

Algoritmo Globale, Basato su Dijkstra (nessun arco ha peso negativo), L'albero è l'albero dei cammini minimi. Tiene conto dello stato del link e la condizione di partenza dell'algoritmo è a tempo periodicamente i nodi aggiornano i vicini sullo stato del link, situazione di terminazione stabile nel grafo (quando tutti i nodi hanno calcolato il proprio cammino minimo verso ogni altro nodo)

## Link State Routing (LSR)

Deve per superare alle problematiche di DV

Link State Routing è un algoritmo che nasce per sopperire ad alcune carenze di Distance Vector, come il fatto che non tiene conto dello **stato dei link** ma attribuisce banalmente un peso unitario a tutti i collegamenti.

**Questo è un algoritmo globale** (tutti i nodi sincronizzati sulle stesse informazioni) **basato su Dijkstra, ipotesi applicabile in pratica dato che nessun arco ha peso negativo**. L'output dell'algoritmo è l'albero dei cammini minimi, ovvero un **minimum spanning tree** che può non essere unico se esistono diversi cammini che portano alla stessa destinazione con uguale peso.

Ogni nodo conosce i suoi vicini e che nessun arco ha peso neg. Si sfrutta il Flooding: illustrando la tabella dei nodi adiacenti e il peso del percorso per inviare al nodo che l'ha ricevuto l'indirizzo per tornare al floodino e utilizzarlo, l'indirizzo solo se l'introduzione di altri nodi non fa cambiare la tabella degli altri nodi (Ogni router implementa Dijkstra)

L'algoritmo tiene conto dello stato del link e la condizione di partenza dell'algoritmo è a tempo, infatti periodicamente ogni nodo informa quelli ad esso adiacenti delle condizioni dei link. Le condizioni di terminazione riguardano il raggiungimento di una situazione stabile nel grafo, ovvero quando tutti i nodi hanno calcolato il proprio cammino minimo verso ogni altro nodo nel grafo, ma come si può conoscere la topologia del grafo?

Ogni nodo sa solo chi sono i propri vicini e che nessun arco ha peso negativo o nullo. Ogni nodo sfrutta il **Flooding** per inviare in broadcast a tutti i vicini la propria tabella di nodi adiacenti, contenente il peso dell'arco che permette di raggiungerli. L'approccio possibile per fermare il Flooding è tenere conto dell'ID ma la soluzione è applicabile solo se il messaggio da inviare è sporadico, ovvero se l'introduzione di altri nodi non fa cambiare le tabelle degli adiacenti molto frequentemente. La conseguenza di questo comportamento è che ogni router applica Dijkstra.

LSR è robusto agli attacchi, in quanto ogni nodo non può fornire informazioni fittizie dei link ai nodi adiacenti perché verrebbero immediatamente contrariate dall'altro nodo, in quanto si riferiscono allo stato di un link che è uguale in entrambe le direzioni.

### Problemi con LSR

Provata dall'alternarsi continuamente diversi percorsi minimi di uguale peso dati da diversi percorsi minimi crea problemi di sincronia e LSR sfrutta la sincronizzazione delle informazioni da parte dei nodi.

- Un problema di **oscillazione** riguarda il fatto che si alternano continuamente diversi percorsi minimi di uguale peso dati da diversi alberi dei cammini minimi. L'oscillazione crea problemi di sincronia in quanto LSR sfrutta la sincronizzazione tra le informazioni nei nodi per la propria esecuzione.
- Un altro problema riguarda la **dimensione del grafo**, infatti il flooding causa una forte congestione all'aumentare del numero di nodi. Inoltre considerare porzioni troppo grandi del grafo comporta ricoprire AS differenti, il che non interessa i link considerati.

(Risolti basando l'ID del messaggio, come Flooding non asincrono problemi)

## RIP (routing information protocol) —[Distance Vector]—

Il Routing Information Protocol (RIP) è un protocollo di routing che **fornisce l'implementazione di distance vector**:

**RIPv1** Utilizza come metrica di distanza il **numero di hop** necessari per raggiungere la destinazione, in ogni router vi è la **tabella di routing** che indica il numero di hop necessari per raggiungere ogni nodo della rete, ogni router sceglie il percorso migliore, in più il router in RIP trasmette periodicamente le proprie informazioni di routing agli altri router.

**RIPv1** utilizza una metrica di distanza basata sul **numero di hop** (o salti) necessari per raggiungere una destinazione, infatti ogni router mantiene una **tabella di routing** che indica il numero di hop necessari per raggiungere ogni nodo nella rete. Si fissa il numero massimo di salti a 15 per prevenire ritardi nella convergenza dell'algoritmo dovuti ad eccessive distanze tra nodi. Un router sceglierà il percorso più breve in termini di numero di hop per raggiungere la destinazione.

Con RIP ogni router in una rete trasmette periodicamente (ogni 30 secondi) le proprie informazioni di routing agli altri router tramite **messaggi di aggiornamento**.

RIP utilizza un **timer di validità** per determinare se un percorso di routing non è più valido, in particolare se un percorso non viene aggiornato entro il tempo di invalidità, viene considerato non valido e rimosso dalla tabella di routing.

Infine, il protocollo RIP sfrutta un **garbage-collection timer** (120 secondi) per rimuovere i router che non sono più raggiungibili, dalla tabella di routing. Se un router non trasmette informazioni di routing per il tempo scandito da questo timer, allora viene considerato irraggiungibile e rimosso dalla tabella di routing.

Nel pacchetto RIPv1 non è specificata una maschera in quanto opera utilizzando le classi di indirizzi.

Se il percorso non viene aggiornato entro un tempo

Se non trasmette informazioni per il tempo scandito dal timer

Vi è il **timer di invalidità** per determinare se un percorso di routing è valido (10), RIP sfrutta **garbage-collection timer** (120 secondi) per eliminare i router **non raggiungibili** dalla **tabella di routing**.

Una **tabella di routing** contiene i seguenti campi:

- Indirizzo di destinazione**
- Distanza dalla destinazione in hop** (massimo 15)
- Next hop**: router adiacente a cui inviare i pacchetti
- Timeout** (ogni 30 secondi)
- Garbage-collection timer** (120 secondi)

**RIP v2** Sono stati aggiunti diverse caratteristiche, **sopprimere il CIDR**, **supporto l'autenticazione dei messaggi** (prevenire di autenticare chi ha inviato il messaggio, in modo da evitare inserimento di informazioni fittizie nella **tabella di routing**, permette pure di specificare il prossimo hop per raggiungere la destinazione), **permette di gestire sottoreti più grandi**, **migliore efficienza sull'utilizzo degli IP**.  
Suppl. **"Split Horizon"** e **Poison Reverse** → inserisce un valore infinito se il router non è raggiungibile.

**RIPv2** aggiunge diverse caratteristiche rispetto a **RIPv1**:  
per prevenire i loop tra i router

**RIPv2** supporta l'**indirizzamento CIDR** e il subnetting con maschere di sottorete variabili, ciò significa che **RIPv2** può gestire sottoreti di dimensioni diverse, migliorando l'efficienza nell'utilizzo degli indirizzi IP.

Questa versione supporta l'**autenticazione dei messaggi**, che consente ai router di autenticare l'origine dei messaggi di aggiornamento della tabella di routing e di evitare l'inserimento di informazioni di routing fittizie.  
Consente ai router di specificare il prossimo hop per raggiungere una determinata destinazione. Ciò significa che i router possono scegliere il prossimo hop per un percorso di routing in base a fattori come il costo del percorso.

**RIPv2** utilizza il **"split horizon"** per prevenire i loop di routing tra i router. Questo significa che un router non annuncerà una destinazione a un altro router se ha appreso quella destinazione da quel router stesso. Inoltre, **RIPv2** supporta anche la tecnica del **"poisoned reverse"**, che prevede che un router annuncerà una destinazione con una metrica di distanza infinita (di solito 16) a un altro router dal quale ha appreso che quella destinazione non è raggiungibile. Questo aiuta a prevenire loop di routing rimuovendo immediatamente la destinazione dalla tabella di routing del router che ha appreso l'informazione.

**RIP ng** (new generation)

Questa è una versione aggiornata di RIP che usa indirizzi IPv6

Protocollo di routing a stato di link che determina il percorso migliore tra due punti della rete.  
Basandosi sulla topologia di rete e sulla metrica di costo (tiene conto di diversi fattori).

**OSPF (Open shortest path first) — [Link State Routing]** —

↳ Protocollo robusto e affidabile. Fornisce buona flessibilità, scalabilità.

L'implementazione di LSR è OSPF: è un protocollo di routing a stato di link che determina il percorso migliore tra due punti di una rete, basandosi sulla topologia di rete e sulla metrica di costo, che tiene conto di fattori come la larghezza di banda, il ritardo e il traffico.

L'implementazione utilizza i messaggi sono scambiati tra i router che partecipano al protocollo OSPF, al fine di costruire un database di stato dei collegamenti che rappresenta la topologia della rete. Una volta che il database di stato dei collegamenti è stato costruito, **OSPF** utilizza l'algoritmo Dijkstra per determinare il percorso più breve tra due punti della rete. Questo percorso viene quindi utilizzato per instradare il traffico da una sorgente a una destinazione.

**OSPF** è un protocollo di **routing dinamico**, il che significa che è in grado di adattarsi automaticamente ai cambiamenti nella topologia della rete, come l'aggiunta o la rimozione di router o sottoreti. In **OSPF** lo stato del link viene valutato tramite l'invio di pacchetti di tipo "hello" tra i router che partecipano al protocollo.

**OSPF** è ampiamente utilizzato nelle reti di grandi dimensioni, come quelle delle aziende o dei provider di servizi Internet (ISP). In generale, **OSPF** è considerato un protocollo di routing robusto e affidabile, che fornisce una buona scalabilità, flessibilità e sicurezza.

L'implementazione: vengono utilizzati i messaggi che sono scambiati tra i router che partecipano al protocollo OSPF al fine di costruire un database di stato dei collegamenti che rappresenta la topologia di rete (algoritmo Dijkstra per determinare il percorso migliore).  
**OSPF** è un protocollo di routing dinamico si adatta bene ai cambiamenti della topologia di rete.  
Lo stato del link viene valutato tramite l'invio del pacchetto di tipo "hello" tra i router che partecipano al protocollo.

## Internet Routing

Internet è organizzata in autonomous systems (AS).

Un Autonomous System (AS) è un insieme di reti interconnesse e gestite da un'unica entità amministrativa che utilizza un protocollo di routing comune. In altre parole, un AS è un sistema autonomo e indipendente che si occupa di instradare il traffico Internet all'interno della propria rete e verso altre reti.

Distinguiamo due tipologie di protocolli di routing :

- Consente ai Router all'interno dello stesso AS di scambiare informazioni sulle rotte disponibili di selezionare il percorso migliore per instradare i pacchetti all'interno dell'AS.  
**Interior gateways protocol (IGP)** : è un tipo di protocollo di routing che consente ai router all'interno dello stesso AS di scambiare informazioni sulle rotte disponibili e di selezionare il percorso migliore per instradare i pacchetti all'interno dell'AS. Alcuni esempi di protocolli IGP includono **RIP** e **OSPF**.
- Protocolli di routing di Gateway esterni sono utilizzati per il routing tra diverse AS (consentono ai Router di scambiare informazioni sulle rotte disponibili tra AS diversi)  
**Exterior gateways protocol (EGP)**: i protocolli di routing di gateway esterni sono utilizzati per il routing tra diverse AS e consentono ai router di scambiare informazioni sulle rotte disponibili tra AS differenti e di condividere informazioni sulla raggiungibilità delle destinazioni. **BGP** (Border Gateway Protocol) è un esempio di protocollo EGP.