

# Complementi di Reti di Telecomunicazione

Samuele Valperta

September 2024

## 1 Distributed Bellman-Ford

Il percorso tra  $X$  e  $Z$  è calcolato come segue, dove  $K$  è un nodo adiacente ad  $X$ :

$$D(X, Z) = \min_k \{d(X, K) + D(K, Z)\}, \forall K$$

L'algoritmo è scalabile (è  $O(n)$ ).

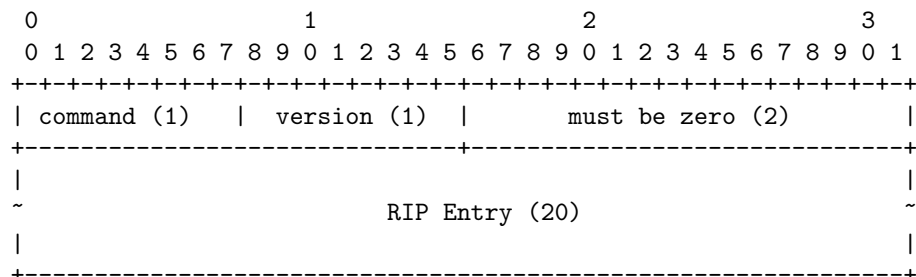
Alcuni cambiamenti possono portare a **counting to infinity** (convergenza lenta, o, al limite, non convergenza) ed, in alcuni casi, l'algoritmo causa **routing loop**.

## 2 RIP

Implemente alcune ottimizzazioni al classico DBF, tra cui:

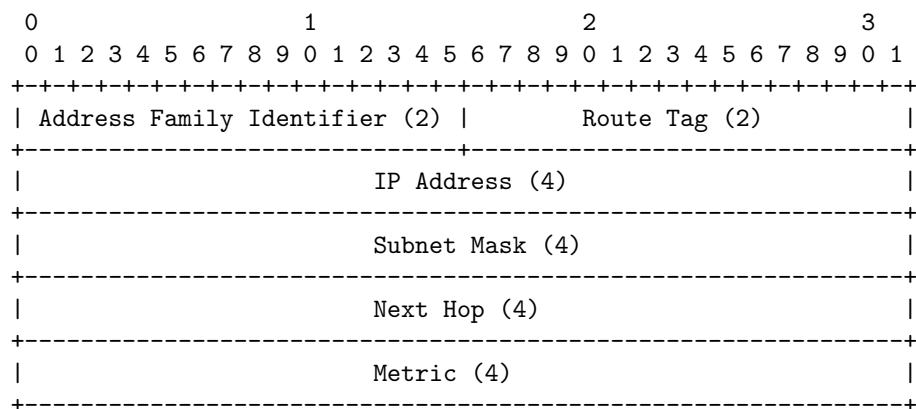
1. **Limite alla massima distanza**, quando una route raggiunge il limite (16), la destinazione viene considerata irraggiungibile;
2. **Split horizon**, se un nodo riceve un aggiornamento da una certa interfaccia, non deve propagare la stessa informazione sulla stessa interfaccia.  
A K mando solo i distance vector delle destinazioni per cui K non è il next-hop;
3. **Poison reverse**, il nodo che riceve un'informazione da una determinata interfaccia propaga la stessa informazione sulla medesima interfaccia ma con metrica  $\infty$ .  
A K mando anche i distance vector delle destinazioni per cui K è il next-hop ma con metrica  $\infty$ ;
4. **Triggered update**, quando una route su un nodo RIP cambia (viene aggiunta, o comunque il flag "changed" viene impostato a 1), il nodo, invece di attendere il normale intervallo di tempo per gli update, manda un triggered update quasi immediatamente.

## 2.1 RIP: formato dei messaggi



- Command
  - 1 : request
  - 2 : response
- RIP Entry: da 1 a 25 Entry per ogni messaggio RIP.

## 2.2 RIP Entry (v2)



- Route Tag: indica le route che sono state apprese attraverso un EGP, rispetto a quelle apprese attraverso RIP

## 3 Link-State

È un algoritmo centralizzato: ogni nodo conosce la topologia della rete, e calcola il proprio albero con Dijkstra in maniera indipendente da tutti gli altri ma richiede la cooperazione di tutti i nodi perché vi sia lo scambio di informazioni sufficiente a far sì che ciascun nodo ricostruisca la topologia della rete.

Il funzionamento avviene in più fasi:

1. **Conoscenza dei nodi adiacenti:** ciascun nodo invia un pacchetto su ogni linea in uscita, e si aspetta un pacchetto di risposta da ciascun nodo adiacente che lo identifichi e ne permetta il calcolo del costo;
2. **Distribuzione dei pacchetti link state:** ciascun nodo replica i pacchetti ricevuti verso i propri vicini causando **flooding**. Il flooding viene controllato tramite meccanismi che utilizzano campi speciali dei pacchetti. Dato che ognuno dei due nodi che stanno a capo di ciascun link manda un pacchetto distinto, questo permette di caratterizzare anche i grafi orientati.
3. **Calcolo delle tabelle di routing:** ciascun nodo conosce la topologia completa della rete e usa Dijkstra per calcolare l'albero dei cammini minimi che lo riguarda, e configura di conseguenza la propria tabella di routing.

La quantità di memoria richiesta su ogni nodo, nel caso peggiore, è proporzionale a  $n^2$ . La complessità computazionale, su ciascun nodo, deriva dalla complessità di Dijkstra:  $O(n^2)$ .

## 4 Open Shortest Path First (OSPF)

È un protocollo di tipo link state in grado di gestire diverse metriche, sfruttare il load balancing e con un meccanismo di autenticazione integrato. Sfrutta una gestione gerarchica: viene introdotto il concetto di "area", ovvero un raggruppamento di più sottoreti, la cui topologia viene nascosta al resto dell'AS. Questo lo rende scalabile e particolarmente adatto a reti di grandi dimensioni.

### 4.1 Dalla rete al grafo che la rappresenta

- Ad ogni interfaccia è associato un arco pesato ed orientato, ad ogni rete o router un vertice.
- Esistono due tipologie di sottorete:
  1. **Transit network**, il traffico passa attraverso la rete per raggiungere altri vertici. Ogni transit network viene collegata con una coppia di archi per ogni router collegato;
  2. **Stub network**, non è di transito, c'è solo un router collegato alla rete. Ogni stub network è collegata con un solo arco entrante, non ci interessa costruire lo shortest-path a partire da una stub network.
- I canali di collegamento verso le reti esterne all'AS sono speciali e il conteggio del loro costo può avvenire in due modalità differenti:

1. il costo del canale verso l'esterno viene sommato al costo del cammino ottimo interno dell'AS e il cammino con risultante inferiore viene scelto come percorso.
2. Il percorso ottimo (interno dell'AS) verso il nodo che ha il canale a costo minimo verso l'esterno diventa il percorso prescelto.

## 4.2 Backbone

È importante introdurre il concetto di **backbone** (Area ID = 0) ovvero l'insieme di tutte le sottoreti che non fanno parte di alcuna area, dei router che sono loro connessi, e dei router che fanno parte di più di un'area.

Il backbone può essere visto come un hub che connette tutte le altre aree, nascondendone i dettagli topologici, per svolgere questa funzione, le sottoreti e i router che compongono il backbone devono ricoprire uno spazio contiguo. Se per motivi particolari ciò non dovesse accadere, si possono configurare **virtual link** (virtual perchè passano attraverso sottoreti non appartenenti al backbone) per connettere le parti non contigue.

## 4.3 Classificazione dei router

La classificazione dei router è:

- **Router interni**, sono connessi SOLO a sottoreti appartenenti ad una sola area;
- **Router di bordo area**, sono i router connessi a più aree. In questi router vengono utilizzate più istanze dell'algoritmo di base, ogni istanza che si occupa di una certa area ha il compito di presentare un "riassunto" della topologia di tale area alle altre aree;
- **Router di backbone**, sono connessi almeno ad una sottorete del backbone. Possono essere a loro volta interni o di bordo area;
- **Rouer di bordo AS**, scambiano informazioni con altri AS.

Vi è poi il **designated router** relativo ad una rete multi-access il quale ha il compito di ricevere e distribuire le informazioni relative alla rete di appartenenza.

## 4.4 Messaggi Link State Update (LSU)

Contengono il link state dei vari router, descritti attraverso una serie di **link state advertisement (LSA)**.

Esistono diversi tipi di LSA:

- **Router LSA**, include la metrica di tutti i link direttamente connessi ad un dato router il cui flooding si ferma all'interno dell'area di appartenenza;

- **Network LSA**, generato per definire i link collegati ad un nodo che rappresenta una rete multi-access.  
Esempio Solo il designed router (eg. A) genera  $[NET1 \rightarrow \{A,B,C\}, 0, N]$  e  $[A \rightarrow NET1, 2, NS]$ . Ogni router X collegato alla rete multi-access manda un LSA del tipo  $[X \rightarrow NET1, 2, R]$ ;
- **Network Summary LSA**, generato solo dai router di bordo area e compattano le destinazioni al di fuori dell'area alla quale si manda il messaggio;
- **AS-External LSA**, generato dai router di bordo-AS per riassumere le destinazioni apprese con un EGP (esterne all'AS). Anche i router BA replicano questi pacchetti nelle loro aree.  
Esempio  $[A \rightarrow 0/0, 2, AS-EXT]$ ;
- **AS-BR Summary LSA**, generato dai router di bordo area, riporta dentro l'area la distanza ai vari AS-BR.  
Esempio Sia B un router backbone collegato con un altro AS e D un router BA, quindi con una interfaccia verso il backbone, allora D riporta sull'altra interfaccia  $[D-B, 4, ASBR-SUM]$ .