

LSP Link State Packet:

Esso è basato sul principio che ogni router descrive la topologia della rete intorno a sé, cioè lo stato dei suoi collegamenti (link state); quindi diffonde questa situazione in tutta la rete inserendo tali informazioni all'interno di pacchetti detti LSP (Link State Packet). Per ogni collegamento è specificata l'identità del nodo all'altro capo e la metrica associata.

In questo modo ogni apparato costruisce la propria tabella come la tessera di un mosaico per produrre un mappa dell'intera rete e sceglie attraverso la mappa il percorso migliore per arrivare ad ogni destinazione, costruendo il più basso albero (**shortest path tree**) che ha se stesso come radice ed ognuna delle destinazioni come foglie.

La distanza dalla radice ad un altro nodo è data dalla composizione dei costi associati ai singoli collegamenti (rami) che portano al nodo. L'albero è costruito utilizzando l'algoritmo di **Dijkstra**, detto anche algoritmo **SPF (Shortest Path First)** per il modo in cui opera.

A partire dallo **shortest path tree**, un Intermediate-System costruisce la tabella di routing associando ad ogni destinazione l'interfaccia dell'apparato da cui parte il sottoalbero che contiene la destinazione ed un costo di raggiungibilità pari alla distanza dalla radice.

**Dijkstra** ha dimostrato che il suo algoritmo garantisce che gli **shortest path tree** calcolati dai vari apparati sono tra loro coerenti, cioè non portano alla creazione di percorsi di inoltro ciclici.

Questo algoritmo è caratterizzato da una elevata stabilità, cioè brevi tempi di convergenza e quasi zero inclinazione a creare routing loop.

I brevi tempi di convergenza derivano da tre fattori importanti:

La complessità dell'algoritmo di **Dijkstra**, cioè il numero di passi di calcolo che portano alla costruzione del più basso albero di rete (shortest path tree) è dell'ordine di  $C \log N$ , dove  $C$  è il numero di collegamenti della rete ed  $N$  il numero di nodi; in altre parole il tempo di calcolo dipende linearmente dalla complessità topologica della rete e logaritmicamente dalla dimensione della rete stessa. Questo garantisce all'algoritmo di routing Link State una maggiore scalabilità rispetto all'algoritmo distance vector, il cui tempo dipende dal quadrato o dal cubo del numero di nodi.

Gli apparati Intermediate-System calcolano lo **shortest path tree** (tabella di routing) in modo indipendente dagli altri, cioè ognuno alla propria velocità non pregiudicando mai le prestazioni complessive della rete come avviene invece nell'algoritmo distance vector.

A seguito di un cambiamento topologico, gli Intermediate-System che si trovano nelle vicinanze aggiornano immediatamente la propria tabella di routing.

Gli apparati lontani devono attendere che gli LSP generati da questi IS che hanno rilevato il cambiamento topologico si propagano fino a loro per innescare il ricalcolo della tabella di routing.

Comunque, in genere più gli apparati sono lontani dalla zona in cui c'è stato il cambiamento, tanto più è probabile che esso non porti ad una modifica della loro tabella di route.

Esse contengono l'identificatore e la distanza (espressa in metrica) di tutti i nodi direttamente collegati ad essi, siano essi Intermediate-System che End-System.

Tramite l'inoltro basato su selective flooding (inondazione selettiva), l'LSP generato da "A" (da D e da tutti gli altri) viene diffuso su tutta la rete e memorizzato da tutti gli Intermediate-System e End-System nel loro LSP database.

La tecnica di inoltro del **flooding** prevede che un router che riceve un pacchetto da un'interfaccia lo inoltra su tutte le altre sue interfacce ed è in grado di eliminare loop interni alla rete in caso di presenza di collegamenti magliati poiché inoltra il pacchetto solo una prima volta; le eventuali copie successivamente ricevute dello stesso pacchetto verranno automaticamente scartate.

Quando un router riceve un LSP (in questo caso particolare inviato da A e D) controlla se lo stesso sia già presente nella propria base di dati:

- se non esiste e ne esiste una più vecchia, memorizza l'attuale LSP e lo inoltra (immutato) su tutte le sue interfacce, tranne su quella da cui ha ricevuto l'LSP.
- se nella base dati è presente una copia identica dell'LSP, ignorano quello appena ricevuto.
- se la copia memorizzata è più recente (numero di sequenza più alto) di quella appena ricevuta, la restituiscono al mittente la cui base di dati è chiaramente non aggiornata