

## EMULAZIONE DI SPANNING TREE MINIMO

Lo scopo del progetto è emulare il comportamento dell'algoritmo di spanning tree che determina quali sono i percorsi utilizzati dai bridges di una rete ethernet per evitare loop. Data la descrizione di una rete di bridges che connettono alcune LANs, deve essere realizzata, in linguaggio ANSI C, una applicazione che scambia i messaggi tra i bridges e costruisce le tabelle di instradamento.

Un processo iniziale legge da un file la configurazione delle LAN presenti nel sistema e dei bridges, come rappresentato in figura 1. Possono esserci al massimo 12 LANs e 5 bridges.

Le comunicazioni necessarie allo spanning tree sono realizzate mediante datagram UDP.

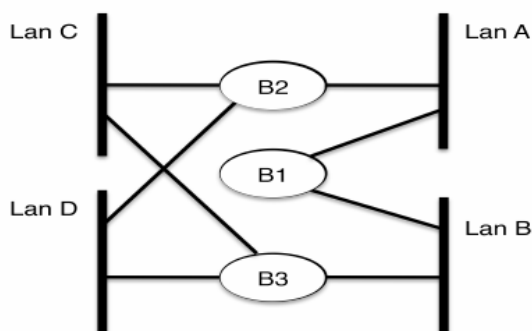


Figura 1: un esempio di rete

Ogni LAN è implementata da un thread che mantiene aperte tante porte UDP quanti sono i bridges a cui è collegata e che conosce la porta UDP di ciascun bridge a cui è collegata.

Ogni Bridge è implementato da un thread che mantiene aperte tante porte UDP quante sono le LANs a cui è collegato e che conosce le porte UDP di ciascuna LAN a cui è collegato.

Quando una LAN riceve un messaggio con una destinazione di tipo broadcast la invia a tutti i bridge che afferiscono a quella LAN, sulle specifiche porte.

Il processo iniziale crea un thread per ogni bridge e li configura passando loro dei parametri per informarli di quale è il loro identificatore univoco ed istruirli su quali porte UDP utilizzare e quali sono le porte UDP delle LANs a cui sono connessi.

Il processo iniziale crea un thread per ogni LAN e li configura passando loro dei parametri per istruirli su quali porte UDP utilizzare e quali sono le porte UDP dei bridges a cui sono direttamente connesse.

Terminata la fase di configurazione iniziale, i thread non possono più utilizzare la memoria condivisa per passarsi informazioni.

Da quel momento in avanti, i bridges devono cominciare a trasmettersi dei messaggi, attraverso le LANs, secondo l'algoritmo dello spanning tree per determinare le tabelle di forwarding (che determinano l'albero di copertura).

Ogni volta che una tabella di un bridge viene modificata, il thread che implementa quel bridge deve stampare a video la propria tabella, preceduta dall'identificatore del bridge.

E' consigliato implementare alcune funzioni, usate da ciascun bridge, per semplificare la gestione e debugging dell'output:

```

stampa_tabella
stampa_pacchetto_trasmesso
stampa_pacchetto_ricevuto
  
```

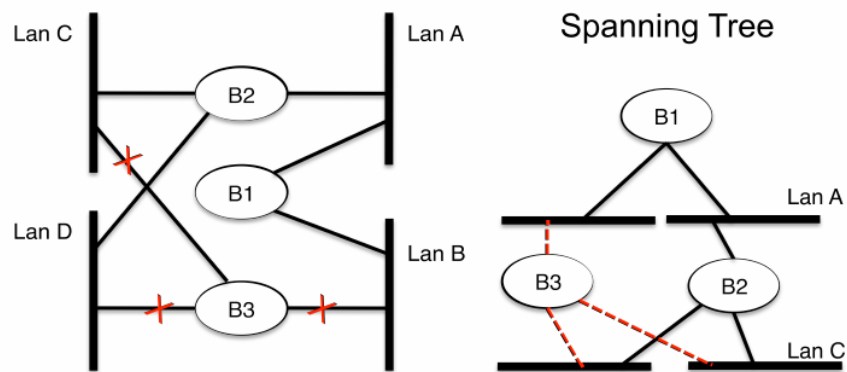


Figura 2: La soluzione finale, descritta dalle tabelle finali.

Ogni altra ipotesi di lavoro deve essere decisa dal progettista.  
Anche la scelta del formato del file di configurazione è responsabilità del progettista.