

# 14

## LE TECNICHE DI INTERNETWORKING

---

### 14.1 INTRODUZIONE

Nel capitolo 2 è stato introdotto il modello di riferimento OSI. In tale modello una rete di calcolatori viene vista come un insieme di sistemi interconnessi tra loro. Su alcuni sistemi (gli ES: *End System*) risiedono le applicazioni che comunicano usando la rete, altri sistemi hanno funzioni di instradamento dei messaggi (gli IS: *Intermediate System*).

Un primo problema consiste nella necessità di identificare in modo univoco ciascun sistema sulla rete. A tal scopo, ad ogni sistema è associato un *indirizzo* numerico (una serie di byte).

Tuttavia, nella maggior parte dei casi, è molto più comodo per l'utente riferirsi ad un sistema utilizzando un *nome* piuttosto che un indirizzo numerico. Il nome e l'indirizzo di un sistema hanno la stessa finalità, cioè quella di identificare in modo univoco un sistema all'interno della rete.

Occorre chiaramente mantenere una relazione biunivoca tra gli indirizzi e i nomi, e questo è più complesso di quanto si possa pensare. Infatti in una rete piccola è pensabile che ogni singolo calcolatore abbia un file che mantiene tale corrispondenza, ma al crescere delle dimensioni della rete è indispensabile dotarsi di una base dati distribuita, detta *name server*.

In modo analogo occorre mantenere delle tabelle di corrispondenza tra i nomi degli applicativi e i loro indirizzi (spesso detti anche identificatori di *porta*) che li identificano in modo univoco all'interno del sistema.

Quando un utente desidera connettersi ad un applicativo su di un dato elaboratore, egli lo richiede alla rete che, consultando la base dati, ricava l'identificativo della porta e l'indirizzo dell'elaboratore.

L'indirizzo dell'elaboratore destinatario del messaggio diventa l'elemento chiave con cui si determina l'instradamento più idoneo a raggiungere il sistema remoto. Un

primo controllo che il mittente effettua è quello di verificare se il destinatario è sulla stessa "rete": in questo caso la trasmissione può avvenire direttamente.

In caso contrario, è indispensabile un'operazione di *internetworking*: il mittente affida il pacchetto ad un IS che si occuperà di farlo giungere a destinazione.

#### 14.1.1 Tecniche di instradamento

Quando un IS riceve un pacchetto deve effettuare un'operazione di instradamento, cioè ritrasmettere il pacchetto verso il destinatario finale. La tecnica di instradamento scelta dipende dall'architettura di rete adottata. Esistono tre tecniche principali:

- *Routing by network address*. Un sistema è indirizzato scrivendo nel pacchetto il suo indirizzo, che deve essere univoco sulla rete. Ogni IS usa tale indirizzo come chiave di ricerca nella sua tabella di instradamento e determina lungo quale cammino il pacchetto debba essere ritrasmesso. Tale tecnica è usata nei transparent-bridge, in DECnet, in OSI-CLNS e in IP. È in generale adottata dai protocolli non connessi.
- *Label swapping*. È generalmente usata nei protocolli connessi e trova applicazioni in ATM (si veda il paragrafo 19.2) e in APPN (si veda il paragrafo 18.3). Ogni pacchetto è marcato con una label che serve come chiave in una tabella di instradamento sull'IS. L'IS, prima di ritrasmettere il pacchetto, sostituisce la label con una nuova label. Le label devono quindi essere univoche solo all'interno di un dato link. Se il protocollo è connesso, le label altro non sono che gli identificativi delle connessioni.
- *Source routing*. È una tecnica usata, ad esempio, dai bridge Token Ring (si veda il paragrafo 10.18). Nel source routing l'instradamento completo, cioè la lista degli IS da attraversare, è scritto nel pacchetto dal nodo mittente, che lo chiede ad un IS o lo scopre con meccanismi di "route location". Il source routing è utilizzato in APPN+.

Quanto sin qui descritto prescinde dal livello a cui viene effettuato l'instradamento. L'OSI delega la funzionalità di instradamento al livello 3 Network (o rete) e in particolare agli IS, spesso detti router. Altre architetture preferiscono effettuare l'instradamento a livello 2, utilizzando dei bridge.

Esempi di reti che instradano a livello 3 sono OSI, X.25, DECnet e IP. Esempi di reti con instradamento a livello 2 sono le BLAN (LAN estese con bridge), Frame Relay, e SMDS. Esistono poi reti in cui la distinzione tra i due livelli non è più così netta, ad esempio in HPR/APPN+.

Se si considerano i bridge source routing, è difficile motivare perché essi siano

considerati dei bridge e non dei router, visto che operano nella terza modalità precedentemente elencata. Sono stati definiti bridge principalmente perché il comitato di standardizzazione che se ne è occupato apparteneva al progetto IEEE 802 che tratta esclusivamente i livelli Fisico e Data Link.

Lungi dal voler entrare in questa diatriba, nel seguito del libro si farà riferimento al modello OSI classico e si assumerà, quando non diversamente specificato, che l'internetworking avvenga a livello 3 in modalità routing by network address.

#### 14.1.2 Indirizzi

Su reti ad accesso multiplo come le LAN si devono anche stabilire delle relazioni tra gli indirizzi di livello 2 sottolivello MAC e gli indirizzi di livello 3 (Network) per poter effettuare l'instradamento. Lo scopo dei due tipi di indirizzo è diverso:

- l'indirizzo di livello 2 MAC, come già visto nel paragrafo 5.6.7, serve a discriminare il destinatario finale di un pacchetto nell'ambito di una LAN;
- l'indirizzo di livello 3 serve invece ad identificare il destinatario finale del pacchetto nell'ambito dell'intera rete.

È ragionevole ipotizzare che un nodo abbia tanti indirizzi di livello 2 MAC quante sono le sue schede di rete locale ed un solo indirizzo di livello 3. Questo è vero nella maggior parte dei protocolli con un'unica eccezione di rilievo: il TCP/IP. Infatti il protocollo IP ha un indirizzo di livello 3 per ogni scheda di rete LAN o WAN (si veda paragrafo 16.5).

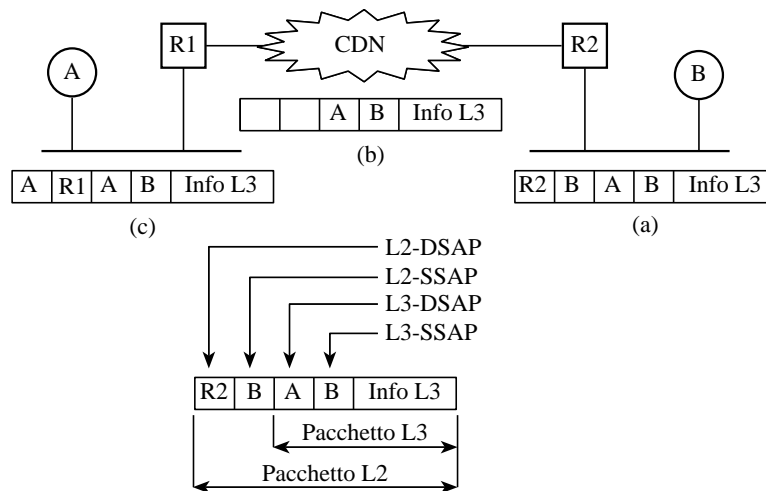
Nell'ambito di una LAN esistono vari metodi per mantenere le corrispondenze tra gli indirizzi di livello 2 MAC e gli indirizzi di livello 3: il più diffuso si basa sull'utilizzo del protocollo ARP (*Address Resolution Protocol*) descritto nel paragrafo 16.7.

Vediamo tramite l'esempio riportato in figura 14.1 il ruolo dei due tipi di indirizzi.

Si supponga di dover trasmettere un pacchetto dall'ES B all'ES A. La trasmissione avviene nelle seguenti quattro fasi, mediante tre pacchetti diversi identificati con (a), (b), e (c) in figura 14.1:

- B genera un pacchetto di livello 3 con L3-DSAP=A e L3-SSAP=B che rimarrà immutato sino a destinazione. B verifica se A è sulla sua stessa LAN e poiché ciò non è vero invia il messaggio a R2 specificando L2-DSAP=R2 e L2-SSAP=B (pacchetto (a)).
- L'IS R2 riceve il pacchetto (a) ed utilizza le sue tabelle di instradamento per decidere di ritrasmettere il messaggio sul Canale Diretto Numerico (CDN). In questo caso, poiché ci troviamo in presenza di un canale punto-punto, non è necessaria la presenza di un indirizzo a livello 2 nel pacchetto (b).

- R1 riceve il pacchetto (b) e decide che deve trasmetterlo ad A tramite la LAN. Usando, ad esempio, un algoritmo di ARP ricava l'indirizzo di livello 2 di A a partire dal suo indirizzo di livello 3 e quindi effettua la trasmissione del pacchetto (c).
- A riceve il pacchetto (c) e, poiché lo L3-DSAP è uguale al suo indirizzo di livello 3, non lo inoltra ulteriormente sulla rete, ma lo passa ai suoi livelli superiori.



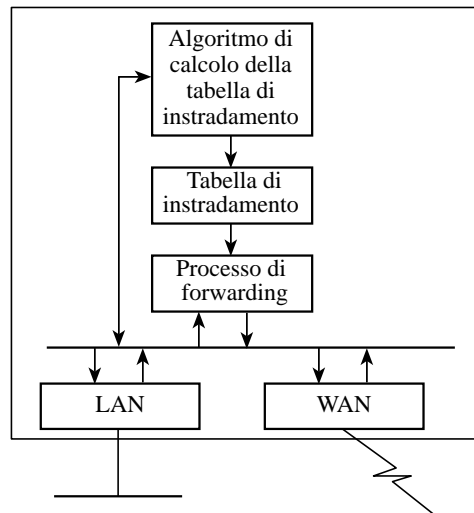
**Fig. 14.1** - Indirizzi MAC e Network.

### 14.1.3 L'instradamento

Esaminiamo più nel dettaglio la funzionalità di instradamento dei router, cioè degli Intermediate System operanti a Livello 3, con l'ausilio della figura 14.2.

Un pacchetto viene ricevuto dal router tramite una sua scheda di LAN o di WAN che gestisce un protocollo di livello 2. La scheda verifica se il pacchetto è destinato al router (condizione sempre vera su linee punto-punto, ma da verificarsi tramite gli indirizzi MAC sulle LAN) e in caso affermativo lo passa al processo di forwarding. Questo determina su quale linea deve essere ritrasmesso il pacchetto consultando le tabelle di instradamento.

Le tabelle di instradamento possono essere scritte manualmente dal gestore della rete oppure calcolate automaticamente da un opportuno algoritmo. Tale algoritmo opera scambiando tra gli IS informazioni relative alla topologia e allo stato della rete.



**Fig. 14.2** - Architettura di un router.

#### 14.1.4 Neighbor Greetings

Un altro problema è quello dei "neighbor greetings", cioè del fatto che gli IS collegati ad una LAN devono conoscere gli ES collegati alla stessa LAN e viceversa. Questo è indispensabile per due motivi:

- gli IS devono conoscere gli ES per inserirli nelle tabelle di instradamento e propagare l'informazione della loro raggiungibilità agli altri IS;
- gli ES devono conoscere gli IS presenti sulla LAN per sapere a chi inviare i messaggi non destinati a nodi collegati alla stessa LAN.

La soluzione a quest'ultimo problema deve essere tale da ammettere LAN prive di router, LAN con un solo router o LAN con più router.

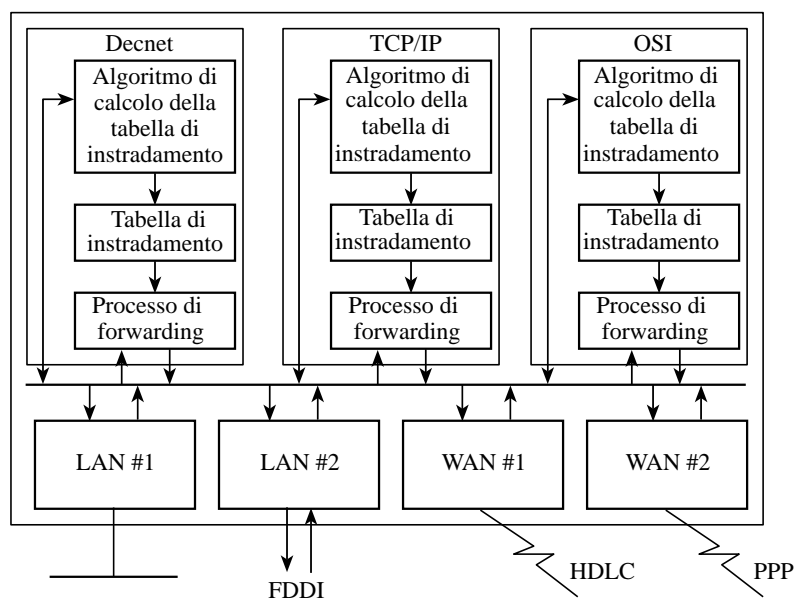
#### 14.1.5 L'internetworking multiprotocollo

A differenza dei bridge che sono assolutamente trasparenti al protocollo di livello 3, nel senso che ignorano il contenuto del campo Info di livello 2, i router operano a livello 3 e quindi utilizzano per l'internetworking tutte le informazioni contenute nella busta di livello 3.

Le buste di livello 3 delle diverse architetture di rete (es. TCP/IP, DECnet, OSI)

hanno formati e contenuti tra loro incompatibili, quindi i router sono progettati facendo riferimento ad un dato formato del pacchetto di livello 3 e quindi ad una data architettura di rete.

Chiaramente, disporre oggi di router in grado di trattare una sola architettura di rete sarebbe inaccettabile e da alcuni anni sono disponibili sul mercato router multiprotocollo che hanno una struttura più complessa, riportata in figura 14.3.



**Fig. 14.3** - Router multiprotocollo.

Da una prima analisi si vede che il modulo di instradamento è replicato per ogni protocollo trattato. Molto spesso esiste anche un modulo di bridging per trattare quei protocolli che non hanno un livello 3 (es: LAT, NetBeui, MOP) e che quindi non sarebbero gestibili dai router. Quando un router multiprotocollo realizza anche la funzionalità di bridging viene detto *brouter*.

Un brouter è in grado, protocollo per protocollo, di:

- non trattare il protocollo e quindi scartare eventuali pacchetti appartenenti a quel protocollo;
- trattare il protocollo tramite una modalità di bridging, quindi a livello 2;
- trattare il protocollo tramite una modalità di routing, quindi a livello 3.

L'ultima opzione è possibile solo se l'architettura di rete ha un livello 3 e quindi è instradabile. Come già accennato precedentemente, esistono architetture di rete

quali il LAT (si veda l'appendice B, paragrafo B.5), il Netbeui, il MOP e altre che, essendo state progettate pensando ad un utilizzo esclusivamente su rete locale, non hanno un livello 3. Se si vuole poter utilizzare tali architetture anche su base geografica l'unica possibilità è l'impiego della funzionalità di bridging.

Possiamo ora precisare meglio il ruolo degli indirizzi di livello 2. Nell'esempio precedente gli indirizzi di livello 2 citati sono indirizzi di livello 2 MAC, poiché servono a gestire la trasmissione sulla LAN. Gli indirizzi di livello 2 LLC non hanno come scopo l'indirizzamento di un nodo, ma di una architettura di rete all'interno di un nodo.

Supponiamo che un pacchetto venga ricevuto dal router multiprotocollo di figura 14.3 sulla scheda LAN#1. La scheda tramite l'indirizzo L2-MAC determina se il pacchetto è destinato al router e quindi, tramite l'indirizzo L2-LLC, determina a quale modulo di instradamento passare il pacchetto (nell'esempio, DECnet, TCP/IP o OSI). Il modulo di instradamento, ricevuto il pacchetto, determina tramite l'indirizzo di livello 3 su quale linea e a quale nodo ritrasmettere il pacchetto.

A questo punto il lettore dovrebbe essersi fatto un'idea dei molti problemi da affrontare per instradare un messaggio su una rete. Altri problemi sono la gestione di topologie complesse, la convivenza di più mezzi trasmissivi diversi, l'organizzazione gerarchica della rete, ecc.

I paragrafi seguenti analizzano questi problemi in modo più approfondito.

## 14.2 IL LIVELLO NETWORK

Una rete di calcolatori si realizza interconnettendo con vari tipi di tecnologie (linee telefoniche commutate, CDN, X.25, ISDN, SMDS, Frame Relay, LAN, ATM) un insieme di IS (commutatori di pacchetto) normalmente chiamati *router* (figura 14.4).

Gli IS si occupano di instradare i messaggi sulla rete ed operano al livello 3 del modello di riferimento OSI, cioè a *livello Network*.

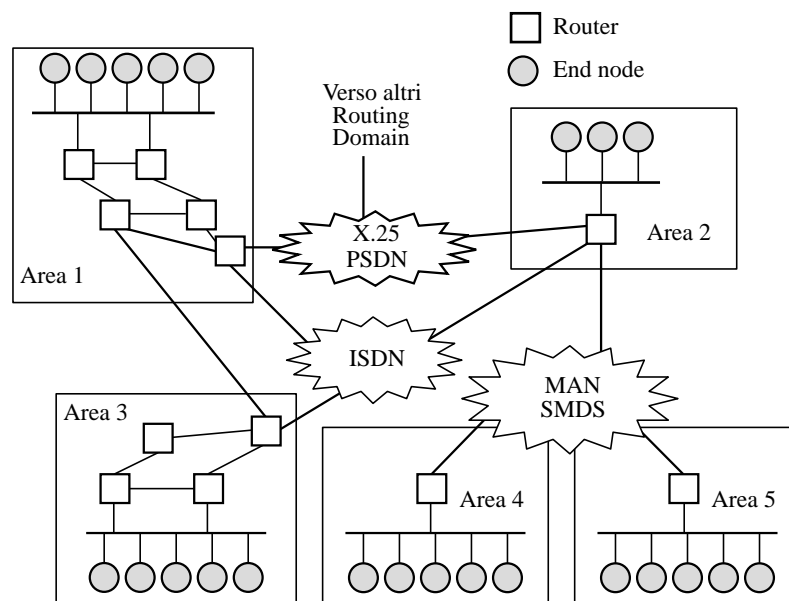
Alcuni router sono a volte detti anche IMP (*Interface Message Processor*) o IP Gateway\*.

Il termine *nodo* viene spesso adottato in luogo del più corretto termine *sistema*. Un nodo è un dispositivo di rete che contiene al suo interno almeno i livelli Fisico, Data Link e Network.

---

\* Il termine IP Gateway di derivazione TCP/IP è particolarmente infelice in quanto OSI assegna un altro significato al termine gateway e per questo motivo se ne sconsiglia l'uso e in questo libro non verrà adottato.

Gli ES (detti anche end node) sono quei nodi che agiscono come mittente e destinatario finale dei dati e tipicamente implementano tutti e sette i livelli del modello di riferimento OSI. Essi hanno un livello 3 molto semplice in quanto non devono preoccuparsi delle problematiche di instradamento.



**Fig. 14.4** - Esempio di WAN con router.

Molto spesso a livello 3, oltre al protocollo per trasportare i dati di utente, sono definiti anche uno o più protocolli ausiliari per il neighbor greetings e per permettere ai router di scambiarsi informazioni di instradamento.

La necessità di realizzare funzionalità di gestione (management) del router rende necessaria la presenza negli IS anche di protocolli specifici per la gestione\*, che sono collocati nei livelli superiori al terzo.

Per instradare i pacchetti, il livello network si basa sull'indirizzo del destinatario finale e sulle tabelle d'instradamento presenti negli IS. Le tabelle d'instradamento possono essere scritte manualmente (soluzione adottata nella rete SNA e a volte anche

\* Il protocollo di gestione, che è ormai oggi uno standard "de facto" e con cui si opera normalmente sui router e sulla altre apparecchiature di rete, è il SNMP (*Simple Network Management Protocol*), che si basa su UDP/IP (si veda il paragrafo 16.12.9).



in quella TCP/IP) o calcolate automaticamente mediante algoritmi che apprendono la topologia della rete e si adattano ai suoi cambiamenti, determinando instradamenti alternativi in caso di guasti.

Il livello Network può offrire servizi di tipo connesso (*connection oriented*) e non connesso (*connectionless*). L'implementazione dei servizi connessi (CONS) a questo livello avviene tramite i circuiti virtuali. Il CCITT e le PTT sono forti sostenitori di questa filosofia, che è realizzata in reti dati a pacchetto, quali quelle conformi ai protocolli X.25 e Frame Relay.

I servizi non connessi (CLNS) a questo livello prendono anche il nome di servizi di *datagram*; essi sono adottati nelle reti proprietarie quali DECnet e TCP/IP, proposti dall'ISO per le reti OSI nello standard ISO 8473 e realizzati da alcune PTT in reti come quelle conformi allo standard SMDS (si veda paragrafo 13.6).

La tabella 14.1 riassume alcune proprietà dei servizi *connection oriented* e *connectionless*, rispetto ad una serie di caratteristiche, che possiamo considerare ai fini di un confronto.

Caratteristica	Connection Oriented	Connectionless
Setup Iniziale	Richiesto	Impossibile
Destination Address	Durante il setup	Ad ogni pacchetto
Ordine dei Pacchetti	Garantito	Non garantito
Controllo Errori	A livello Network	A livello trasporto
Controllo di Flusso	Fornito dal Network	Non fornito dal Network
Negoziazione delle Opzioni	Sì	No
Uso di Connection Identifier	Sì	No

**Tab. 14.1** - Confronto tra L3 connesso e non connesso.

Per quanto riguarda il controllo dell'errore, bisogna ricordare che, anche se esso viene implementato al livello 3, solitamente l'affidabilità non è considerata sufficiente ai livelli superiori e quindi il livello 4 è comunque connesso.

### 14.3 ALGORITMI DI INSTRADAMENTO

La scelta di un algoritmo di instradamento è difficile, in quanto esistono più criteri di ottimalità spesso contrastanti, ad esempio minimizzare il ritardo medio di ogni

pacchetto o massimizzare l'utilizzo delle linee.

Occorre che la scelta sia preceduta dalla definizione di criteri misurabili. Occorre cioè introdurre dei parametri metrologici in base ai quali misurare le caratteristiche di un cammino per scegliere, ad esempio, il migliore tra due cammini alternativi.

Gli unici due parametri universalmente accettati sono:

- il numero di salti effettuati (*hop*), cioè il numero di IS attraversati lungo un cammino;
- il *costo*, cioè la somma dei costi di tutte le linee attraversate lungo un cammino.

Entrambi questi parametri sono di demerito, in quanto il costo di una linea è assegnato in modo inversamente proporzionale alla velocità della linea stessa, e gli hop indicano router attraversati e quindi ritardi introdotti.

Metriche che tengano in considerazione il carico della rete sono più difficili da mettere a punto, in quanto portano facilmente a situazioni di routing instabile. Le tecniche più moderne consentono al più di operare un *load splitting* (bilanciamento del traffico) tra cammini paralleli, eventualmente attivando circuiti commutati, quali quelli forniti dalla rete ISDN (si veda paragrafo 12.6) in presenza di un guasto (ad esempio, funzionalità di backup di un CDN) o per gestire un eccesso di traffico su di un link (traffico di trabocco).

La scelta dell'algoritmo di instradamento ottimale è anche complicata dalle limitate risorse di memoria e CPU disponibili oggi sui router, specialmente se confrontate con la complessità delle reti ed in particolare con l'elevato numero di nodi collegabili con una topologia qualsiasi. Algoritmi troppo complessi, operanti su reti molto grandi, potrebbero richiedere tempi di calcolo inaccettabili.

I router attualmente installati sulle reti vanno dai vecchi router con CPU da 1 MIPS e con memoria da 1 Mbyte, ai più moderni router con più CPU RISC da 25 MIPS e memoria da 16 Mbyte. Uno dei fattori che limita la produzione di router sempre più potenti è soprattutto il costo, che deve mantenersi ragionevolmente contenuto.

Riassumendo, le caratteristiche che in generale si richiedono ad un algoritmo di routing sono:

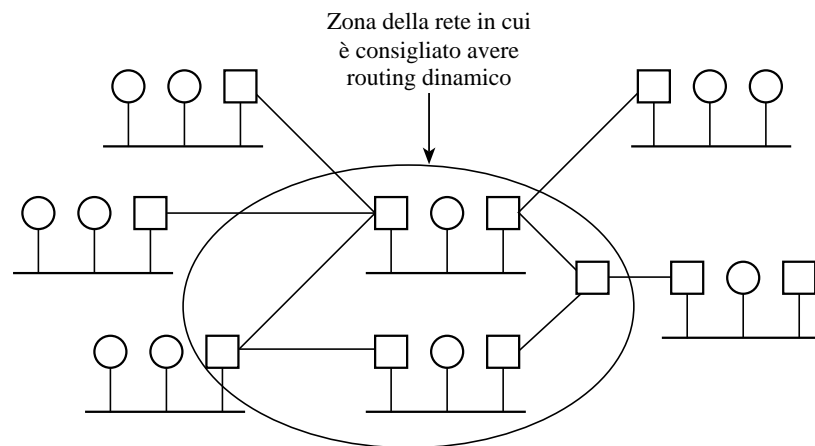
- *semplicità* dell'algoritmo, poiché i router hanno CPU e memoria finite e devono impiegare la maggior parte del loro tempo a instradare pacchetti, non a calcolare nuove tabelle di instradamento;
- *robustezza* e adattabilità alle variazioni di topologia; non deve esistere nessun presupposto né vincolo sulla topologia di rete, che deve poter essere modificata dinamicamente senza interrompere il funzionamento della rete;
- *ottimalità* nella scelta dei cammini, rispetto soprattutto ai due criteri elencati precedentemente;

- *stabilità*: a fronte di una rete stabile l'algoritmo deve sempre convergere velocemente ad un instradamento stabile, cioè non deve modificare le tabelle di instradamento se non a fronte di una variazione di topologia;
- *equità*: nessun nodo deve essere privilegiato o danneggiato.

Gli algoritmi di routing si dividono in due gruppi: *non adattativi* (statici e deterministici) e *adattativi* (dinamici e non deterministici). I primi utilizzano criteri fissi di instradamento, mentre gli altri calcolano le tabelle di instradamento in funzione della topologia della rete e dello stato dei link.

Sono algoritmi del primo gruppo il *fixed directory routing* e il *flooding*, mentre appartengono al secondo gruppo il *routing centralizzato*, il *routing isolato* e il *routing distribuito*.

Entrambi i gruppi hanno la loro ragione di esistere, in zone diverse della rete, come evidenziato in figura 14.5. Infatti, se per sfruttare al meglio le magliature della rete è indispensabile avere algoritmi di routing dinamico, nelle zone più periferiche della rete con topologia ad albero, cioè con un solo cammino che le interconnette al resto della rete, un routing statico può risultare più semplice e non presentare svantaggi.



**Fig. 14.5** - Routing statico e dinamico.

Gli algoritmi di più moderna concezione sono quelli distribuiti, che si suddividono ulteriormente in due famiglie: *distance vector* e *link state packet*.

## 14.4 ALGORITMI STATICI

### 14.4.1 Fixed directory routing

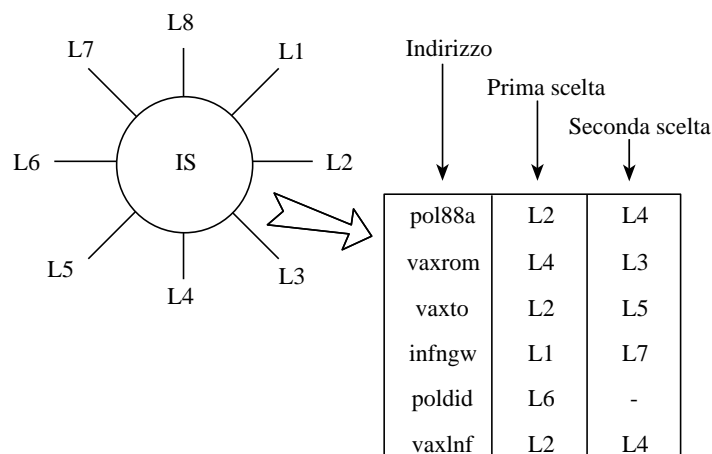
Il fixed directory routing prevede che ogni nodo abbia una tabella di instradamento che metta in corrispondenza il nodo da raggiungere con la linea da usare, e che tale tabella sia scritta manualmente dal gestore della rete nel router tramite un'operazione di management.

Il gestore ha il totale controllo dei flussi di traffico sulla rete, ma è necessario un suo intervento manuale per il reinstradamento di detti flussi in presenza di guasti. Questo approccio è spesso utilizzato in TCP/IP per le parti non magliate della rete e le regole di instradamento specificate su ogni singolo router prendono il nome di *route statiche*.

Esiste una variante, detta quasi-statica, che adotta tabelle con più alternative da scegliere secondo un certo ordine di priorità, in funzione dello stato della rete. Questo approccio, che consente di avere cammini alternativi in caso di guasto, è adottato, ad esempio, dalla rete SNA.

Occorre comunque evidenziare che la gestione manuale delle tabelle risulta molto complessa e difficoltosa, soprattutto per reti di grandi dimensioni.

In figura 14.6 vediamo un esempio di tabella di instradamento per fixed directory routing, che fornisce due scelte possibili: per ragioni esemplificative, al posto dell'indirizzo vi è un nome simbolico, poiché il formato dell'indirizzo può variare molto secondo il tipo rete considerata.



**Fig. 14.6** - Esempio di fixed directory routing.

#### 14.4.2 Flooding

Il flooding è un altro algoritmo non adattativo, in cui ciascun pacchetto in arrivo viene ritrasmesso su tutte le linee, eccetto quella su cui è stato ricevuto.

Concepito per reti militari a prova di sabotaggio, se realizzato nel modo sopra descritto massimizza la probabilità che il pacchetto arrivi a destinazione, ma induce un carico elevatissimo sulla rete.

Si può cercare di ridurre il carico utilizzando tecniche di *selective flooding*, in cui i pacchetti vengono ritrasmessi solo su linee selezionate.

Un primo esempio, senza applicazioni pratiche, è l'algoritmo *random walk* che sceglie in modo pseudo-casuale su quali linee ritrasmettere il pacchetto.

Una miglioria più efficace si ha scartando i pacchetti troppo vecchi, cioè quelli che hanno attraversato molti router: a tal scopo nell'header del pacchetto viene inserito un age-counter.

Un'ultima miglioria, ancora più significativa, consiste nello scartare un pacchetto la seconda volta che passa in un nodo: in tal modo si realizza una tecnica per trasmettere efficientemente la stessa informazione a tutti i nodi, qualsiasi sia la topologia. Lo svantaggio è che bisogna memorizzare tutti i pacchetti su ogni nodo per poter verificare se sono già passati.

Una tecnica di selective flooding è utilizzata per il calcolo delle tabelle di instradamento dal protocollo IS-IS (ISO 10598).

#### 14.5 ALGORITMI ADATTATIVI

Gli algoritmi di instradamento adattativi sono quelli in cui le tabelle dipendono dalle informazioni raccolte sulla topologia della rete, sul costo dei cammini e sullo stato degli elementi che la compongono.

Gli algoritmi adattativi possono essere centralizzati (in un unico punto della rete vengono raccolte e analizzate tutte le informazioni, e calcolate le tabelle), isolati (ogni router è indipendente dagli altri) o distribuiti (i router cooperano al calcolo delle tabelle).

##### 14.5.1 Routing centralizzato

Il routing centralizzato è tra i metodi adattativi quello che più si avvicina al fixed directory routing. Presuppone l'esistenza di un RCC (*Routing Control Center*) che conosce la topologia della rete, riceve da tutti i nodi informazione sul loro stato e su quello dei collegamenti, calcola le tabelle di instradamento e le distribuisce.

È un metodo che consente una gestione della rete molto accurata, in quanto permette di calcolare le tabelle anche con algoritmi molto sofisticati, ma implica l'esistenza di un unico gestore, ipotesi questa oggi molto spesso non realistica.

Il RCC, per ragioni di affidabilità, deve essere duplicato e la porzione di rete intorno ad esso è soggetta ad un elevato volume di traffico di servizio: informazioni di stato che arrivano al RCC e tabelle di instradamento che escono dal RCC.

In caso di guasti gravi possono verificarsi situazioni in cui il RCC perde il contatto con una parte periferica della rete e si verificano quindi degli aggiornamenti parziali di tabelle che possono determinare situazioni di loop.

Questo metodo è usato con successo nella rete TymNet, che è un'importante rete X.25 internazionale.

#### 14.5.2 Routing isolato

Il routing isolato è l'opposto di quello centralizzato, visto che non solo non esiste un RCC, ma ogni IS si calcola in modo indipendente le tabelle di instradamento senza scambiare informazioni con gli altri IS.

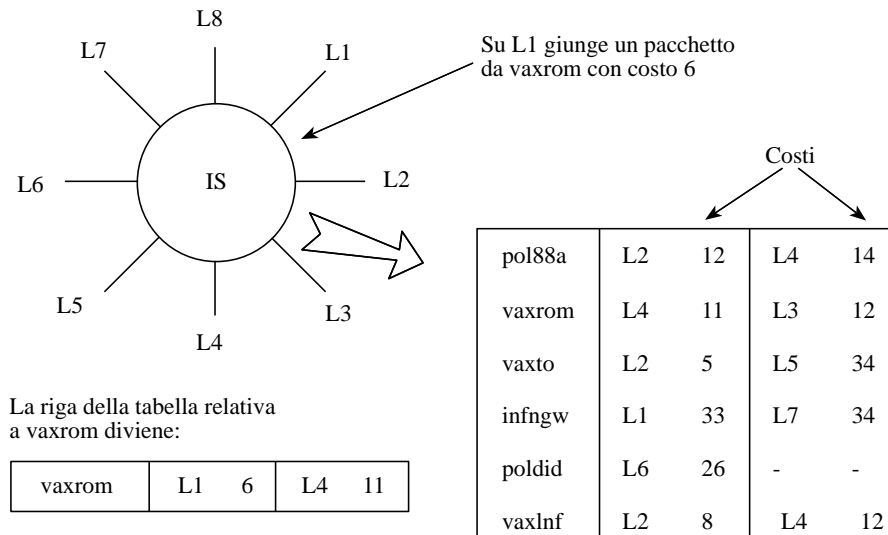
Esistono due algoritmi di routing isolato riportati in letteratura: *hot potato* e *backward learning*.

Il primo è famoso solo per il suo nome simpatico (*hot potato*). Ogni IS considera un pacchetto ricevuto come una patata bollente e cerca di liberarsene nel minor tempo possibile, ritrasmettendo il pacchetto sulla linea con la coda di trasmissione più breve.

Il metodo di *backward learning* è invece utilizzato per calcolare le tabelle di instradamento nei bridge conformi allo standard IEEE 802.1D (si veda paragrafo 10.6). L'IS acquisisce una conoscenza indiretta della rete analizzando il traffico che lo attraversa: se riceve un pacchetto proveniente dal nodo A sulla linea L3, il backward learning impara che A è raggiungibile attraverso la linea L3.

È possibile migliorare il backward learning inserendo nell'header del pacchetto un campo di costo inizializzato a zero dalla stazione mittente ed incrementato ad ogni attraversamento di un IS. In tale modo gli IS possono mantenere più alternative per ogni destinazione, ordinate per costo crescente.

Tale situazione è mostrata in figura 14.7 in cui l'IS mantiene due alternative (entry) per ogni destinazione nella tabella di instradamento. Quando da *vaxrom* giunge un pacchetto con costo 6, la riga relativa a *vaxrom* viene aggiornata in quanto si è scoperto un cammino più conveniente di uno già noto.



**Fig. 14.7** - Esempio di routing isolato.

Il limite di questo metodo consiste nel fatto che gli IS imparano solo le migliori e non i peggioramenti nello stato della rete: infatti se cade un link e si interrompe un cammino, semplicemente non arrivano più pacchetti da quel cammino, ma non giunge all'IS nessuna informazione che il cammino non è più disponibile.

Per tale ragione occorre limitare temporalmente la validità delle informazioni presenti nelle tabelle di instradamento: ad ogni entry viene associata una validità temporale che viene inizializzata ad un dato valore ogni volta che un pacchetto in transito conferma l'entry, e decrementata automaticamente con il passare del tempo. Quando la validità temporale di un'entry giunge a zero, questa viene invalidata ed eliminata dalla tabella di instradamento.

Qualora ad un IS giunga un pacchetto per una destinazione ignota, l'IS ne fa il flooding.

Il backward learning può generare loop su topologie magliate, per cui, ad esempio nei bridge, lo si integra con l'algoritmo di spanning tree per ridurre la topologia magliata ad un albero (si veda paragrafo 10.18).

### 14.5.3 Routing distribuito

Il routing distribuito è indubbiamente quello di maggior interesse per la soluzione dei problemi di internetworking. Esso si pone come una scelta di compromesso tra i

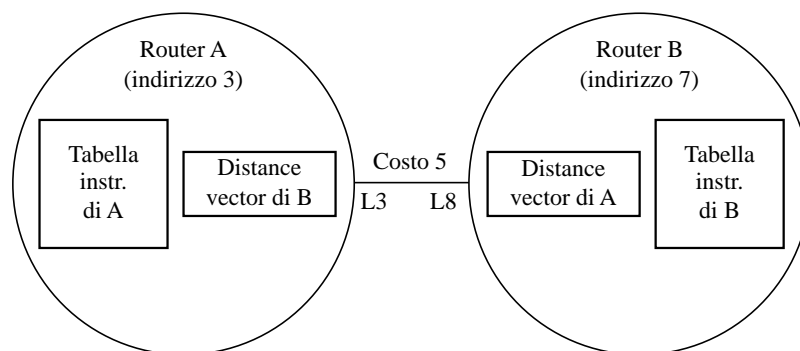
due precedenti: non esiste un RCC, ma le sue funzionalità sono realizzate in modo distribuito da tutti gli IS della rete, che a tal scopo usano un protocollo di servizio per scambiare informazioni tra loro ed un secondo protocollo di servizio per scambiare informazioni con gli ES. Tali protocolli vengono detti di servizio in quanto non veicolano dati di utente, che sono gestiti da un terzo protocollo, ma solo informazioni utili al calcolo delle tabelle di instradamento e al neighbor greetings.

Le tabelle di instradamento vengono calcolate a partire dai due parametri di ottimalità precedentemente descritti: costo e hop. Il costo di ciascuna linea di ciascun router è un parametro che viene impostato dal network manager tramite il software di gestione dei router stessi.

Gli algoritmi di routing distribuito sono oggi adottati da DECnet, TCP/IP, OSI, APPN, ecc., e si suddividono ulteriormente in due famiglie: algoritmi distance vector e algoritmi link state packet. Vista l'importanza di tali algoritmi, essi vengono descritti nei due appositi paragrafi che seguono.

#### 14.6 ALGORITMI DI ROUTING DISTANCE VECTOR

L'algoritmo distance vector è anche noto come algoritmo di Bellman-Ford. Per realizzare tale algoritmo ogni router mantiene, oltre alla tabella di instradamento, una struttura dati, detta *distance vector* per ogni linea. Il distance vector associato a ciascuna linea contiene informazioni ricavate dalla tabella di instradamento del router collegato all'altro estremo della linea (si veda figura 14.8).



**Fig. 14.8** - Router distance vector.

Il calcolo delle tabelle di instradamento avviene tramite un processo di fusione (merge) di tutti i distance vector associati alle linee attive di un router. Tutte le volte



che un router calcola una nuova tabella di instradamento, la invia agli IS adiacenti sotto forma di distance vector.

La tabella di instradamento è un insieme di quadruplette {indirizzo, hop, costo, linea} che contiene per ogni nodo della rete (indirizzo), sia esso un IS o un ES, l'informazione relativa al cammino migliore per raggiungere tale nodo in termini di numero di IS da attraversare (hop), somma dei costi delle linee da attraversare (costo) e linea su cui ritrasmettere il messaggio (linea). In figura 14.9 è riportata la tabella di instradamento del router A dell'esempio precedente.

Indirizzo	Hops	Costo	Linea
1	5	25	3
2	3	20	2
3	0	0	0
4	2	15	3
5	7	55	1
6	4	23	1
7	1	5	3
...	...	...	...

**Fig. 14.9** - Tabella di instradamento di A.

Appare evidente che A ha indirizzo 3, in quanto esso appare raggiungibile in zero hop e con costo zero.

Ciascun router apprende tramite un protocollo di neighbor greetings (paragrafo 14.8) le informazioni relative ai nodi adiacenti, siano essi IS o ES, e le inserisce nella tabella di instradamento.

Quando un IS modifica la sua tabella di instradamento per una delle ragioni descritte nel seguito, esso invia a tutti gli IS adiacenti, cioè collegati da un cammino fisico diretto (solo a quelli adiacenti, non a tutti gli IS della rete) il suo distance vector che ricava dalle prime tre colonne della sua tabella di instradamento e che risulta quindi un insieme di triplette del tipo {indirizzo, hop, costo}. Un esempio di distance vector reale è riportato in appendice B, paragrafo B.4.4.

Quando un router riceve un distance vector da un router adiacente, prima di memorizzarlo, somma uno a tutti i campi hop e il costo della linea da cui ha ricevuto il distance vector a tutti i campi costo.

Nell'esempio precedente il router B che ha indirizzo 7, come appare evidente dalla tabella di instradamento di A, quando riceve il distance vector di A, gli aggiorna i campi hop e costo, e lo memorizza nella sua struttura dati locale come indicato in figura 14.10.

Indirizzo	Hops	Costo
1	6	30
2	4	25
3	1	5
4	3	20
5	8	60
6	5	28
7	2	10
...	...	...

**Fig. 14.10** - Distance vector di A memorizzato in B.

Quando un router memorizza un distance vector nella sua struttura dati locale, verifica se sono presenti variazioni rispetto al distance vector precedentemente memorizzato: in caso affermativo ricalcola le tabelle di instradamento fondendo (*merge*) tutti i distance vector delle linee attive. Analoga operazione di ricalcolo avviene quando una linea passa dallo stato ON allo stato OFF o viceversa. Molte implementazioni ricalcolano anche le tabelle di instradamento periodicamente.

La fusione avviene secondo il criterio di convenienza del costo: a parità di costo secondo il minimo numero di hop e a parità di hop con scelta casuale. In figura 14.11 è riportato un esempio di calcolo della tabella di instradamento. La linea L0 indica il router stesso.

Se la tabella di instradamento risulta variata rispetto alla precedente, il distance vector relativo viene inviato ai router adiacenti. Alcune implementazioni di protocolli distance vector inviano anche i distance vector periodicamente, ad esempio il RIP (si veda il paragrafo 16.9.1) invia il distance vector ogni 30 secondi.

La modalità operativa descritta crea un fenomeno simile a quello di una pietra che cade in un catino di acqua. La pietra è una variazione dello stato della rete, il catino è la rete, le onde generate dalla caduta della pietra sono i distance vector che si dipartono dal luogo di impatto, arrivano ai bordi della rete, si specchiano e tornano verso il centro e ancora verso la periferia e poi verso il centro, con un moto che si ripete più volte prima di giungere a stabilità (acqua ferma).

Il vantaggio di questo algoritmo è la facilità di implementazione. Gli svantaggi sono:

- la complessità elevata, esponenziale nel caso peggiore e normalmente compresa tra  $O(n^2)$  e  $O(n^3)$ . Questo rende improponibile l'utilizzo di tale algoritmo per reti con più di 1000 nodi, a meno che non venga adottato un partizionamento gerarchico come descritto in il paragrafo 14.9;

- la lenta convergenza ad un instradamento stabile. Infatti l'algoritmo converge con una velocità proporzionale a quella del link più lento e del router più lento presenti nella rete;
- la difficoltà di capirne e prevederne il comportamento su reti grandi, poiché nessun nodo ha la mappa della rete.

Questo algoritmo è usato in DECnet fase IV e in alcune realizzazioni TCP/IP (protocolli RIP e IGRP).

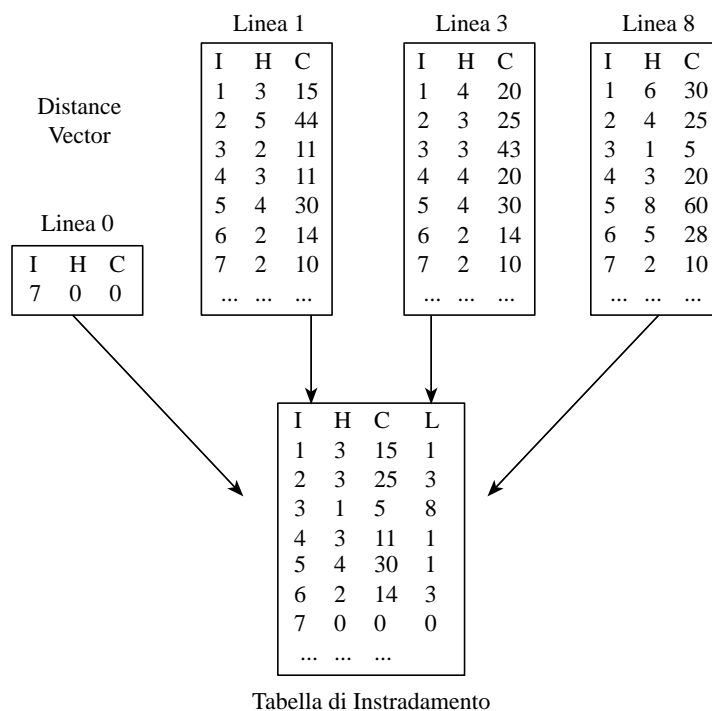


Fig. 14.11 - Fusione di distance vector in una tabella di instradamento.

## 14.7 ALGORITMI DI ROUTING LINK STATE PACKET

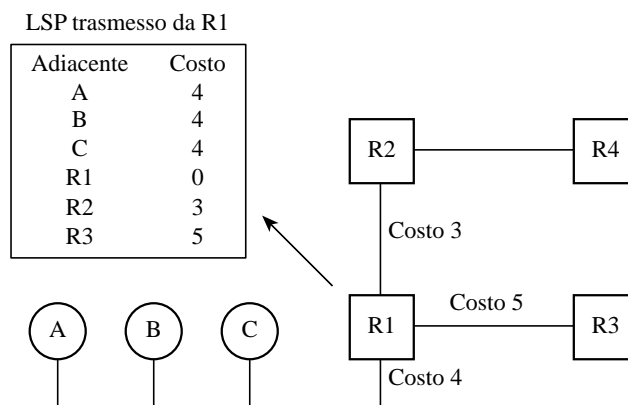
L'algoritmo link state packet assume che ogni IS disponga della mappa completa della rete su cui calcolare gli instradamenti ottimali utilizzando l'algoritmo di Dijkstra o *Shortest Path First* (SPF).

La mappa della rete non è scritta nei router dal sistema di gestione (sarebbe impraticabile per reti grandi), ma è costruita direttamente dai router tramite l'utilizzo

di *Link State Packet (LSP)*.

Ogni router, tramite protocolli di neighbor greetings, apprende quali nodi sono a lui adiacenti e lo comunica agli altri router inviando un LSP che descrive tali adiacenze.

La figura 14.12 riporta un esempio di rete e del relativo LSP inviato dal router R1. Si noti che il LSP non contiene una entry per il router R4 in quanto non adiacente a R1.



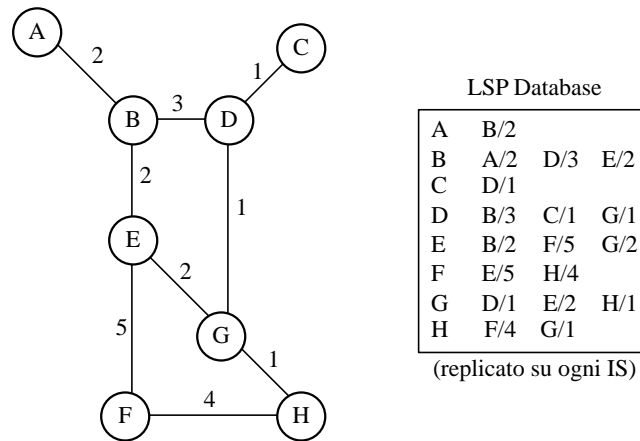
**Fig. 14.12** - Trasmissione di LSP.

I Link State Packet sono trasmessi in selective flooding (si veda il paragrafo 14.4.2) a tutti gli IS della rete, secondo una modalità dettagliata nel seguito. Ogni IS contiene un LSP database in cui memorizza il LSP più recente generato da ogni IS.

Il LSP database è una rappresentazione del grafo della rete data come matrice delle adiacenze (si veda [1]). Si osservi che per definizione il LSP database deve essere esattamente lo stesso su tutti gli IS della rete. La figura 14.13 riporta un ipotetico grafo di rete, i cui vertici sono i nodi della rete (ES o IS) e i cui archi sono le linee con i costi associati, e il relativo LSP database.

Il LSP database, rappresentando la mappa della rete con i costi associati, è l'informazione necessaria e sufficiente affinché un router possa calcolare le sue tabelle di instradamento. Si noti la differenza con il distance vector: in quel caso i router cooperano direttamente per calcolare le tabelle di instradamento, qui i router cooperano per mantenere aggiornata la mappa della rete, poi ogni router calcola la propria tabella di instradamento in modo autonomo.

Il calcolo delle tabelle equivale al calcolo dello spanning tree di tipo Shortest Path First e si effettua con l'algoritmo di Dijkstra.



**Fig. 14.13** - Grafo della rete e LSP database.

Ogni nodo ha a disposizione il grafo pesato della rete ed assegna a tutti gli altri nodi un'etichetta che rappresenta il costo massimo per la raggiungibilità del nodo in esame; l'algoritmo di calcolo modifica tali etichette cercando di minimizzarle e di renderle permanenti.

Le strutture dati coinvolte sono:

- l'insieme  $V$  dei vertici del grafo (i nodi della rete);
- il costo  $C[i, j]$  della connessione diretta da  $i$  a  $j$  (assunto infinito se tale connessione non esiste);
- il costo  $D[i]$  del cammino dal vertice 1 (assunto come radice dell'albero degli instradamenti, è cioè il nodo che sta effettuando il calcolo della tabella) al vertice  $i$ .

L'algoritmo inizializza  $D[i]=C[1, i]$  e poi iterativamente cerca di minimizzare  $D[i]$ , mantenendo un insieme  $S$  in cui memorizza quali vertici hanno già un valore definitivo (non ulteriormente riducibile).

La figura 14.14 riporta lo pseudo-codice dell'algoritmo e la figura 14.15 un esempio di applicazione.

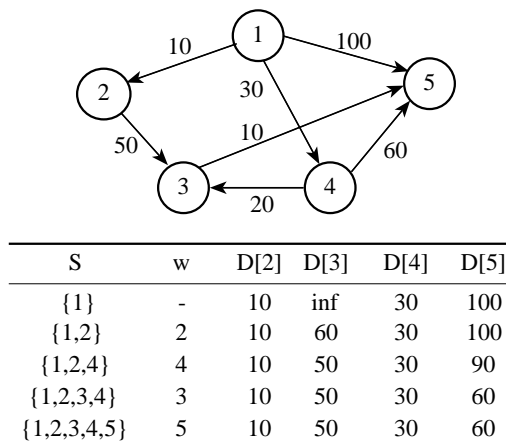
```

procedure Dijkstra; {calcola il costo minimo tra tutti i cammini dal
nodo 1 a tutti gli altri nodi}
begin
  S := {1};
  for i := 2 to n do D[i] := C[1,i]; {inizializza D}
  for i := 1 to n-1 do
  begin
    si scelga un nodo w in V-S tale che D[w] sia minimo;
    si aggiunga w a S;
    per ogni vertice v in V-S do D[v] := min(D[v], D[w] + C[w,v])
  end
end; {Dijkstra}

```

**Fig. 14.14** - Algoritmo di Dijkstra.

L'applicazione dell'algoritmo di Dijkstra, all'esempio di figura 14.13, ad opera dell'IS B produce l'albero di instradamento di figura 14.16a, mentre quello ad opera dell'IS F produce l'albero di instradamento di figura 14.16b.



**Fig. 14.15** - Esempio di applicazione dell'algoritmo di Dijkstra.

I forwarding database risultanti, che verranno utilizzati dai router durante la normale operatività per inoltrare i pacchetti, sono riportati in figura 14.17.

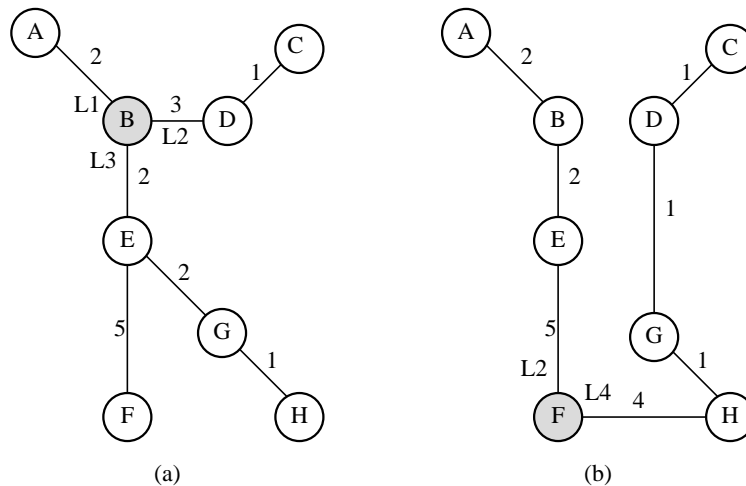


Fig. 14.16 - Alberi di instradamento.

Forwarding Table di B		Forwarding Table di F	
A	L1	A	L2
C	L2	B	L2
D	L2	C	L4
E	L3	D	L4
F	L3	E	L2
G	L3	G	L4
H	L3	H	L4

(a)

(b)

Fig. 14.17 - Forwarding table.

La complessità dell'algoritmo link state packet è pari a  $L \cdot \log(N)$ , dove  $L$  è il numero di link e  $N$  è il numero di nodi, ma poiché i costi dei link sono numeri interi piccoli, si riescono a realizzare strutture sofisticate che fanno tendere questo valore a  $N$ . Ad esempio, su un router da 1 MIPS inserito su una rete con 600 nodi e 300 link, il tempo di calcolo è di circa 150 ms.

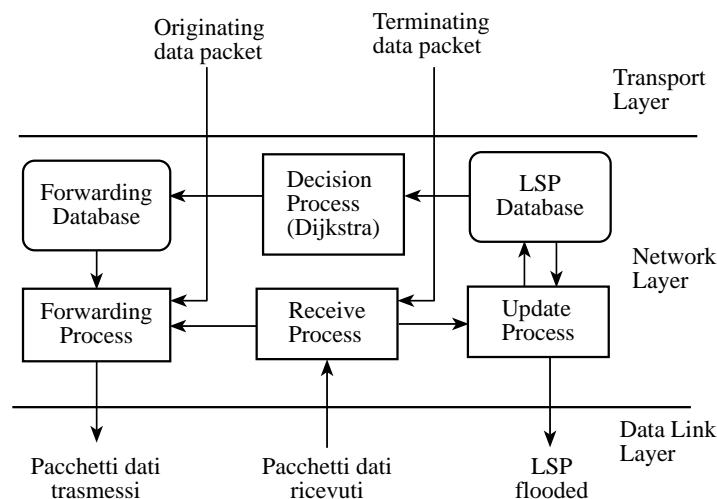
Algoritmi di tipo link state packet sono utilizzati negli standard ISO 10589 (IS-IS) e nel protocollo OSPF (adottato in alcune reti TCP/IP).

L'algoritmo link state packet può gestire reti di grandi dimensioni (10000 nodi), ha convergenza rapida, difficilmente genera loop, e comunque è in grado di identificarli e interromperli facilmente, ed è facile da capire e prevedere poiché ogni nodo contiene l'intera mappa della rete.

Gli svantaggi sono invece nella difficoltà di programmazione e nella necessità di meccanismi speciali per gestire le LAN.

#### 14.7.1 Architettura di un router LSP

Per meglio comprendere il funzionamento dell'algoritmo link state packet esaminiamo l'architettura di un router OSI CLNS che adotta l'algoritmo ISO 10589 (IS-IS), (figura 14.18).



**Fig. 14.18** - Architettura di un router LSP.

Quando il receive process riceve un pacchetto, verifica di quale tipo sia. Possono porsi tre casi:

- il pacchetto è un pacchetto dati in transito verso altre destinazioni; il receive process lo passa al forwarding process, che consulta il forwarding database usando come chiave l'indirizzo di destinazione e determina il nuovo instradamento, cioè su quale linea ritrasmettere il pacchetto;
- il pacchetto è un pacchetto dati destinato al router; ci troviamo in presenza di un pacchetto di gestione (management) che viene passato ai protocolli di livello superiore;
- il pacchetto è un LSP o un pacchetto di neighbor greetings; questo è il caso che necessita della trattazione più approfondita, riportata nel seguito.



Nel caso di un pacchetto di neighbor greetings il router verifica se si tratta di un nuovo nodo adiacente o di un nodo già noto. Nel secondo caso non fa nulla, nel primo caso genera un LSP per informare dell'esistenza del nuovo nodo tutti gli IS, in modo che il nuovo nodo diventi raggiungibile da qualsiasi punto della rete.

Un Link State Packet contiene, oltre alle informazioni di adiacenza già descritte, anche una checksum, un lifetime e un numero di sequenza che serve per distinguere, da parte di un router che riceve più LSP, quelli generati dallo stesso IS.

I LSP vengono trasmessi in flooding su tutti i link del router che li ha originati. Un router che riceve un LSP lo ritrasmette in flooding solo se esso ha modificato il LSP database del router stesso (selective flooding).

All'atto del ricevimento di un LSP un router compie le seguenti azioni:

- se non ha mai ricevuto LSP da quel mittente o se il numero di sequenza del LSP è maggiore di quello del LSP proveniente dalla stessa sorgente e memorizzato nel LSP database, allora memorizza il pacchetto nel LSP database e lo ritrasmette in flooding su tutte le linee eccetto quella da cui l'ha ricevuto;
- se il LSP ricevuto ha lo stesso numero di sequenza di quello posseduto, allora non occorre fare nulla poiché lo stesso pacchetto era già stato precedentemente trasmesso in flooding;
- se il LSP è più vecchio di quello posseduto, cioè è obsoleto, allora il router ricevente trasmette il LSP aggiornato al router mittente.

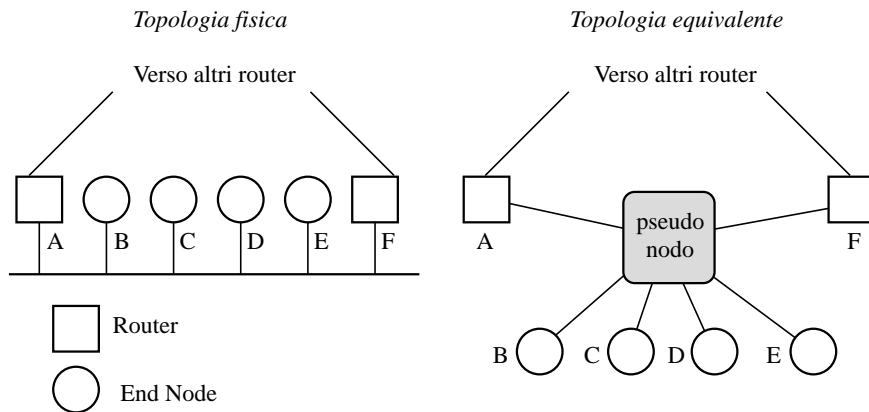
Questo meccanismo serve a fare in modo che i LSP database di tutti i router si mantengano perfettamente allineati e coerenti, condizione indispensabile per un corretto instradamento.

#### 14.7.2 LSP e LAN

Le LAN sono strutture trasmissive broadcast che mal si prestano ad essere modellate come grafi. Infatti su una LAN tutti i nodi sono adiacenti a tutti gli altri e questo porterebbe ad un grafo completamente connesso con un numero di archi quadratico nel numero di nodi. Poiché il numero di nodi su una LAN può anche essere molto elevato tale approccio è improponibile oltre che inutile.

Per questo e per altri motivi si preferisce modellare la LAN come uno *pseudo-nodo*, un nodo fittizio non esistente sulla rete, che viene realizzato da uno dei router presenti sulla LAN (*designated router*): la topologia equivalente diventa dunque una stella con al centro lo pseudo-nodo.

La figura 14.19 mostra un esempio di LAN e il suo modello a stella mediante l'introduzione dello pseudo-nodo.



**Fig. 14.19** - Lo pseudo-nodo.

Il calcolo delle tabelle di instradamento fatto sul modello a stella delle LAN è utile in quanto ogni ES vede la LAN come un collegamento punto-punto con lo pseudo-nodo e quindi non ha necessità di avere informazioni di routing in quanto, indipendentemente dalla destinazione con cui vuole comunicare, è sufficiente che l'ES invii i pacchetti allo pseudo-nodo. Chiaramente un approccio di questo genere risulta particolarmente inefficiente quando due nodi sulla stessa LAN vogliono comunicare, ma, integrato con la problematica del neighbor greetings, può essere la base su cui risolvere molti problemi.

## 14.8 NEIGHBOR GREETINGS

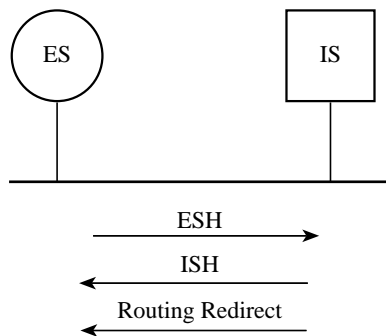
Per la gestione delle interazioni tra ES e IS su rete locale esiste un protocollo apposito che secondo la terminologia OSI si chiama ES-IS (*End System to Intermediate System*).

Tale protocollo ha due scopi:

- Permettere agli ES di conoscere gli IS presenti sulla LAN e viceversa. Questo viene realizzato con la trasmissione periodica in multicast di tipo advertisement di due pacchetti: gli ISH (*Intermediate System Hello*) e gli ESH (*End System Hello*). Gli ISH sono trasmessi dai router e ricevuti da tutti gli ES che memorizzano, in questo modo, l'esistenza di un IS in una cache locale. Gli ESH sono trasmessi dagli ES e ricevuti dagli IS che, in questo modo, scoprono le adiacenze da includersi nei LSP.

- Permettere agli ES di apprendere tramite pacchetti di *routing redirect* se un nodo è direttamente collegato alla LAN oppure qual è il miglior router tramite il quale raggiungerlo.

La figura 14.20 evidenzia i tre tipi di pacchetti e la loro direzione.



**Fig. 14.20** - Neighbor greetings.

Quando un ES deve trasmettere un pacchetto ad un altro ES può ignorare dove si trovi l'ES destinatario ed inviare il pacchetto allo pseudo-nodo, ad un IS sulla sua LAN o al suo router di default (dipende dalle architetture di rete). A ricevere il pacchetto sarà comunque sempre un router che, se verifica che il pacchetto deve essere ritrasmesso sulla stessa LAN da cui è stato ricevuto, genera un pacchetto di routing redirect, oltre a recapitare comunque il pacchetto.

Il pacchetto di routing redirect indica all'ES l'esistenza di un cammino migliore per raggiungere la destinazione. Tale cammino può essere diretto nel caso che l'ES di destinazione si trovi sulla stessa LAN o indiretto se il pacchetto deve transitare per un altro router.

L'ES mittente, all'atto della ricezione di un pacchetto di routing redirect, impara l'informazione in esso contenuta e la scrive nella sua cache locale. I pacchetti successivi per lo stesso destinatario verranno inviati direttamente nel modo indicato dal contenuto del pacchetto di routing redirect.

Questo meccanismo è flessibile perché consente a tutti gli ES di comunicare nel modo ottimale con tutti gli altri nodi della rete (ad eccezione del primo pacchetto) ed inoltre limita la dimensione della cache sugli ES: infatti in essa sono contenuti solo gli indirizzi degli IS collegati alla LAN e la raggiungibilità degli ES con cui sono in corso scambi di informazioni.

## 14.9 ROUTING GERARCHICO

Anche se si adottano algoritmi di tipo LSP non è certo pensabile che essi riescano a trattare qualsiasi rete di qualsiasi dimensione: si pensi a Internet con le sue decine di milioni di calcolatori collegati e un tasso di crescita del 5% al mese.

Quindi occorre organizzare il routing in modo gerarchico, cioè partizionare la rete in aree (a volte si usano anche i termini di dominio, net o subnet). All'interno dell'area (routing intra-area) il routing segue esattamente le regole sin qui descritte. Quando invece bisogna far comunicare due nodi appartenenti ad aree diverse (routing inter-area) si divide il problema in tre sottoproblemi:

- un problema di instradamento tra il nodo mittente e la periferia dell'area cui il nodo mittente appartiene;
- un problema di instradamento tra l'area mittente e l'area destinazione;
- un problema di instradamento all'interno dell'area destinazione.

Tutte le principali architetture di rete attuali hanno il concetto di routing gerarchico. Ad esempio, SNA ha il concetto di subarea, OSI ha i concetti di dominio e area, DECnet quello di area, TCP/IP quello di network e subnetwork.

La figura 14.21 illustra una rete ripartita in tre aree. Supponiamo di dover instradare il messaggio dal nodo G al nodo A. In una prima fase si invia il messaggio ad F (router di area per l'area 15). F deve instradare il messaggio all'area 10 e ha due possibilità: il cammino diretto con costo 3 o quello indiretto con costo 4 (tramite l'area 12). Sceglie ovviamente il cammino diretto e passa il messaggio ad E, che lo passa a D, che lo passa a B che lo recapita ad A, nodo destinatario.

Si noti che l'instradamento ottenuto è ottimale, ma non ottimo: infatti il suo costo è pari a 16, mentre se si fosse passati da C il costo sarebbe stato pari a 10.

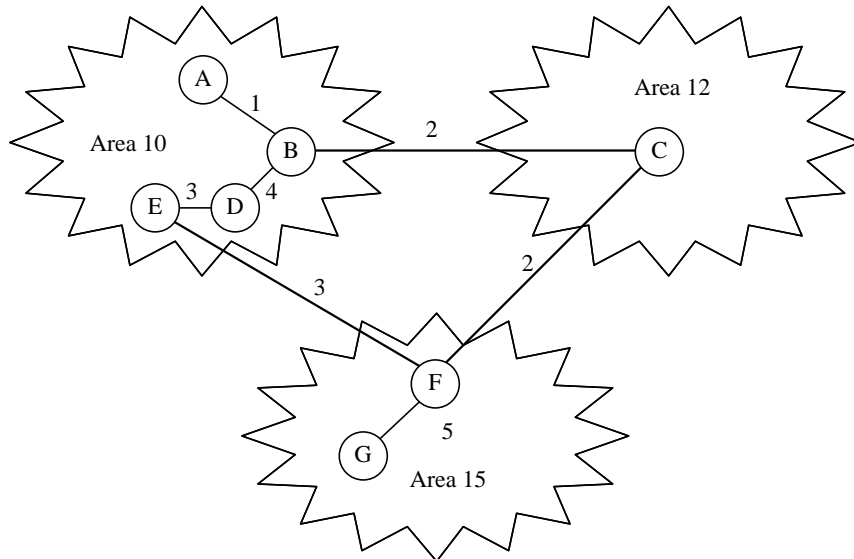
Questo è uno degli svantaggi del routing gerarchico in quanto, non avendo una visione globale della rete, si compiono tante scelte ottime di per sé, ma che considerate nell'insieme possono non rappresentare l'ottimo globale.

Il grande vantaggio del routing gerarchico è che ogni area ha dimensioni ragionevoli e può essere gestita da algoritmi di routing distribuito.

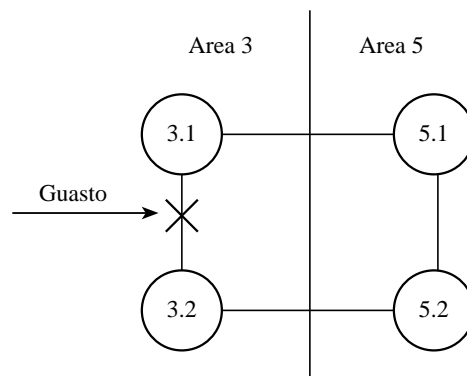
Si noti infine che nulla vieta di avere più livelli di gerarchia, cosa che avviene in OSI e TCP/IP.

Il routing gerarchico necessita di alcune cautele nella configurazione dei cammini inter-area. Per capire il problema si consideri la semplice rete riportata in figura 14.22.

In presenza del guasto indicato, il nodo 3.1 non riesce a scambiare messaggi con il nodo 5.2. Questo può essere compreso se si applica il criterio di instradamento inter-area prima discusso.



**Fig. 14.21** - Routing gerarchico.



**Fig. 14.22** - Area partizionata.

Infatti 3.1 cerca per prima cosa di far giungere il messaggio nell'area destinazione e quindi lo passa a 5.1 che, effettuando un instradamento intra-area, lo consegna al nodo 5.2. Il nodo 5.2, ai livelli superiori (tipicamente a livello trasporto), genera un acknowledge che viene instradato verso il nodo 3.1 a livello 3. Il nodo 5.2 cerca per prima cosa di inviare il messaggio all'area di destinazione e lo passa al nodo 3.2. Il

nodo 3.2 consulta le sue tabelle di instradamento, constata che non esiste un cammino intra-area con il nodo 3.1 e scarta il messaggio in quanto non recapitabile. Questo avviene perché l'area 3 è partizionata: è come se sulla rete esistessero due diverse aree 3, una raggiungibile tramite il nodo 5.1 e l'altra tramite il nodo 5.2.

Questo esempio ci permette di fare un'osservazione importante: quando si usano protocolli di livello 3 connectionless con routing distribuito non si è certi che il messaggio da A a B faccia lo stesso percorso del messaggio da B ad A. Occorre quindi configurare le aree in modo fortemente connesso, in modo che la caduta di un solo link non possa portare al loro partizionamento. È inoltre opportuno minimizzare i punti di contatto tra le aree cercando di renderli il più possibile affidabili.

Sempre con riferimento alla figura 14.22, si noti che se a guastarsi fosse stato il link tra 3.2 e 5.2 invece di quello tra 3.1 e 3.2, nessuna area si sarebbe partizionata e la rete avrebbe continuato a funzionare correttamente.

## BIBLIOGRAFIA

- [1] J. V. Aho, J. E. Hopcroft, J. D. Ullman, "Data Structures and Algorithms", Addison-Wesley, Reading MA (USA), 1983.
- [2] Cisco Systems, "Internetworking Technology Overview", Codice documento DOC-ITO13 78-1070-01, 1993.
- [3] A. Tanenbaum, "Computer Networks," Prentice-Hall.
- [4] J. Martin, J. Leben, "DECnet Phase V: An OSI Implementation", Digital Press, Bedford MA (USA), 1992.
- [5] ISO 8802-2 (ANSI/IEEE Std 802.2), "Logical Link Control".
- [6] Cisco Systems, "Router Products Configuration and Reference", Cisco Systems DOC-R9.1, Menlo Park CA (USA), September 1992.
- [7] ISO, "TR 9577: Information technology - Telecommunications and information exchange between systems - Protocol identification in the OSI network layer", 1990.
- [8] ISO, "DTR 9577: Information technology - Telecommunications and information exchange between systems - Protocol identification in the OSI network layer", 1993.
- [9] ISO 8473, "Protocol for Providing the Connectionless-mode Network Service".
- [10] ISP 9542, "End system to Intermediate system routing exchange protocol for

use in conjunction with the Protocol for providing the connectionless-mode network service".

- [11] ISP 10589, "Intermediate system to Intermediate system Intra-Domain routing information exchange protocol for use in conjunction with the Protocol for providing the connectionless-mode network service".
- [12] J. Postel, "RFC 791, Internet Protocol", 09/01/1981.
- [13] J. Postel, "RFC 792: Internet Control Message Protocol", 09/01/1981.
- [14] C. Hedrick, "RFC 1058, RIP: Routing Information Protocol", 06/01/1988.
- [15] G. Malkin, "RFC 1388: RIP Version 2 Carrying Additional Information", 01/06/1993.
- [16] J. Moy, "RFC 1583: OSPF Version 2", 03/23/1994. (Pages=212).
- [17] K. Lougheed, Y. Rekhter, "RFC 1267: A Border Gateway Protocol 3 (BGP-3)", 10/25/1991.
- [18] M. L. Peters, "APPN and Extensions: The New Industry Standard for SNA Internetworking", IBM Corp, Research Triangle Park, NC, USA.
- [19] J. P. Graym Marcia L. Peters, "A Preview of APPN High Performance Routing", IBM Corp, Research Triangle Park, NC, USA, July 1993.