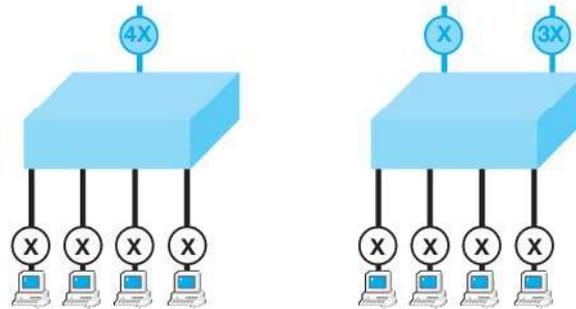
Nei casi analizzati si fanno due ipotesi base che consentono di generalizzare i calcoli, ovvero:

- il traffico degli host è omogeneo: ogni host connesso allo stesso dispositivo genera la stessa quantità di traffico (esempio, X);
- non si calcola l'effetto delle collisioni, ritenendole equipartite per ogni host.

Per ogni configurazione va analizzato il flusso di traffico generato dagli host sulla porta di destinazione.

Dato il numero  $n$  di host che generano ognuno un traffico  $X$ , se la porta di destinazione è una sola, il flusso su di essa vale  $n \cdot X$ .

Se le porte di destinazione sono più di una, è necessario specificare la percentuale di traffico su ognuna. Per esempio, due porte di destinazione con 25% di traffico sulla prima e 75% sulla seconda, subiranno rispettivamente da  $n$  host i due flussi  $(1/4) \cdot n \cdot X$  e  $(3/4) \cdot n \cdot X$  (la somma dei due traffici deve ritornare ad essere  $n \cdot X$ ).



In condizioni ideali nel primo caso ogni host può disporre di un quarto della banda totale, dovendo poi immettere sulla porta di destinazione l'intero traffico  $4X$  delle stazioni.

Nel secondo caso, invece, se su ogni porta di destinazione è disponibile l'intera banda del dispositivo, ecco che la condizione peggiore viene imposta dalla porta che deve veicolare  $3X$  di traffico, per cui il singolo host può disporre di un terzo della banda disponibile.

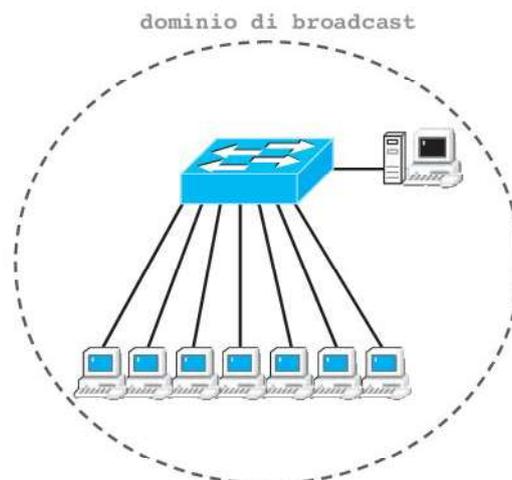
**ESEMPIO**

**Dimensionamento di una LAN**

Un'Azienda usa una LAN su cui lavorano due utenti dell'amministrazione e 5 utenti della progettazione. I calcolatori sono connessi a uno switch full duplex 100Mbit/s a 8 porte a cui è collegato l'unico server. Calcolare quanta banda ha a disposizione ogni utente.

Calcolare quanta banda si guadagna usando due server differenti.

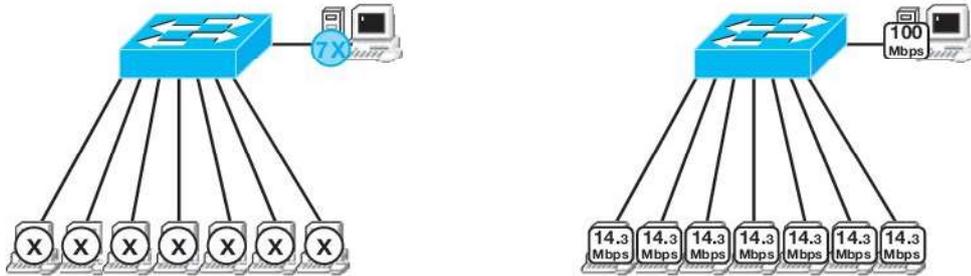
La LAN ha questa struttura:



Siccome i 7 host generano traffico omogeneo verso l'unico server, ogni singolo host avrà un traffico denominato  $X$ . ▶

Quando lavorano «contemporaneamente» generano quindi un flusso verso il server di **7X**.

Dovendo spartirsi la banda dello switch 100 Mbit/s sull'unico ramo di destinazione al server, vale l'equazione **7X = 100 Mbit/s**, da cui si ricava **X = 100/7 = 14,3 Mbit/s**.



Con due server si possono distinguere due tipi di traffico: **X** originato dagli utenti dell'amministrazione verso il proprio server, e **Y** generato dagli utenti progettisti.

Ora sono presenti due flussi, **2X** e **5Y** verso i rispettivi server.

La banda per gli utenti amministrativi sarà ora determinata dall'equazione **2X = 100 Mbit/s** e la banda per gli utenti progettisti dall'equazione **5Y = 100 Mbit/s**.

Risolviendo si ottiene **X = 50 Mbit/s** e **Y = 20 Mbit/s**.



In definitiva due utenti guadagnano **50 - 14,3 = 35,7 Mbit/s** di banda e cinque utenti guadagnano **20 - 14,3 = 5,7 Mbit/s** di banda.

Per il calcolo della banda disponibile per gli host e per la porta di destinazione il caso ideale è rappresentato dagli switch full duplex.

Per hub e switch half duplex bisogna differenziare i casi, dato che la banda che mettono a disposizione va ripartita sulle varie porte in diversi modi.

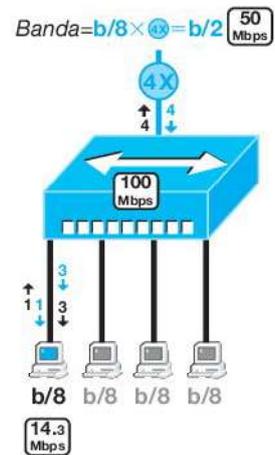
In un hub ogni singolo pacchetto trasmesso da una stazione viene ripetuto su ogni porta.

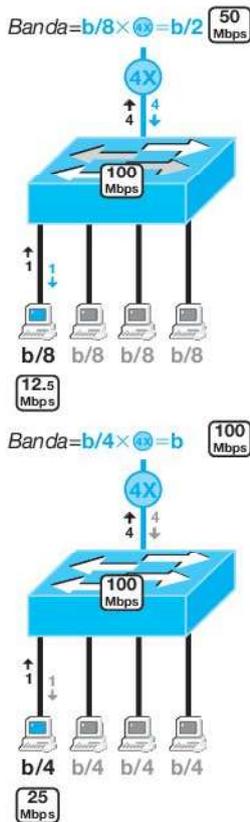
Per esempio, nel caso peggiore in cui 4 host devono comunicare attraverso una porta di destinazione, quando un singolo host invia un pacchetto, questo si propaga a tutti gli altri host; quindi dalla porta di destinazione giunge un pacchetto di risposta, che a sua volta si propaga a tutti gli altri host. Ora su ogni singolo ramo dell'hub stanno circolando 2 pacchetti. Se la stessa operazione viene ripetuta dagli altri tre host, abbiamo altri 6 pacchetti che circolano sui singoli rami dell'hub, per un totale di 8 pacchetti su ogni singolo ramo.

In definitiva, detto **X** il traffico di un utente su hub con **n** porte e una porta di destinazione, il traffico su ogni ramo vale **2 × n × X**.

In generale su ogni porta di un hub si genera un traffico uguale alla somma di tutti i traffici su tutte le sue porte.

Nel caso in figura (a lato), il traffico su ogni ramo è uguale a 8 pacchetti, perciò ogni singolo host può generare al più un ottavo di banda **b** totale.





Siccome sulla porta di destinazione deve transitare un flusso di  $4X$ , la banda rimanente per la porta di destinazione vale  $(1/8)b \times 4 = b/2$ .

In uno switch half duplex i pacchetti non vengono duplicati, ma nel ramo di destinazione abbiamo comunque 8 pacchetti (4 utili e 4 risposte), mentre su ogni ramo degli host abbiamo solo 2 pacchetti (1 utile e 1 risposta): lo switch manda a destinazione i pacchetti senza duplicarli su altre porte.

Il caso peggiore rimane sul ramo della porta di destinazione con 8 pacchetti, pertanto ogni singolo host può generare solo un ottavo della banda totale. Di nuovo, sulla porta di destinazione transita un flusso di  $4X$ , e la banda rimanente per la porta di destinazione vale  $(1/8)b \times 4 = b/2$ .

In uno switch full duplex i pacchetti non vengono né duplicati né si occupa banda per le risposte, pertanto nel ramo di destinazione abbiamo solo 4 pacchetti (4 utili e 4 risposte contemporanee che non sprecano banda); per lo stesso motivo su ogni ramo degli host abbiamo solo 1 pacchetto (la risposta non occupa banda utile).

Il caso peggiore rimane sul ramo della porta di destinazione con 4 pacchetti, pertanto ogni singolo host può generare un quarto della banda totale.

Ora, sulla porta di destinazione transitando un flusso di  $4X$ , la banda disponibile vale  $(1/4)b \times 4 = b$ , cioè la banda totale dello switch full duplex.

**ESEMPIO**

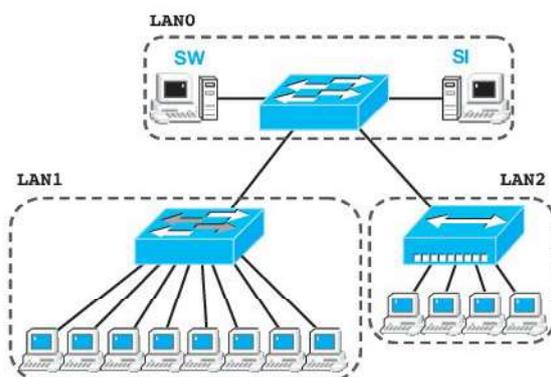
**Dimensionamento di una LAN segmentata**

Un'Azienda usa una XLAN con tre LAN: in una operano 4 utenti dell'amministrazione, collegati ad un hub; in un'altra LAN operano 10 addetti alla produzione connessi ad uno switch half duplex; lo switch e l'hub sono interconnessi da un altro switch a cui sono collegati anche due server SI (interno) e SW (dati web).

Si ipotizza che in amministrazione gli utenti lavorino solo con SI, mentre in produzione gli addetti lavorano al 50% su SI e al 50% su SW.

Calcolare le bande sulle porte dello switch principale e confrontarle con quelle ottenute se tutti i dispositivi fossero switch full duplex.

La XLAN descritta è la seguente:



I flussi sulle porte si calcolano con le rispettive regole:

**LAN1**

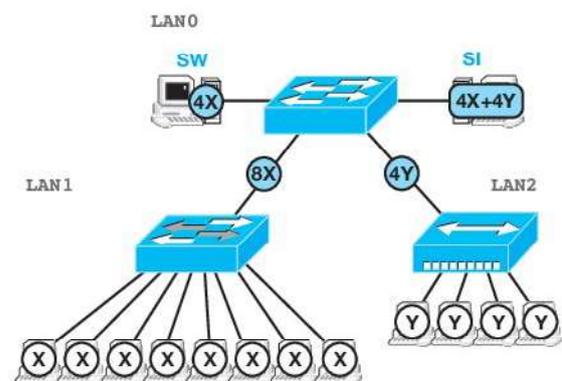
8 host omogenei, detto  $X$  il traffico di ognuno, sulla porta di destinazione si ha un flusso  $8X$ .

**LAN2**

4 host omogenei, detto  $Y$  il traffico di ognuno, sulla porta di destinazione si ha un flusso di  $4Y$ .

**LAN0**

verso **SI** giunge un flusso  $4X+4Y$ , mentre su **SW** un traffico di  $4X$ .



Le bande si calcolano con le rispettive regole:

**LAN1**

Siccome abbiamo uno switch half duplex con ▶

una sola porta di destinazione e 8 host, la banda di un host vale  $b/16$ , cioè  $16X = 100$  Mbit/s, da cui  $X = 6,25$  Mbit/s.

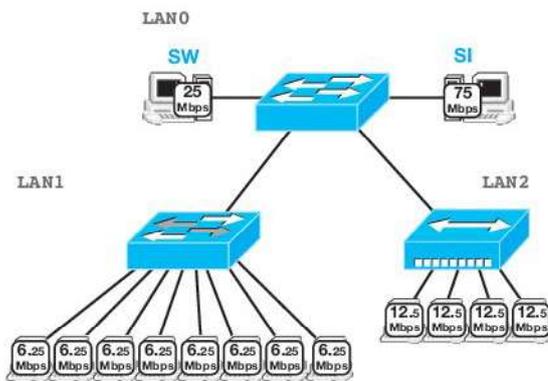
**LAN2**

Con un hub e 4 host a prescindere dal numero di porte di destinazione, la banda di un host vale  $b/8$ , cioè  $8Y = 100$  Mbit/s, da cui  $Y = 12,5$  Mbit/s.

**LAN0**

su **SI** abbiamo  $4X+4Y$ , cioè  $4 \cdot 6,25 + 4 \cdot 12,5 = 25+50 = 75$  Mbit/s. Siccome si tratta di uno switch a 100 Mbit/s, si conclude che per **SI** la rete è ben dimensionata.

Su **SW** abbiamo  $4X$ , cioè  $4 \cdot 6,25 = 25$  Mbit/s, e di nuovo per **SW** la rete è ben dimensionata.



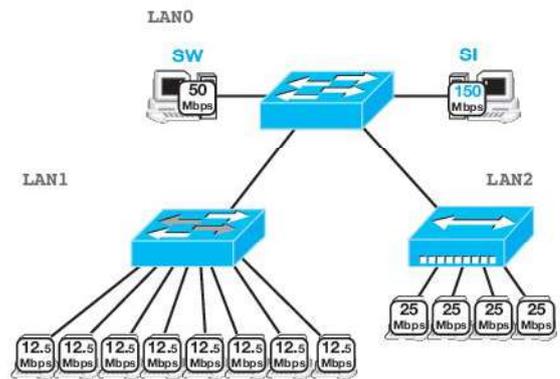
Se i tre dispositivi fossero tutti switch full duplex il calcolo dei flussi rimane ovviamente invariato ma quello delle bande cambia:

**LAN1**

switch full duplex con 8 host. la banda di un host vale  $b/8$ , cioè  $8X = 100$  Mbit/s, da cui  $X = 12,5$  Mbit/s. Il guadagno di banda vale 6,25 Mbit/s.

**LAN2**

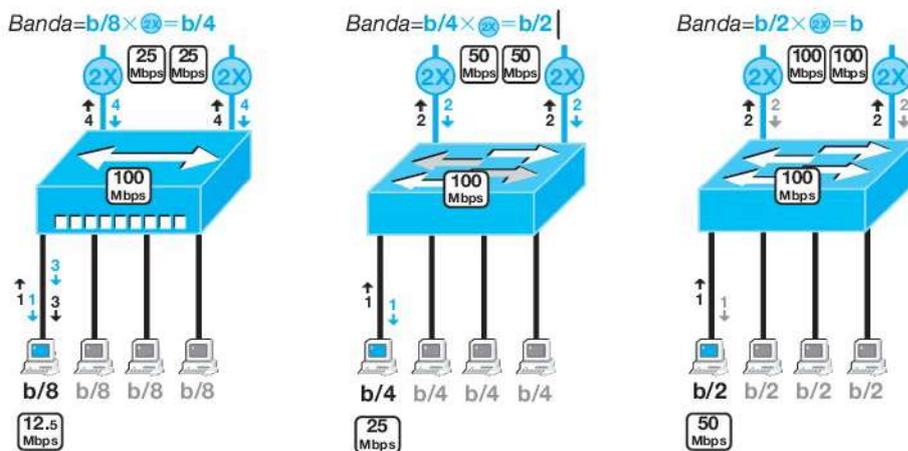
switch full duplex con 4 host, la banda di un host vale  $b/4$ , cioè  $4Y = 100$  Mbit/s, da cui  $Y = 25$  Mbit/s. Il guadagno di banda vale 12,5 Mbit/s.



Si nota però che la porta verso **SI** della **LAN0**, dovendo sopportare  $4X+4Y = 4 \cdot 12,5 + 4 \cdot 25 = 50+100 = 150$  Mbit/s, è insufficiente.

Il traffico interno ad uno switch half duplex è sensibilmente più veloce di un equivalente traffico interno ad un hub, dato che se le comunicazioni sono disgiunte è sempre disponibile la banda massima su ogni porta. Ma quando si deve calcolare il flusso nel caso di una sola porta di destinazione, lo switch half duplex si comporta come un hub.

Se invece le porte di destinazione aumentano, lo switch half duplex ritorna ad essere più efficiente di un hub equivalente, come si nota dalla figura.



## Dimensionamento di una LAN con router

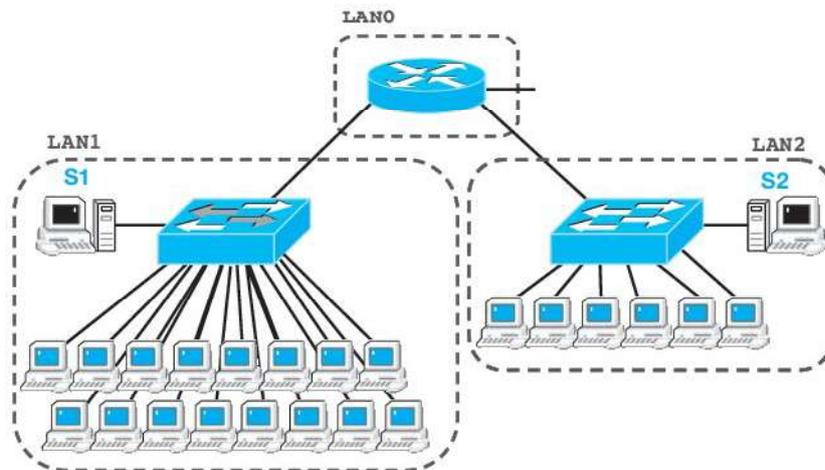
Un'Azienda usa una XLAN con tre LAN connesse da un router: in una operano 16 addetti alla produzione e un server S1 connessi ad un hub; in un'altra LAN operano 6 utenti dell'amministrazione e un secondo server S2 collegati ad uno switch full duplex; hub e switch sono connessi da un router che ha anche un collegamento Internet.

In produzione gli utenti lavorano per l'80% sul proprio server e il rimanente su Internet.

In amministrazione invece gli utenti lavorano per il 50% sul proprio server e il rimanente su Internet.

Calcolare le bande disponibili e confrontarle con quelle ottenute sostituendo l'hub con uno switch half duplex.

Lo schema della rete è il seguente:



I flussi sulle porte si calcolano con le rispettive regole:

### LAN1

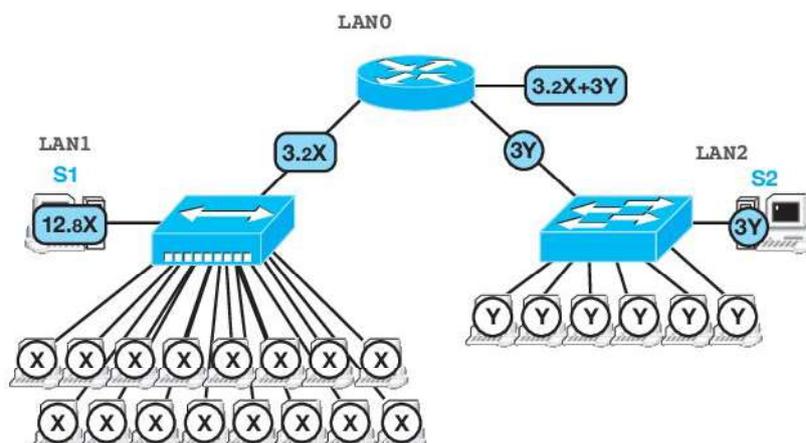
con 16 host omogenei e traffico per l'80% su S1 e 20% su Internet (**router**), detto **X** il traffico di ognuno, verso il server S1 si ha un flusso  $(80 \cdot 16/100)X = 12,8X$ , mentre verso il router  $(20 \cdot 16/100)X = 3,2X$ .

### LAN2

con 6 host omogenei con traffico suddiviso tra **S2** e Internet (**router**), detto **Y** il traffico di ognuno, verso **S2** si ha un flusso di  $3Y$  e verso il router di  $3Y$ .

### LAN0

Sulla porta verso **Internet** il flusso è la somma dei due flussi provenienti da **LAN1** e **LAN2**, ovvero  $3,2X+3Y$ .



Per il calcolo delle bande disponibili:

### LAN1

Un hub con 16 host, a prescindere dal numero di destinazioni, suddivide la banda del dispositivo per 32, cioè ogni host ha  $b/32$  di banda, cioè  $32X = 100$  Mbit/s, da cui  $X = 3,125$  Mbit/s.

Sulle due destinazioni si hanno quindi:  $S1 = 12,8X = 12,8 \cdot 3,125 = 40$  Mbit/s e verso il router  $3,2X = 3,2 \cdot 3,125 = 10$  Mbit/s.

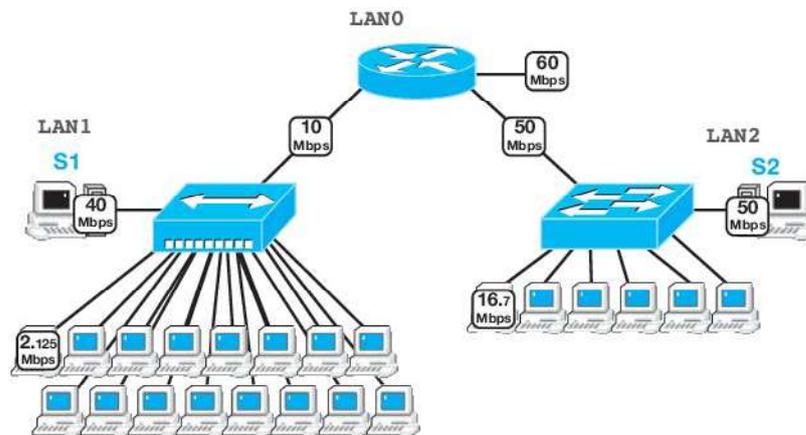
### LAN2

Uno switch full duplex con 6 host comporta che ogni host possa generare un traffico  $b/6$ , ovvero  $6Y = 100$  Mbit/s, da cui  $Y = 16,7$  Mbit/s. La banda verso il router e verso S2 è identica,  $3Y = 3 \cdot 16,7 = 50$  Mbit/s.

### LAN0

Effettuando i calcoli, per Internet la banda vale

$$3,2X + 3Y = 3,2 \cdot 3,125 + 3 \cdot 16,7 = 10 + 50 = 60 \text{ Mbit/s.}$$



Se al posto dell'hub si utilizzasse uno **switch half duplex**, avendo due destinazioni all'80% e al 20% ognuna in grado di supportare la massima velocità del dispositivo, l'80% del traffico degli host può usufruire della banda  $b$  e il 20% della stessa banda  $b$ , ovvero:

$$80\% \cdot 32X = 100 \text{ Mbit/s (e } 20\% \cdot 32X = 100 \text{ Mbit/s),}$$

da cui

$$X = 3,9 \text{ Mbit/s (e } X = 15,6 \text{ Mbit/s).}$$

Tra questi due valori si considera il peggiore, pertanto  $X = 3,9$  Mbit/s.

In questo caso le rimanenti bande risultano:

$$S1 = 12,8X = 50 \text{ Mbit/s;}$$

$$\text{router} = 3,2X = 12,5 \text{ Mbit/s;}$$

$$\text{internet} = 3,2X + 3Y = 12,5 + 50 = 62,5 \text{ Mbit/s.}$$

Se infine si usasse uno **switch full duplex** al posto dell'hub, come prima l'80% del traffico degli host (e il 20%) avrebbero a disposizione l'intera banda  $b$  del dispositivo, cioè:

$$80\% \cdot 16X = 100 \text{ Mbit/s (e } 20\% \cdot 16X = 100 \text{ Mbit/s),}$$

da cui

$$X = 7,8 \text{ Mbit/s (e } X = 31,25 \text{ Mbit/s).}$$

Tra questi due valori, per  $X$  va preso quello peggiore,  $7,8$  Mbit/s, che comunque fa guadagnare alla LAN una prestazione più che raddoppiata rispetto al caso con l'hub.

In questo caso le rimanenti bande risultano:

$$S1 = 12,8X = 99,8 \text{ Mbit/s;}$$

$$\text{router} = 3,2X = 25 \text{ Mbit/s;}$$

$$\text{internet} = 3,2X + 3Y = 25 + 50 = 75 \text{ Mbit/s.}$$

## ESERCIZI PER LA VERIFICA ORALE

Saper rispondere ai **requisiti fondamentali** dà una sufficiente garanzia per sentirsi pronti all'interrogazione. Saper anche rispondere ai **requisiti avanzati** dimostra una padronanza eccellente degli argomenti del capitolo.

### Requisiti fondamentali

- 1 Elencare tutti gli apparati di interconnessione per le LAN e collocarli al proprio livello operativo OSI.
- 2 Descrivere la circolazione dei frame all'interno di un hub.
- 3 Descrivere la circolazione dei frame all'interno di uno switch.
- 4 Spiegare perché uno switch appena avviato si comporta come un hub.
- 5 Illustrare la topologia fisica di una switched LAN e la ragione per cui deve essere connessa in quel modo.
- 6 Mostrare il funzionamento dello Spanning Tree Protocol attraverso lo schema di una switched LAN disegnato su un foglio.
- 7 Ricordare la caratteristica fondamentale di uno switch full duplex e discuterne gli effetti.
- 8 Il dominio di broadcast e il dominio di collisione: per hub, switch e switch full duplex.
- 9 Spiegare cos'è e a cosa serve l'autonegoziazione.
- 10 Discutere le ragioni per cui una LAN dovrebbe essere organizzata tramite VLAN.
- 11 Spiegare cosa si intende per porta trunk.
- 12 Spiegare in cosa consiste una Virtual Machine (VM).
- 13 Definire e descrivere la nozione di virtual switch.

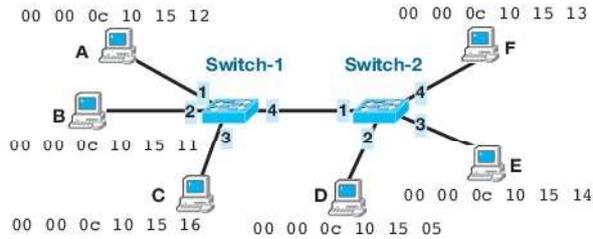
### Requisiti avanzati

- 1 Spiegare cosa si intende con «management» di una LAN e mostrarne un esempio.
- 2 Mostrare quando avvengono le collisioni rispettivamente in un hub, in uno switch e in uno switch full duplex.
- 3 Descrivere quando e perché un apparato di interconnessione inoltra un pacchetto in flooding.
- 4 Illustrare il concetto di CAM e filtering database. **MAC table**
- 5 Spiegare come si ottiene fault tolerance in una rete LAN.
- 6 Spiegare quando avviene il broadcast storming e come si previene.
- 7 Elencare e discutere le tre fasi operative dell'algoritmo che guida STP.
- 8 Access port e porte di trunk per una VLAN: illustrarne ruoli e funzioni.
- 9 Spiegare la funzione del campo ageing time nel filtering database.
- 10 Illustrare l'utilità dell'impiego di macchine virtuali per la gestione delle reti LAN.
- 11 Fornire qualche esempio per cui è importante poter fruire della tecnica di Link Aggregation.
- 12 Dominio di collisione e di broadcast nelle VLAN.
- 13 Descrivere la tecnologia denominata per antonomasia VMWare.

## ESERCIZI PER LA VERIFICA SCRITTA

### Apparati e instradamento

- 1 Data la seguente switched LAN e la seguente sequenza di traffico, riportare il contenuto della CAM dello Switch-1.



Traffico:

0. switch e host avviati

1. F spedisce a E

2. E spedisce a A

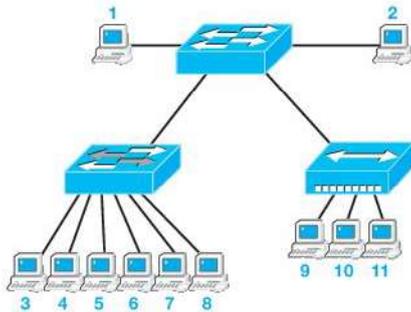
3. A spedisce a D

4. D spedisce a C

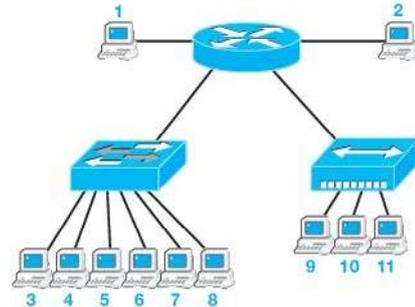
5. B spedisce a D

6. B spedisce a A

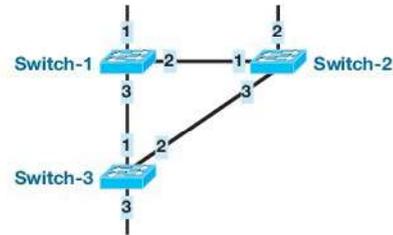
- 2 Dato lo schema dell'esercizio precedente, elencare quali trasmissioni sono realizzate in flooding e quali inoltrate direttamente.
- 3 Dato lo schema dell'esercizio precedente, riportare il contenuto della CAM dello Switch-2.
- 4 Data la seguente switched LAN, individuare i domini di collisione e i domini di broadcast elencando i nomi degli host.



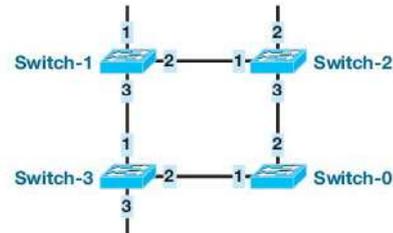
- 5 Data la seguente rete, individuare i domini di collisione e i domini di broadcast elencando i nomi degli host.



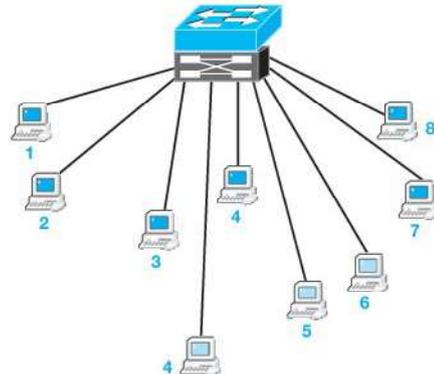
- 6 Data la seguente topologia fisica, applicare STP considerando che i MAC address degli Switch-X siano 00:00:00:cc:cc:cX; che le priorità siano tutte di default (32768) e che i costi delle porte siano tutti uguali a 10.



- 7 Data la seguente topologia fisica, applicare STP considerando che i MAC address degli Switch-X siano 00:00:00:cc:cc:cX; che le priorità siano tutte di default (32768) e che i costi delle porte siano tutti uguali a 10.



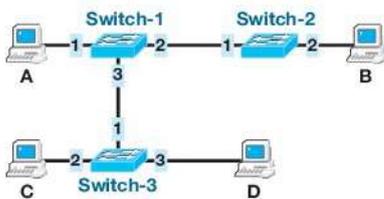
- 8 Data la seguente VLAN (con 3 VLAN-ID), individuare i domini di collisione e i domini di broadcast elencando i nomi degli host.



9 Data la VLAN dell'esercizio precedente, detti ethX le sue porte con X numerato da sinistra a destra (Host 1 su eth1), configurare il managed switch.

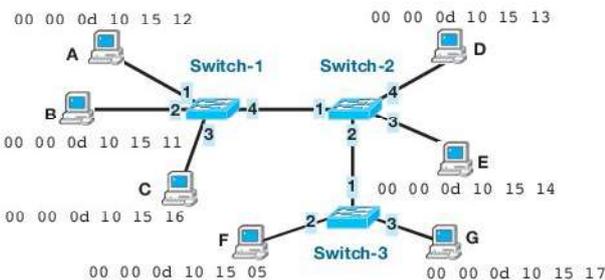
10 Dato lo schema seguente (4 host e 3 switch con le CAM inizialmente vuote):

- B spedisce a D.
  - Quali switch imparano dove si trova B?
  - A vede il pacchetto?
- D spedisce a B.
  - Quali switch imparano dove si trova D?
  - A vede il pacchetto?
- A spedisce a B.
  - Quali switch imparano dove si trova A?
  - D vede il pacchetto?
- D spedisce a A.
  - Quali switch imparano dove si trova D?
  - C vede il pacchetto?



11 Dato lo schema e il traffico a), b) c) d) dell'esercizio precedente, determinare il filtering database dei tre switch.

12 Data la seguente switched LAN e la seguente sequenza di traffico, riportare il contenuto della CAM dello Switch-1.



Traffico:

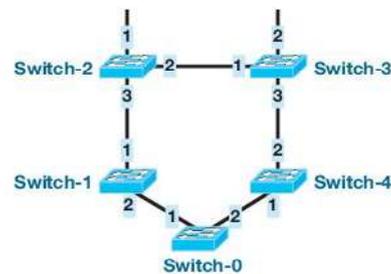
- switch e host avviati
- F spedisce a E
- E spedisce a A
- A spedisce a D
- D spedisce a C
- E spedisce a F
- D spedisce a A

13 Dato lo schema dell'esercizio precedente, elencare quali trasmissioni sono realizzate in flooding e quali inoltrate direttamente.

14 Dato lo schema dell'esercizio precedente, riportare il contenuto della CAM dello Switch-2.

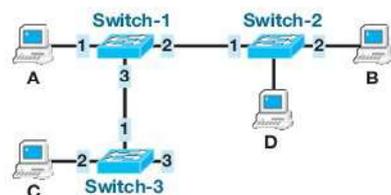
15 Dato lo schema dell'esercizio precedente, riportare il contenuto della CAM dello Switch-3.

16 Data la seguente topologia fisica, applicare STP considerando che i MAC address degli Switch-X siano 00:00:00:cc:cc:cX; che le priorità siano tutte di default (32768) e che i costi delle porte siano tutti uguali a 10.



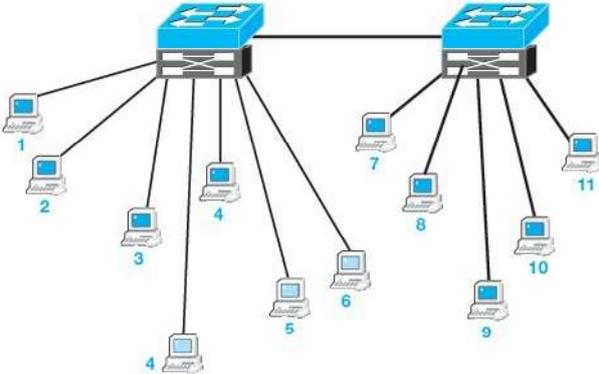
17 Dato lo schema seguente (4 host e 3 switch con le CAM inizialmente vuote):

- B spedisce a C.
  - Quali switch imparano dove si trova B?
  - D vede il pacchetto?
- C spedisce a B.
  - Quali switch imparano dove si trova C?
  - D vede il pacchetto?
- D spedisce a A.
  - Quali switch imparano dove si trova D?
  - C vede il pacchetto?
- A spedisce a C.
  - Quali switch imparano dove si trova A?
  - B vede il pacchetto?



18 Dato lo schema e il traffico a), b) c) d) dell'esercizio precedente, determinare il filtering database dei tre switch.

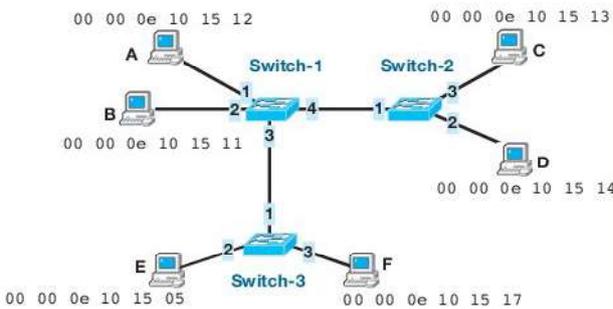
- 19** Data la seguente VLAN (con 3 VLAN-ID), individuare i domini di collisione e i domini di broadcast elencando i nomi degli host.



- 20** Data la VLAN dell'esercizio precedente, detti faX le porte dello switch di sinistra, numerate da sinistra a destra (Host 1 su fa1), configurare il managed switch di sinistra, considerando che le due porte trunk siano due Giga Ethernet (Gi/1 e Gi/2).

- 21** Data la VLAN dell'esercizio precedente, detti faX le porte dello switch di destra numerate da sinistra a destra (Host 7 su fa1 e Host 11 su fa5), configurare il managed switch di destra, considerando che le due porte trunk siano due Giga Ethernet (Gi/1 e Gi/2).

- 22** Data la seguente switched LAN e la seguente sequenza di traffico, riportare il contenuto della CAM dello Switch-1.



Traffico:

0. switch e host avviati  
 1. A spedisce a F

2. F spedisce a D  
 3. D spedisce a F  
 4. C spedisce a A  
 5. F spedisce a A  
 6. B spedisce a E

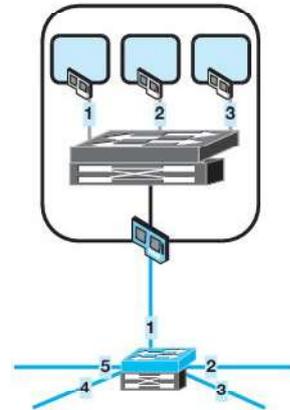
- 23** Dato lo schema dell'esercizio precedente, elencare quali trasmissioni sono realizzate in flooding e quali inoltrate direttamente.

- 24** Dato lo schema dell'esercizio precedente, riportare il contenuto della CAM dello Switch-2.

- 25** Dato lo schema dell'esercizio precedente, riportare il contenuto della CAM dello Switch-3.

- 26** Dato lo schema VMWare seguente, composto da 3 virtual machine, 1 virtual switch e un managed switch, elencare i comandi di configurazione se le prime due VM trattano il traffico aziendale e la terza garantisce il collegamento a Internet.

Riportare le ipotesi aggiuntive per giustificare le scelte di configurazione.



- 27** Aggiungere un secondo virtual switch alla configurazione dell'esercizio precedente (rinominando alcune porte all'interno del VMWare), dedicato al servizio di connessione a Internet. Elencare i comandi di configurazione opportuni per soddisfare la richiesta.

## ESERCIZI PER LA VERIFICA DI LABORATORIO



Per compilare gli esercizi proposti è stato usato l'ambiente gratuito multiplatforma Windows/Linux **Code::Blocks 10.5** con **gcc 4.4.1** e **Eclipse SDK 4.2.0** con **jre7**.

I testi degli esercizi presentano uno o più **layout** di input/output richiesti; in questo modo sono indicate, a volte, alcune specifiche del programma come l'output, il controllo dell'input o i valori ammissibili.

### Apparati e instradamento

- 1 Scrivere un programma che acquisisce in input l'attività su uno switch a 8 porte nella forma (n. porta, MAC destinatario, MAC sorgente), quindi ne mostri il procedimento di inoltramento e il filtering database.

Layout:

```

                                OUTPUT
Traffico
Porta: 1
d: 0 7 d af f4 54 s: 8 0 37 15 e6 bc
Porta: 2
d: 8 0 37 15 e6 b1 s: 0 12 3f 4a 33 d2
Porta: 3
d: 8 0 37 15 e6 bc s: 0 0 c0 9f a0 97
Porta: 2
d: ff ff ff ff ff ff s: 8 0 37 15 e6 bc
Porta: 8
d: 0 12 3f 4a 33 d2 s: 0 12 3f 76 12 1a
Porta: 0
Learning e forwarding
.....
Trama rx su 1 -> broadcast su 2 3 4 5 6 7 8
Trama rx su 2 -> broadcast su 1 3 4 5 6 7 8
Trama rx su 3 -> 1
Trama rx su 2 -> broadcast su 1 3 4 5 6 7 8
Trama rx su 8 -> 2

Filtering database
.....
P: 1 MAC: 08 00 37 15 E6 BC
P: 2 MAC: 00 12 3F 4A 33 D2
P: 3 MAC: 00 00 C0 9F A0 97
P: 2 MAC: 08 00 37 15 E6 BC
P: 8 MAC: 00 12 3F 76 12 1A
    
```

- 2 Scrivere un programma che consenta all'utente di scegliere il valore del campo ability di un

LCW e stampi a schermo la codifica impulsiva relativa.

Layout:

```

                                OUTPUT
Selezionare il valore ability:
1) 10BASE-T;
2) 10BASE-T full duplex;
3) 100BASE TX;
4) 100BASE-TX full duplex;
5) 100 BASE-T4;
6) pause
7) asymmetric pause
: 3
: 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
| | | | | | | | | | | | | | | |
    
```

- 3 Scrivere un programma che legge da un file di testo la sequenza di un frame Ethernet e, se di tipo 801.1Q (VLAN), ne mostri a schermo le caratteristiche.

Layout:

```

                                OUTPUT 1
Frame Ethernet:
ff ff ff ff ff ff ca 03 0d b4 00 1c 81 00 00
64 81 00 00 c8 08 06 00 01 08 00 06 04 00
01 ca 03 0d b4 00 1c c0 a8 02 c8 00 00 00 00
00 00 c0 a8 02 fe 00 00 00 00 00 00 00 00 00
00 00 00 00 00
Frame 802.1Q (TPID = 8100h)
VLAN-ID: 100 (000001100100b)

                                OUTPUT 2
Frame Ethernet:
ff ff ff ff ff ff 00 16 ce 6e 8b 24 08 06 00
01 08 00 06 04 00 01 00 16 ce 6e 8b 24 c0 a8
00 72 00 00 00 00 00 00 c0 a8 00 01
Frame non 802.1Q
    
```

- 4 Scrivere un programma che legge da un file di testo la sequenza di un frame Ethernet e incapsularla in 802.1Q con VLAN-ID a piacimento.

Layout:

```
OUTPUT
Frame Ethernet:
ff ff ff ff ff ff 00 16 ce 6e 8b 24 08 06 00
01 08 00 06 04 00 01 00 16 ce 6e 8b 24 c0 a8
00 72 00 00 00 00 00 00 c0 a8 00 01

Incapsulato in 802.1Q con VLAN-ID=100:
ff ff ff ff ff ff 00 16 ce 6e 8b 24 81 00
00 64 08 06 00 01 08 00 06 04 00 01 00 16
ce 6e 8b 24 c0 a8 00 72 00 00 00 00 00
c0 a8 00 01
```

- 5 Scrivere un programma che acquisisce in input un numero su 16 bit come LCW, isola i bit del campo ability e stampa a schermo la sua decodifica.

Layout:

```
OUTPUT
Inserisci un numero (<65536): 35081
35081=1000100100001001b (LCW)
bit8=1, 100BASE-TX full duplex
bit1=1, asymmetric pause
```