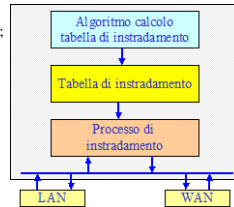


## Algoritmi di routing

- La funzione principale un router è quella di instradare i pacchetti nella rete,
- Ogni router possiede una tabella di instradamento in cui è indicata, per ogni possibile destinazione, la linea di uscita da utilizzare.
- Per costruire la tabella di instradamento vengono utilizzati gli algoritmi di routing, dei quali ne esistono diversi tipi, ciascuno con un impatto diverso sulla rete e sui router stessi.
- I requisiti comuni che si desidera gli algoritmi di routing abbiano, si possono riassumere nei seguenti criteri:
  1. ottimalità;
  2. semplicità e basso costo computazionale;
  3. robustezza e stabilità;
  4. rapida convergenza;
  5. flessibilità.



## Algoritmi di routing

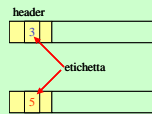
- Le tecniche di routing possono essere suddivise in due classi

Routing basato sull'indirizzo di rete:

- l'intestazione del pacchetto contiene l'indirizzo del destinatario, che rappresenta un indirizzo universale (ad esempio l'indirizzo IP).
- Quando un router riceve un pacchetto utilizza una tabella di routing per decidere su quale interfaccia deve trasmetterlo per inviarlo a destinazione.

Routing basato su label swapping o scambio di etichetta

- L'intestazione di ogni pacchetto contiene una etichetta, che attraverso la tabella di routing consente di definire l'interfaccia su cui deve essere trasmesso in uscita e l'indicazione di una nuova etichetta in uscita.



## Routing

- L'instradamento avviene mediante la consultazione di una tabella (tabella di routing). Generalmente, tali informazioni indicano qual'è il "prossimo salto" (next hop) che un datagram deve effettuare nel suo trasferimento verso la destinazione, vale a dire qual è il prossimo router a dover elaborare il datagram in transito.

Formato di una riga della tabella di instradamento

[Net-id] [prossimo router] [num. hop] [interfaccia]

dove:

- [Net-id] indica l'indirizzo di rete di un possibile destinatario;
- [prossimo router] indica l'indirizzo IP del router che deve ricevere i pacchetti destinati a [Net-id];
- [num. hop] indica il numero di hop (consegne dirette da effettuare) per raggiungere [Net-id];
- [interfaccia] è l'identificativo dell'interfaccia di uscita del router che permette di raggiungere [prossimo router].
- Si osservi che, se l'interfaccia di uscita consente di raggiungere il prossimo router con un collegamento punto-punto, l'indirizzo di tale router è superfluo e spesso non viene indicato.

## Tabella di routing

- Il funzionamento di un algoritmo di routing dipende in modo diretto dalla tabella di routing, che contiene le indicazioni necessarie a instradare i diversi pacchetti.
- I contenuti della tabella di routing dipendono strettamente dal protocollo utilizzato.

Esempio di tabella di routing basata sull'indirizzo di rete

Indirizzo di destinazione	Interfaccia di uscita	Costo
XXXX	0	120
YYYY	1	210
ZZZZ	2	80
VVVV	1	20

Esempio di tabella di routing basata su etichette (label)

Etichetta ingresso	Interfaccia Uscita	Nuova etichetta
1	0	5
2	3	1
3	1	2

Destinazione	Next hop	Interfaccia
28.1.1.0/24	---	1
3.1.2.0/24	---	2
9.12.3.0/24	---	3
132.1.1.0/24	---	4
118.10.0.0/16	132.1.1.2	4
111.1.1.0/24	9.12.3.12	3
115.0.0.0/8	9.12.3.12	3
1.2.3.0/10	3.1.2.1	2
0.0.0.0/0	3.1.2.1	2

## Tabella di routing

- Interfacce del router: non viene indicato il next hop

Destinazione	Next hop	Interfaccia
28.1.1.0/24	---	1
3.1.2.0/24	---	2
9.12.3.0/24	---	3
132.1.1.0/24	---	4
118.10.0.0/16	132.1.1.2	4
111.1.1.0/24	9.12.3.12	3
115.0.0.0/8	9.12.3.12	3
1.2.3.0/10	3.1.2.1	2
0.0.0.0/0	3.1.2.1	2

- Next hop per alcune reti

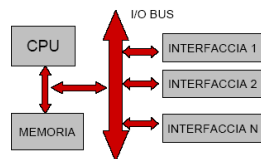
Destinazione	Next hop	Interfaccia
28.1.1.0/24	---	1
3.1.2.0/24	---	2
9.12.3.0/24	---	3
132.1.1.0/24	---	4
118.10.0.0/16	132.1.1.2	4
111.1.1.0/24	9.12.3.12	3
115.0.0.0/8	9.12.3.12	3
1.2.3.0/10	3.1.2.1	2
0.0.0.0/0	3.1.2.1	2

- Su alcuni router è possibile indicare un cammino di default per tutti gli indirizzi non presenti nella tabella.

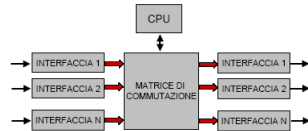
## Architettura di un router di prima generazione

Workstation con schede di rete, bus e CPU centrale

- i pacchetti ricevuti dalle interfacce vengono trasferiti nella memoria (DMA) tramite il bus
- la CPU elabora le istanze e consulta la tabella di routing, inoltre esegue i protocolli di routing
- i pacchetti elaborati vengono trasferiti sulle interfacce ancora tramite il bus
- Sono limitate dal fatto che tutti i pacchetti devono transitare per un solo punto di cortesia rappresentato tipicamente dal bus di I/O, che viene attraversato due volte per ciascun trasferimento
- Esempio: un processore con un bus di I/O a 1Gbit/s può supportare fino a 5 link a 100 Mbit/s o 3 link a 155 Mbit/s, ma neanche un link a 622 Mbit/s
- Inoltre, in presenza di tabelle di routing di grandi dimensioni, il tempo di accesso alla memoria può risultare un collo di bottiglia
- Occorrono quindi soluzioni più veloci, tipicamente realizzate con hardware specializzato (ASIC)



## Architettura dei router moderni



- La CPU calcola la tabella di routing, esegue i protocolli e gestisce il sistema nel suo complesso
- Le interfacce memorizzano i pacchetti ed eseguono il lookup su copie locali della tabella (più o meno complete)
- La matrice di commutazione permette il trasferimento diretto e contemporaneo di più pacchetti
- Elaborazioni realizzate in hardware (ASIC)

---

---

---

---

---

---

---

---

## Matrice di commutazione

- Permettono più trasferimenti in parallelo

Esempi:

- **crossbar**
  - $N^2$  punti di incrocio aperti/chiusi da un dispositivo di controllo (**scheduler**)
  - matrice monostadio
  - non bloccante: è sempre possibile instaurare un percorso tra due terminazioni libere qualunque sia lo stato delle connessioni in atto
- **rete di Clos**
  - matrice multistadio costituita da elementi di commutazione non bloccanti di dimensioni inferiori
  - riduce il numero di punti di incrocio necessari (complessità)
  - non bloccante se si rispettano alcune condizioni
  - velocità di commutazione limitata dalla velocità dello scheduler

---

---

---

---

---

---

---

---

## Posizionamento dei buffer

- Nelle reti a commutazione di pacchetto in ogni nodo tipicamente si memorizzano i pacchetti prima di trasmetterli in uscita (**store and forward**)
  - memorie nelle interfacce di ingresso durante l'elaborazione dell'instestazione e il lookup
  - memorie nelle interfacce di uscita prima della trasmissione
- L'utilizzo di matrici di commutazione (anche se non bloccanti) per trasferire pacchetti necessita di code (**buffer**) per risolvere le contese tra pacchetti diretti verso la stessa porta di uscita
  - alternative:
    - **accodamento in uscita** (Output Queuing – OQ)
    - **accodamento in ingresso** (Input Queuing – IQ)

---

---

---

---

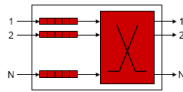
---

---

---

---

## Buffer in ingresso



- I pacchetti sono messi in coda **prima** di essere commutati
- Il pacchetto in testa alla coda viene trasferito quando l'uscita a cui è destinato risulta libera
- il pacchetto di testa attende che si liberi l'uscita a cui è destinato
- tutti gli altri pacchetti dietro di lui devono attendere che questo venga servito, anche se la loro uscita è disponibile (code **FIFO**)
- throughput massimo = 58%

---

---

---

---

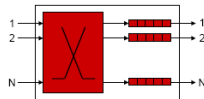
---

---

---

---

## Buffer in uscita



- I pacchetti sono messi in coda **dopo** essere stati commutati
- I pacchetti che trovano una linea di uscita occupata vengono messi in coda in attesa che essa si liberi
- Se arrivano N pacchetti destinati alla stessa uscita, la matrice di commutazione deve inserirli tutti nella relativa coda
- La velocità di commutazione della matrice deve essere almeno pari alla somma delle velocità di trasmissione dei pacchetti delle N linee di ingresso (**speed-up** con fattore N)
- questa soluzione può risultare molto costosa con linee ad elevata bit-rate

---

---

---

---

---

---

---

---

## Caratteristiche di routing

- Gli algoritmi di routing differiscono per le modalità con cui la tabella di instradamento viene creata ed eventualmente aggiornata nel tempo
- Un algoritmo di routing ideale dovrebbe presentare le seguenti caratteristiche:
  - corretto
  - semplice e veloce
  - robusto (rispetto a cambiamenti di topologia e di traffico)
  - stabile
  - equo
  - ottimale
  - scalabile
- **Tipologie di routing**
  - Statico
  - Dinamico
  - Centralizzato
  - Distribuito

---

---

---

---

---

---

---

---

## Routing statico e dinamico

### Routing statico

- I percorsi per l'inoltro dei pacchetti sono determinati (ed eventualmente cambiati) dall'amministratore di rete, che costruisce quindi la tabella di routing
- Il principale svantaggio del routing statico è l'impossibilità di reagire automaticamente ai cambiamenti della rete.
- Per questo si prevedono diversi percorsi per raggiungere un altro router della rete in modo da poter avere varie possibilità in caso di guasti o congestioni di nodi.

### Routing dinamico

- I percorsi per l'inoltro dei pacchetti sono individuati da un protocollo di routing e aggiornato automaticamente in funzione delle modifiche di topologia o di traffico.
- Risulta più complesso, ma presenta prestazioni superiori.
- Occorre che i router si scambino informazioni sulla rete (generalmente un router passa informazioni sullo stato dei suoi collegamenti ai router più vicini).

---

---

---

---

---

---

---

---

---

---

## Routing dinamico

- Le tabelle di instradamento vengono create e periodicamente aggiornate **in modo automatico**
  - adottano algoritmi adattativi
- Consentono di adattare le decisioni di instradamento a
  - variazioni topologiche della rete
    - inserimento di nuovi nodi o collegamenti
    - caduta di un nodo o collegamento per guasto
  - condizioni di traffico
    - si evita la scelta di percorsi che comprendono collegamenti congestionati

---

---

---

---

---

---

---

---

---

---

## Routing distribuito

- Ogni nodo calcola in modo autonomo le sue tabelle di instradamento
- Il calcolo può essere basato su informazioni:
  - locali
    - riguardanti il solo nodo in cui sta avvenendo il calcolo,
    - senza scambio di informazioni tra i nodi
  - distribuite
    - si utilizzano informazioni relative agli altri nodi e collegamenti della rete
- Nel caso di routing basato su informazioni distribuite deve essere previsto un meccanismo di scambio delle informazioni (protocollo)

---

---

---

---

---

---

---

---

---

---

## Routing IP

- Per quanto riguarda l'instradamento IP di un pacchetto si possono distinguere due casi :
  - **consegna diretta**: si usa questo termine nel caso in cui il datagram debba essere trasferito tra due host che sono connessi alla stessa rete fisica (o più precisamente, che siano raggiungibili attraversando solo apparati di livello 2 come switch, bridge o concentratori);
  - **consegna indiretta**: si usa questo termine per indicare il trasferimento di un datagram in cui sono coinvolti uno o più router, con gli host sorgente e destinazione connessi a reti differenti.
- Quando l'host deve trasmettere un datagram, esegue un AND bit a bit tra l'indirizzo di destinazione del datagram e la propria netmask, confrontando il risultato con quello della stessa operazione eseguita tra il proprio indirizzo IP e la propria netmask.
- Se i risultati coincidono e segno che l'host destinazione ha lo stesso Net-id dell'host sorgente, e quindi i due si trovano sulla stessa rete.
- La consegna diretta viene effettuata mappando l'indirizzo IP di destinazione nell'indirizzo di livello 2 ( algoritmo ARP)

---

---

---

---

---

---

---

---

## Routing statico

- Semplice da realizzare su reti di piccole dimensioni
- Adatto a reti con bassa ridondanza di collegamenti
- Difficile da gestire al crescere della dimensione della rete.
- Scarsa tolleranza ai guasti
  - Ogni variazione topologica impone la riconfigurazione delle tabelle nei nodi interessati
  - Ogni modifica richiede l'intervento dell'amministratore di rete
- Si possono distinguere due fasi:
  - Fase di progetto
  - Fase di configurazione

---

---

---

---

---

---

---

---

## Routing statico - Fase di progetto

- **Obiettivi della fase di progetto**
  - Individuazione dei percorsi per raggiungere le diverse destinazioni
  - I percorsi sono determinati tenendo conto di:
    - Topologia della rete
    - Velocità delle linee di trasmissione
    - Aspetti economici ( costi delle linee e vincoli amministrativi)
  - Tutte le destinazioni devono essere raggiungibili da qualunque nodo.
  - Devono essere evitati inoltri ciclici (routing loop)

---

---

---

---

---

---

---

---

## Tabella di progetto

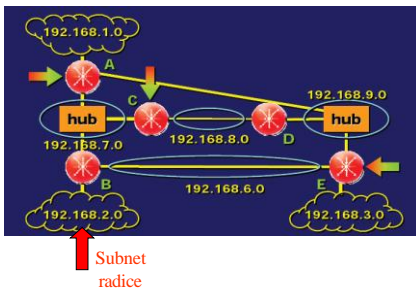
- La tabella di progetto contiene
  - Una riga per ogni rete di destinazione (o subnet IP)
  - Una colonna per ogni router

	Router 1	Router 2		Router M
Subnet 1				
Subnet 2				
Subnet N				

- La posizione (i,j) della tabella contiene il next hop usato dal router della j-ma colonna per raggiungere la destinazione presente nella i-ma riga
- L'amministratore di rete, partendo dal router radice, identifica un percorso per ciascuna delle subnet IP
- La tabella viene riempita per colonne successive

## Esempio

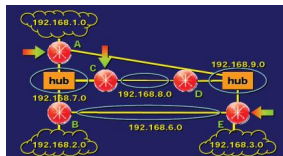
- Si considera le reti collegate mediante i router A,B,C,D,E mostrati nella figura
- Tutte le subnet hanno una netmask di 24 uni (classi C)
- Primo passo: scelta della subnet radice. Nell'esempio la subnet 192.168.7.0



## Esempio

- Secondo passo: Si decide attraverso quale nodo dell'albero ogni router vicino raggiungerà il nodo radice.

Subnet IP	A	B	C	D	E
192.168.1.0					
192.168.2.0	B	---	B		B
192.168.3.0					
192.168.6.0					
192.168.7.0					
192.168.8.0					
192.168.9.0					



## Metrica

- Costi dei link del grafo
- Esempio 1: Costo=1 assegnato a tutti i link del grafo**
  - Il percorso ottimo (a costo minimo) coincide con lo SHORTEST PATH
- Svantaggi:**
  - non considera la latenza (250 msec per i link satellitari, 1 msec per i link terrestri)
  - non considera la capacità dei link
  - non considera il carico dei link
- Esempio 2: Costo=coda sui link di uscita dei nodi**
- Svantaggi:**
  - non considera la latenza (250 msec per i link satellitari, 1 msec per i link terrestri)
  - non considera la capacità dei link
- Esempio 3: Costo=Tempo medio di ritardo nei nodi**
- Svantaggi:**
  - con carichi pesanti i link congestionati hanno costi molto alti
    - il traffico si sposta su altri link
    - Oscillazioni
  - range dei costi troppo elevati (per esempio un link satellitare potrebbe avere un costo 200 volte superiore a quello dei link terrestri e quindi restare scarico)

[illegible]

---

---

---

---

---

---

## Routing in Internet

---

---

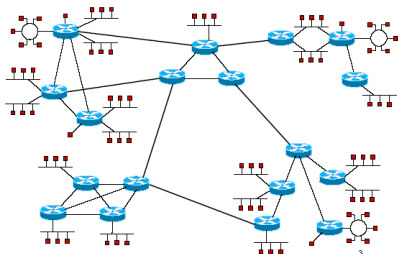
---

---

---

---

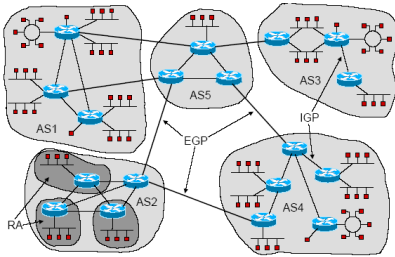
## Internet: rete di reti

[illegible]

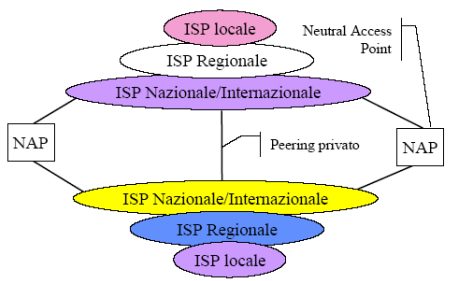


## Autonomous system (AS)

- Si organizza la rete in autonomous system (AS)



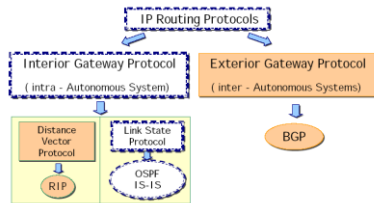
## Topologia in Internet



## Routing gerarchico in Internet

- In Internet si usa il routing gerarchico e le aree di routing sono chiamate **Autonomous System (AS)**
- un AS può essere ulteriormente suddiviso in porzioni dette **Routing Area (RA)** interconnesse da un **backbone** (dorsale)
- ogni network IP è tutta contenuta in un AS o in una RA
- oggi secondo il CIDR
  - gli AS decidono autonomamente i protocolli e le politiche di routing che intendono adottare al loro interno
  - i vari enti di gestione si devono accordare su quali protocolli utilizzare per il dialogo tra i router che interconnettono AS diversi
- I protocolli di routing all'interno di un AS sono detti
  - **Interior Gateway Protocol (IGP)**
  - I protocolli di routing fra AS sono detti **Exterior Gateway Protocol (EGP)**

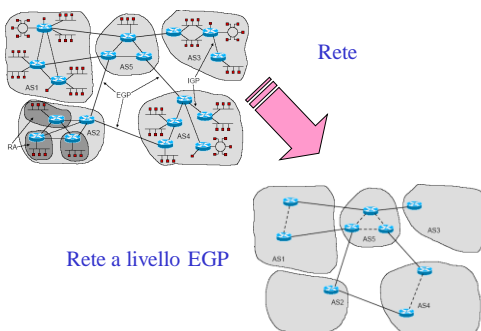
## Routing in Internet



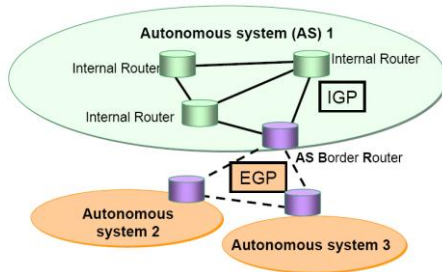
## Principali protocolli di routing

- Protocolli IGP
  - RIP: protocollo IGP appartenente alla classe dei protocolli distance vector
  - OSPF: protocollo IGP appartenente alla classe dei protocolli link vector
  - IGRP (Interior Gateway Routing Protocol) è stato sviluppato da CISCO per risolvere alcune limitazioni di RIP. Come RIP, è un protocollo "distance-vector".
  - IS-IS è molto simile ad OSPF (è un protocollo "link state")
- Protocolli EGP
  - EGP (Exterior Gateway Protocol) è stato il primo protocollo di routing interdominio utilizzato. Presenta molte limitazioni ed è stato quindi sostituito dal BGP
  - BGP (Border Gateway Protocol) è il protocollo di routing interdominio attualmente utilizzato (nella versione BGP-4)

## Rete a livello EGP



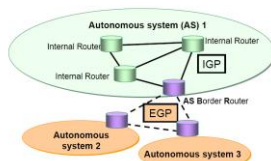
## Routing in Internet



## Autonomous System

- Un Autonomous Sistem era originariamente definito come un insieme di router gestiti da un'unica amministrazione (enti, aziende, università,...)
- Nuova definizione (1996 – RFC 1930): un AS è un gruppo connesso di una o più reti IP (classless) gestite da uno o più operatori ma con una **singola e ben definita** politica di routing
- **politica di routing di un AS**: modalità con cui si prendono decisioni di routing nel resto della rete Internet sulla base delle informazioni provenienti dall'AS (attraverso un EGP)
- **Esempio**:
  - Università di Bologna 137.204.0.0/16
  - Politecnico di Torino 130.192.0.0/16
  - entrambi comunicano con il resto del mondo tramite il GARR e le sue scelte di peering (**stessa politica di routing**)
  - non c'è bisogno di avere un AS per ogni ateneo
  - il GARR (e tutte le reti connesse ad esso) costituiscono un unico AS (AS137)

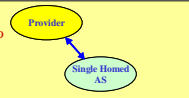
## Autonomous System



- Un AS con un solo border router è detto **Stub** o “single-homed” (piccole corporate)
- Un AS con più border router è detto “multi-homed”
  - **Transit** (provider): Accetta di essere attraversato da traffico diretto ad altri AS
  - **Nontransit** (grandi corporate): Non accetta di essere attraversato da traffico diretto ad altri AS
- Uno o più router interni a un AS sono selezionati per svolgere le funzioni di Exterior Gateway o ASBR (AS boundary router)
- Gli ASBR devono partecipare sia al protocollo di routing interno, sia a quello esterno per propagare le informazioni verso altri AS

## Routing in Internet

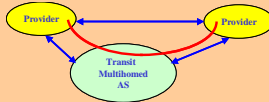
- Un AS con un solo border router è detto **Stub** o "single-homed" (piccole corporate)



- Un AS con più border router è detto "multi-homed"

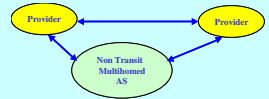
### Transit (provider)

- Accetta di essere attraversato da traffico diretto ad altri AS



### Nontransit (grandi corporate)

- Non accetta di essere attraversato da traffico diretto ad altri AS



## Autonomous System

- Ogni AS è responsabile del routing all'interno delle sue reti
  - Routing interno
- Gli AS devono scambiarsi informazioni di raggiungibilità
  - Routing esterno
- Garantisce la correttezza e la consistenza delle informazioni memorizzate nelle tabelle dei router
- Ogni AS deve essere identificato da un nome
- AS number (16 bit)

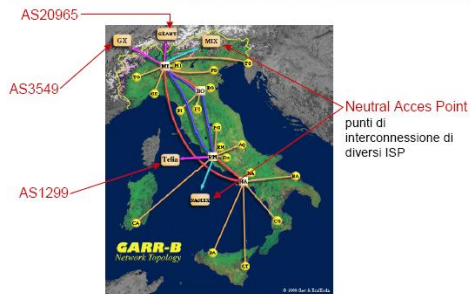
## Peering tra AS

- Gli accordi amministrativi (e i conseguenti accordi tecnici) tra i gestori di AS differenti per stabilire le politiche di transito e raggiungibilità sono detti accordi di *peering*
- Una relazione di peering si stabilisce tutte le volte che un EGP viene attivato tra due AS differenti
- Esistono generalmente siti particolari dove viene creato un punto di contatto tra diversi AS: NAP o Neutral Access Point

Topologia in Internet

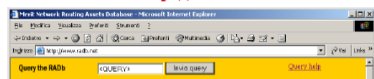


## GARR = AS137



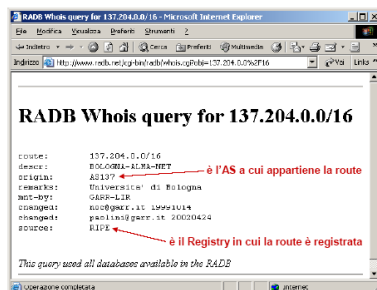
## Internet Routing Registries (IRRs)

- Sono database contenenti le politiche di routing degli AS allocati dai diversi Regional Internet Registry e da altre organizzazioni nazionali e internazionali
- Possono essere interrogati tramite il client "whois"
  - da shell linux: `whois -h whois.radb.net <QUERY>`
  - da client web: <http://www.radb.net>



- Alcuni esempi di query:
  - `a.b.c.d/m` per avere informazioni su una route
  - `ASxxxx` per avere informazioni su un AS
  - `!gASxxxx` per conoscere tutte le reti IP in un AS

## Esempio: whois 137.204.0.0/16



## Routing interno ed esterno

Il Routing interno ed esterno ad un Autonomous System

### Routing interno a un AS

- Le tabelle di routing interne di un AS sono mantenute dall'**Internal Gateway Protocol (IGP)**
  - i messaggi IGP sono scambiati tra routers appartenenti all'AS
  - Contengono solo informazioni sulle reti dell'AS.

### Routing tra AS ( routing esterno a un AS)

- Le tabelle di routing esterne di un AS sono mantenute dall'**External Gateway Protocol (EGP)**
  - i messaggi EGP sono scambiati tra router designati dai rispettivi AS (border router)
  - Contengono informazioni sulle rotte conosciute dai due AS

## Protocolli di routing in Internet

- Interior Gateway Protocol
  - ✓ **RIP**: **R**outing **I**nformation **P**rotocol (distance vector)
  - ✓ **OSPF**: **O**pen **S**hortest **P**ath **F**irst (link state)
- Exterior Gateway Protocol
  - ✓ **EGP**: **E**xterior **G**ateway **P**rotocol ( obsoleto)
  - ✓ **BGP**: **B**order **G**ateway **P**rotocol (path distance)

## Algoritmi di routing

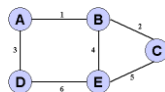
## Tempo di convergenza

- La convergenza si ottiene quando tutti i router hanno una visione consistente della topologia della rete
- Il processo ed il tempo richiesto per raggiungere la convergenza dipende dal protocollo di routing utilizzato

## Algoritmo Distance Vector

### Esempio

- Consideriamo la rete mostrata nella figura
- *Il costo di ogni link nell'esempio è uguale a 1*



- La rete è puramente esemplificativa: non differenziamo tra host e router, subnets e link.
- I nodi della nostra rete devono semplicemente essere in grado di replicare i pacchetti.
- Ogni nodo è identificato dal proprio indirizzo (nel caso A, B, C, D ed E) e la tabella di routing avrà una voce per ciascuno degli indirizzi. Per semplicità assumiamo che tutti i link siano simmetrici.

## Cold start

- Supponiamo di inizializzare la rete alimentando tutti i nodi contemporaneamente, quello che viene definito dagli specialisti un "cold start".
- In questo stato, ciascuno dei nodi e' caratterizzato unicamente da una conoscenza locale, che significa che ciascun nodo conosce il proprio indirizzo, ma ignora totalmente la topologia della rete.
- La tabella di routing sara' quindi minima, caratterizzata da una singola voce, quella del nodo stesso.

Esempio tabella di A

Da A a	Link	Costo
A	Locale	0

Esempio tabella di B

Da B a	Link	Costo
B	Locale	0

## Aggiornamento delle distanze

- B riceve, attraverso il link 1, il distance vector da A = 0.
- B aggiorna tutte le distanze aggiungendo loro il costo del link locale, che assumiamo essere uguale a 1 e trasforma quindi il messaggio da A = 0 ad A = 1.

Aggiornamento  
tabella di B

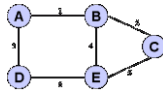
Da B a	Link	Costo
B	Locale	0
A	1	1

- Dopo l'aggiornamento della tabella B invia il proprio distance vector (B = 0, A = 1) sui suoi link locali, ossia 1, 2 e 4.
- Nel frattempo, anche il nodo D avra' ricevuto il messaggio iniziale da A e avra' aggiornato la propria tabella in :

Aggiornamento  
tabella di D

Da D a	Link	Costo
D	Locale	0
A	3	1

- Il nodo D inviera' il proprio distance vector (D = 0, A = 1) sui link 3 e 6.



## Aggiornamento delle distanze

- Il messaggio proveniente da B sara' ricevuto da A, C ed E, mentre quello da D sara' ricevuto da A ed E.
- Supponiamo, per comodita', che il messaggio da B sia ricevuto prima.
- Ricevendo il messaggio attraverso il link 1, A aggiorna' le distanze a B = 1 ed A = 2.
- Poiche' quest'ultima e' maggiore della voce relativa al link locale, la tabella di A verra' aggiornata con la sola voce relativa al nodo B.
- Infine, ricevendo il messaggio attraverso il link 3, verra' inserita anche l'informazione relativa al nodo D.

Tabella ricevuta da B e aggiornata

Da B a	Link	Costo
B	1	1
A	1	2

Tabella da A

Da A a	Link	Costo
A	Locale	0

Aggiornamento  
tabella di A

Da A a	Link	Costo
A	Locale	0
B	1	1
D	3	1



## Aggiornamento delle distanze

- A sua volta, il nodo C riceverà il vettore  $B = 0, A = 1$  attraverso il link 2, e aggiungerà la propria tabella:

Aggiornamento  
tabella di C

Da C a	Link	Costo
C	Locale	0
B	2	1
A	2	2

- Il nodo E riceverà il vettore  $B = 0, A = 1$  attraverso il link 4 e aggiungerà la propria tabella:

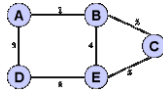
Aggiornamento  
tabella di E

Da E a	Link	Costo
E	Locale	0
B	4	1
A	4	2

- Il nodo E riceverà il vettore  $D = 0, A = 1$  attraverso il link 6, lo aggiungerà in  $D = 1, A = 2$ , osservando che la voce D dovrà essere inserita in tabella mentre la distanza da A attraverso il link 6 è esattamente uguale a quella già presente in tabella attraverso il link 4. Assumeremo che in questo caso la tabella non venga modificata e che quindi diventi quella mostrata nella figura.

Da E a	Link	Costo
E	Locale	0
B	4	1
A	4	2
D	6	1

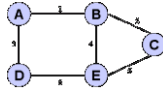
Aggiornamento  
tabella di E



## Aggiornamento delle distanze

- Dopo che A, C ed E avranno quindi calcolato le loro nuove tabelle, invieranno sui loro link locali i propri distance vectors

- Da A :  $A = 0, B = 1, D = 1$ , attraverso i link 1 e 3
- Da C :  $C = 0, B = 1, A = 2$ , attraverso i link 2 e 5
- Da E :  $E = 0, B = 1, A = 2, D = 1$  attraverso i link 4, 5 e 6 che causeranno un aggiornamento alle tabelle di B, D ed E



Aggiornamento tabelle

Da B a	Link	Costo
B	Locale	0
A	1	1
D	1	2
C	2	1
E	4	1

Da D a	Link	Costo
D	Locale	0
A	3	1
B	3	2
E	6	1

Da E a	Link	Costo
E	Locale	0
B	4	1
A	4	2
D	6	1
C	5	1

## Aggiornamento delle distanze

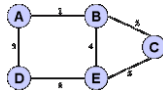
- Da B :  $B = 0, A = 1, D = 2, C = 1, E = 1$ , attraverso i link 1, 3 e 4
- Da D :  $D = 0, A = 1, B = 2, E = 1$ , attraverso i link 3 e 6
- Da E :  $E = 0, B = 1, A = 2, D = 1, C = 1$ , attraverso i link 4, 5 e 6
- Queste informazioni saranno ricevute da A, C e D che aggiungeranno quindi le loro tabelle

Da C a	Link	Costo
C	Locale	0
B	2	1
A	2	2
E	5	1
D	5	2

Da A a	Link	Costo
A	Locale	0
B	1	1
D	3	1
C	1	2
E	1	2

Da D a	Link	Costo
D	Locale	0
A	3	1
B	3	2
E	6	1
C	6	2

- A questo punto, l'algoritmo converge. A, C e D prepareranno dei nuovi vettori e li invieranno attraverso i link locali, ma la ricezione di questi ultimi non causeranno ulteriori aggiornamenti alle tabelle di B, D ed E.
- E' quindi possibile affermare che, attraverso queste operazioni di calcolo distribuito, i nodi hanno **rilevato la topologia della rete**



## Esempio di tabelle e distance vector

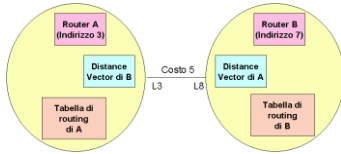


Tabella di routing di A

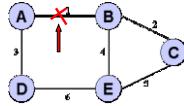
Indirizzo	Costo	Linea
1	25	3
2	20	2
3	0	0
4	15	3
5	55	1
6	23	1
7	5	3
...	...	...

Distance Vector di A in B

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

## Indisponibilità di un link

- Ipotizziamo che, per un qualsiasi motivo, il link 1 nella rete precedente diventi indisponibile.
- I due nodi A e B, che sono i terminali del ramo in esame, rilevano il guasto ed aggiornano di conseguenza le loro tabelle di routing, rilevando che il link1 assume ora un costo "infinito" e che tutti i nodi precedentemente raggiungibili attraverso tale link, sono ora collocati ad una distanza infinita.



Da A a	Link	Costo
A	Locale	0
B	1	infinito
D	3	infinito
C	1	infinito
E	4	infinito

Da B a	Link	Costo
B	Locale	0
A	1	infinito
D	1	infinito
C	2	1
E	4	1

## Indisponibilità di un link

- I nodi A e B invieranno i nuovi vettori .:
- Da A : A = 0, B = inf, D = 1, C = inf, E = inf, attraverso il link 3
- Da B : B = 0, A = inf, D = inf, C = 1, E = 1, attraverso i link 2 e 4
- Il vettore inviato da A sarà ricevuto da D, che aggiungerà le distanze aggiungendo il costo del ramo 3 in :
  - A = 1, B = inf, D = 2, C = inf, E = inf
- Gli elementi di questo vettore saranno quindi confrontati con le voci presenti nella tabella di D: verrà rilevato che tutte le distanze del vettore sono maggiori o al limite uguali a quelle memorizzate, ma che il ramo 3, attraverso il quale è stato ricevuto il vettore, è esattamente quello utilizzato per raggiungere B. Quindi, la voce per il nodo B dovrebbe essere aggiornata per riflettere il nuovo valore..

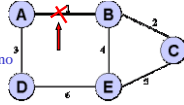
Da D a	Link	Costo
D	Locale	0
A	3	1
B	3	infinito
E	6	1
C	6	2

Da C a	Link	Costo
C	Locale	0
B	2	1
A	2	infinito
E	5	1
D	5	2

Da E a	Link	Costo
E	Locale	0
B	4	1
A	4	infinito
D	6	1
C	5	1

## Indisponibilità di un link

- I precedenti 3 nodi D,C,E) prepareranno ed invieranno quindi i nuovi distance vector



- Questi 3 nodi, prepareranno ed invieranno quindi i nuovi distance vector :
- Da D : D = 0, A = 1, B = inf, E = 1, C = 2, attraverso i link 3 e 6
- Da C : C = 0, B = 1, A = inf, E = 1, D = 2, attraverso i link 2 e 5
- Da E : E = 0, B = 1, A = inf, D = 1, C = 1, attraverso i link 4, 5 e 6
- Questi messaggi causeranno un aggiornamento alle tabelle dei nodi A, B, D ed E

Da A a	Link	Costo
A	Locale	0
B	1	infinito
D	3	1
C	3	3
E	3	2

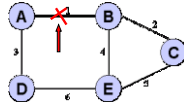
Da B a	Link	Costo
B	Locale	0
A	1	infinito
D	4	2
C	2	1
E	4	1

Da D a	Link	Costo
D	Locale	0
A	3	1
B	6	2
E	6	1
C	6	2

Da E a	Link	Costo
E	Locale	0
B	4	1
A	6	2
D	6	1
C	5	1

## Indisponibilità di un link

- Dopo gli aggiornamenti, i nodi invieranno i nuovi vettori



- Dopo gli aggiornamenti, i nodi invieranno i nuovi vettori
- Da A : A = 0, B = inf, D = 1, C = 3, E = 2, attraverso il link 3
- Da B : B = 0, A = inf, D = 2, C = 1, E = 1, attraverso i link 2 e 4
- Da D : D = 0, A = 1, B = 2, E = 1, C = 2, attraverso i link 3 e 6
- Da E : E = 0, B = 1, A = 2, D = 1, C = 1, attraverso i link 4, 5 e 6
- Queste informazioni provocano l'aggiornamento delle tabelle dei nodi A, B e C

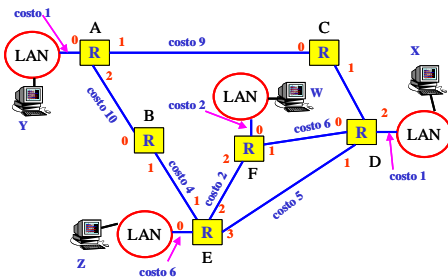
Da A a	Link	Costo
A	Locale	0
B	2	1
D	3	1
C	3	3
E	3	2

Da B a	Link	Costo
B	Locale	0
A	4	2
D	4	2
C	2	1
E	4	1

Da C a	Link	Costo
C	Locale	0
B	2	1
A	6	3
E	5	1
D	5	2

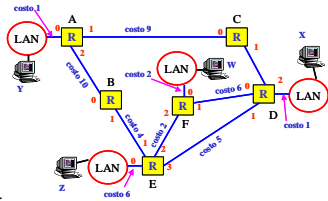
- I nodi A,B, e C invieranno i nuovi vettori ma questi non produrranno nessun nuovo aggiornamento delle tabelle di routing.
- Tutti i nodi sono raggiungibili e quindi è stato ristabilito il funzionamento completo della rete.

## Esempio



- In questo esempio i link hanno costi diversi

Inizio a freddo



- Nodo F: costruzione matrice

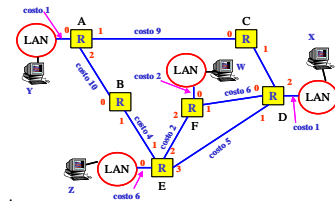
Indirizzo	Link	Costo
F	-----	0

Indirizzo	Link	Costo
W	-----	2

↓

Indirizzo	Link	Costo
F	-----	0
W	-----	2

Inizio a freddo



- Nodo E: costruzione matrice

Indirizzo	Link	Costo
F	-----	0

Indirizzo	Link	Costo
Z	-----	6

↓

Indirizzo	Link	Costo
F	-----	0
Z	-----	6

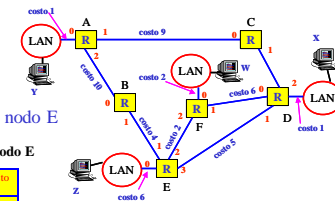
- Nodo A

Indirizzo	Link	Costo
A	-----	0
Y	-----	1

- Nodo B

Indirizzo	Link	Costo
B	-----	0

Inizio a freddo



- Aggiornamento matrice nodo E

Il nodo F invia la sua tabella al nodo E

Tabella originaria

Indirizzo	Link	Costo
F	-----	0
W	-----	2

Costo link = 2

Tabella modificata

Indirizzo	Link	Costo
F	2	2
W	0	4

## Inizio a freddo Passo successivo

- Aggiornamento matrice nodo E

Ricevuta da interfaccia 1

Indirizzo	Costo
B	4

Ricevuta da interfaccia 2

Indirizzo	Costo
F	2
W	4

Ricevuta da interfaccia 2

Indirizzo	Costo
D	3
X	6

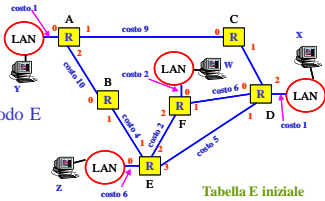


Tabella E iniziale

Indirizzo	Costo
E	0
Z	6

Indirizzo	Link Interfaccia	Costo
B	1	4
D	3	5
E		0
F	2	2
X	3	6
W	2	4
Z	6	0

Tabella nodo E

## Aggiornamento tabella nodo A

Distance vector  
nodo A

Indirizzo	Costo
A	0

Distance vector  
Interfaccia 0

Indirizzo	Costo
Y	1

Indirizzo	Link Interfaccia	Costo
A		0
B	1	4
C	9	9
D	3	5
E	6	6
F	2	2
X	3	6
W	2	4
Z	6	0

Fusione dei distance vector  
al nodo A

- Quando una tabella di routing di un nodo viene aggiornata essa viene inviata su tutte le proprie interfacce per consentire ai nodi vicini di aggiornare le proprie tabelle

Indirizzo	Costo
A	0
B	4
C	9
D	5
E	6
F	2
X	6
W	4
Z	0

Distance vector  
Interfaccia 2

Indirizzo	Costo
A	0
B	4
C	9
D	5
E	6
F	2
X	6
W	4
Z	0

Distance vector  
Interfaccia 1

## Malfunzionamento di un link

Malfunzionamento interfaccia 3  
del Nodo E

Interfaccia 1

Indirizzo	Costo
A	0
B	4
C	9
D	5
E	6
F	2
X	6
W	4
Z	0

Interfaccia 2

Indirizzo	Costo
A	0
B	4
C	9
D	5
E	6
F	2
X	6
W	4
Z	0

Indirizzo	Link Interfaccia	Costo
A		0
B	1	4
C	9	9
D	3	5
E	6	6
F	2	2
X	3	6
W	2	4
Z	6	0

Tabella modificata  
Tenendo presente il  
malfunzionamento della  
interfaccia 3

Indirizzo	Costo
A	0
B	4
C	9
D	5
E	6
F	2
X	6
W	4
Z	0

Indirizzo	Costo
A	0
B	4
C	9
D	5
E	6
F	2
X	6
W	4
Z	0

Interfaccia 0

Indirizzo	Costo
A	0
B	4
C	9
D	5
E	6
F	2
X	6
W	4
Z	0

Nodo E

- I percorsi per nodo C, D e per X sono ricalcolati trascurando i dati provenienti dall'interfaccia 3

## Proprietà dell'algoritmo di distance vector

- L'algoritmo di distance vector è uno dei primi ad essere stato utilizzato per i suoi meccanismi molto semplici di aggiornamento.
- Presenta numerosi limiti su reti a grandi dimensioni, quali:
  - Possono crearsi facilmente dei percorsi chiusi nell'instradamento dei pacchetti. Infatti i nodi conoscono soltanto il nodo adiacente a cui sarà instradato il pacchetto attraverso un'interfaccia, ma non conoscono il percorso del pacchetto
  - La quantità di informazione che i nodi devono scambiarsi è molto elevata: il distance vector di ogni nodo contiene l'elenco di tutte le destinazioni che esso è in grado di raggiungere e su una rete di grandi dimensioni queste possono essere un numero molto elevato.
  - La complessità dell'algoritmo (numero di passi prima che il routing converga) è uguale a  $N^2$ , dove  $N$  è il numero di nodi.

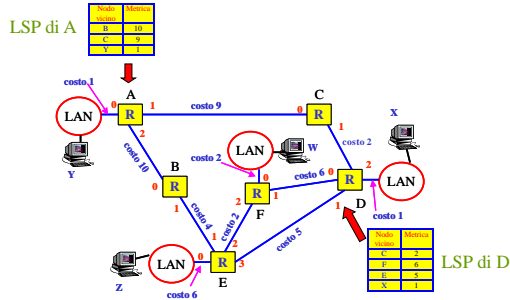
## Algoritmo Link State

### Generalità

- I protocolli di tipo Link State sono basati sul concetto di "mappa distribuita": tutti i nodi posseggono una copia della mappa della rete, che viene regolarmente aggiornata.
- Il funzionamento dell'algoritmo è basato sul fatto che ogni nodo della rete descrive la topologia della rete intorno a lui e invia questa descrizione in tutta la rete mediante opportuni pacchetti detti LSP (Link State Packet)
- Ogni nodo riceve i pacchetti LSP da tutti gli altri nodi e quindi può mettere insieme le varie descrizioni per costruire una propria mappa della rete.
- Mediante un opportuno algoritmo ogni nodo è in grado a questo punto di valutare sulla mappa il percorso migliore per raggiungere ogni destinazione della rete

## Esempio

Pacchetti LSP (Link State Packet): I pacchetti LSP di un nodo contengono l'identificatore del nodo e la distanza (espressa secondo un'opportuna metrica) di tutti i nodi collegati direttamente al nodo in esame



## Pacchetti LSP

- Stato di ogni link connesso al router
- Identità di ogni vicino connesso all'altro estremo del link (sulle LAN possono esserci migliaia di vicini)
- Costo del link
- Numero di sequenza per il LSP (a seguito di frequenti variazioni di topologia un router può ricevere un LSP vecchio dopo uno nuovo, quindi deve essere in grado di valutare il più recente)
- Checksum
- Lifetime (la validità di ogni LSP è limitata nel tempo, diversamente un errore sul numero di sequenza potrebbe rendere un LSP valido per anni)
- Un LSP viene generato periodicamente, oppure quando viene rilevata una variazione nella topologia locale (adiacenze), ossia :
  - Viene riconosciuto un nuovo vicino
  - Il costo verso un vicino e' cambiato
  - Si e' persa la connettività verso un vicino precedentemente raggiungibile

## Pacchetti LSP

- Il LSP è trasmesso in flooding su tutti i link del router e tutti i router del dominio di routing ricevono il LSP.
- All'atto del ricevimento di un LSP il router compie le seguenti azioni:
  - Se non ha mai ricevuto LSP da quel router o se il LSP è più recente di quello precedentemente memorizzato (campo Sequence Number), memorizza il pacchetto e lo ritrasmette in flooding su tutte le linee eccetto quella da cui l'ha ricevuto
  - Se il LSP ha lo stesso numero di sequenza di quello posseduto non fa nulla (scarta quello ricevuto successivamente perché possiede già una copia dello stesso LSP)
  - Se il pacchetto LSP è più vecchio di quello posseduto trasmette al mittente il pacchetto più recente

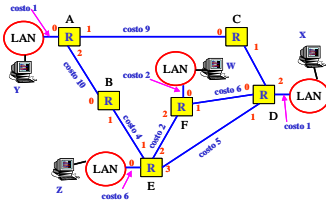
## Diffusione dei pacchetti LSP

- Il pacchetto LSP generato da un nodo (ad esempio il nodo A) sono inviati su tutta la rete

### Database LSP

Database LSP di ogni nodo

Minimo	Destinatari	Link Interfaccia	Costo
A	B	2	10
A	C	1	9
A	Y	0	1
E	A	0	10
B	E	1	4
C	A	2	9
C	D	1	2
D	C	1	2
D	E	1	4
D	F	0	6
E	X	2	6
E	B	1	4
F	E	2	2
E	Z	0	6
F	D	1	6
F	E	2	2
F	W	0	2



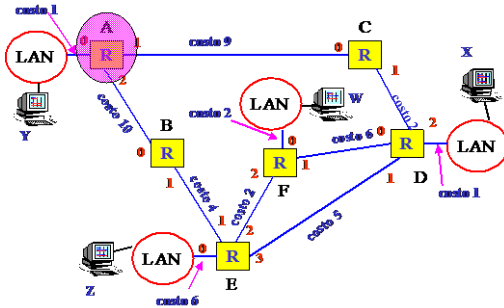
- A regime ogni nodo possiede il database mostrato in figura.

## Costruzione della mappa della rete

- A partire dalle informazioni contenute nei database ogni nodo è in grado di costruire una propria mappa della rete.
- La mappa della rete viene costruita mediante l'algoritmo di Dijkstra o Shortest Path First (SPF)



## Esempio Costruzione Albero nodo A



Si parte dal nodo A con costo 0 e si procede su percorsi a costo minimo

---

---

---

---

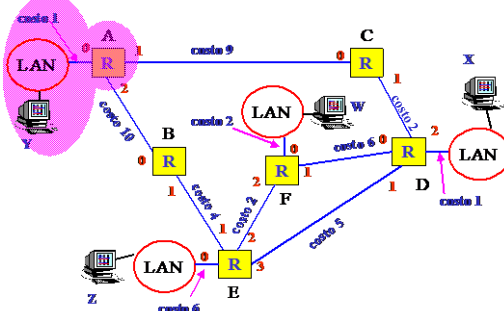
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considerano cammini a costo 1 partendo da A

---

---

---

---

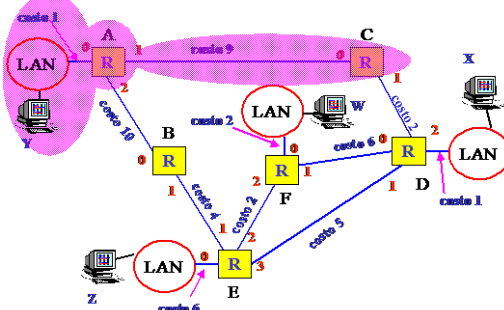
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considerano cammini a costo 9 partendo da A

---

---

---

---

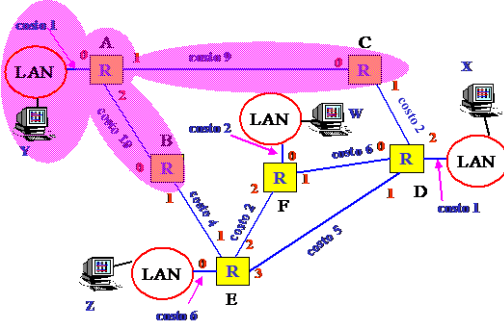
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considera il cammino a costo 10 partendo da A

---

---

---

---

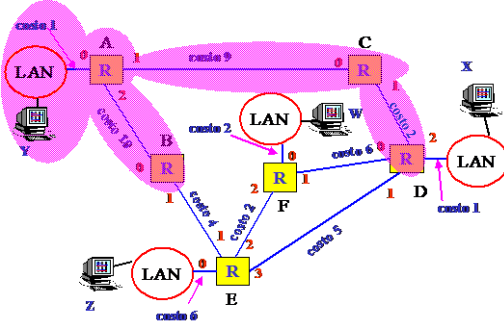
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considera il cammino a costo 12 partendo da A

---

---

---

---

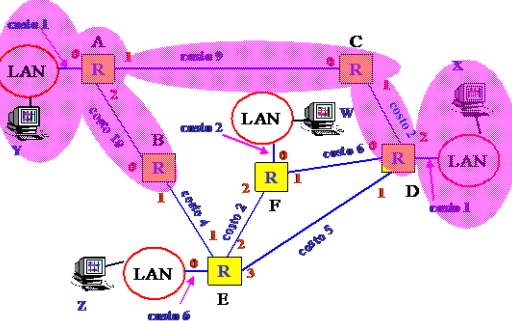
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considera il cammino a costo 13 partendo da A

---

---

---

---

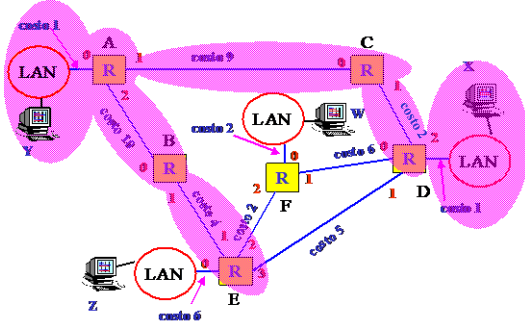
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considera il cammino a costo 14 partendo da A

---

---

---

---

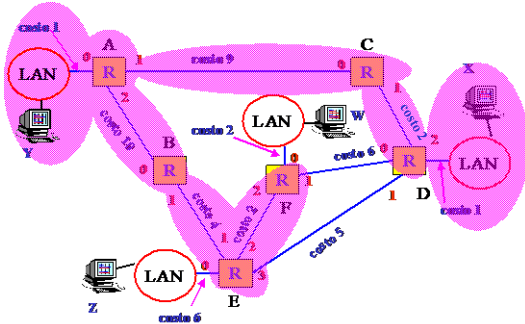
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considera il cammino a costo 16 partendo da A

---

---

---

---

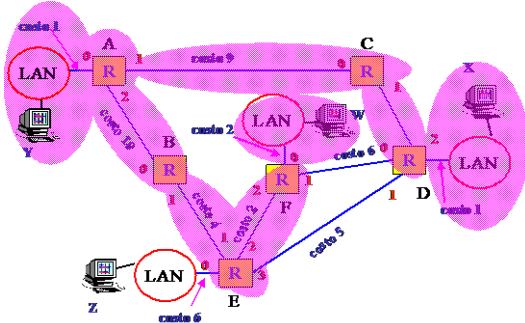
---

---

---

---

## Esempio Costruzione Albero nodo A



Si considera il cammino a costo 12 partendo da A

---

---

---

---

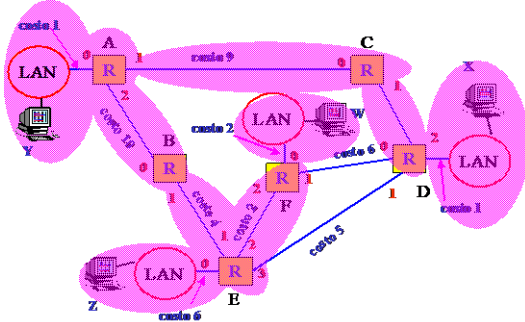
---

---

---

---

### Esempio Costruzione Albero nodo A



L'albero partendo da A è completato

### Caratteristiche dell'algoritmo di link state

- L'algoritmo di link state è caratterizzato da un'elevata stabilità e quindi brevi tempi di convergenza e bassissima probabilità di routing loop.
- Queste caratteristiche sono una conseguenza di:
  - La complessità dell'algoritmo di Dijkstra, cioè il numero di passi di calcolo che portano alla costruzione dello shortest path tree è  $C \log N$ , dove  $C$  = numero di link di rete,  $N$  = numero di nodi
  - I nodi calcolano la tabella di routing e quindi lo shortest path tree in modo indipendente. I nodi più lenti non influenzano le prestazioni dei nodi più veloci.
  - Nel caso di un malfunzionamento, i nodi vicini cambiano immediatamente la propria tabella di routing
  - La struttura ad albero dovrebbe ridurre al minimo la probabilità di loop.

### Protocollo RIP

- Il protocollo RIP (routing Information Protocol) è il protocollo IGP (Interior Gateway Protocol) maggiormente utilizzato oggi su Internet.
- Esso viene utilizzato perciò nell'interno di un AS
- RIP è un protocollo relativamente semplice e appartenente alla famiglia dei protocolli distance vector.
- Adatto a reti di piccole dimensioni, mentre è molto meno adatto a reti complesse.

- Esistono due versioni del protocollo:

- **Versione 1 (RFC 1058)**
- **Versione 2 (RFC 1723, 2453)**

## Protocollo RIP

- Il protocollo RIP prevede l'invio di messaggi routing update con le seguenti caratteristiche:
  - Ogni 30 sec
  - A seguito di cambiamenti della topologia
- I router partecipano al sistema di routing distribuito inviando messaggi di **RIP Update** che sono inviati a tutti gli altri router.
- Ogni RIP Update contiene il distance vector del router Mittente
- I router memorizzano l'ultimo distance vector ricevuto per ogni interfaccia

---

---

---

---

---

---

---

---

## Protocollo RIP

- Un router modifica la propria tabella di routing se:
  - Cade una linea attiva
  - Riceve un distance vector da un nodo adiacente diverso dall'ultimo memorizzato.
- Il calcolo consiste nella fusione (*merge*) di tutti i distance vector delle linee attive
- Se la tabella risulta diversa da quella precedente, il router invia ai nodi adiacenti un nuovo distance vector

---

---

---

---

---

---

---

---

## Caratteristiche del protocollo

### RIP versione 1

- Un'unica netmask per ciascuna rete IP (*classful routing*) - (RFC 1058)

---

---

---

---

---

---

---

---

### RIP versione 2

- Annuncia le netmask (RFC 1723, 2453) (RFC 1723, 2453)
- Supporta l'impiego di netmask diverse (*classless routing*)

---

---

---

---

---

---

---

---

## Metрика utilizzata in RIP

- In RIP la metрика è molto semplice: la distanza ( hop count) è il numero di link che vengono attraversati per raggiungere la destinazione.
- La metрика ( hop count) può variare da 1 a 15; il valore 16 indica una distanza infinita.
- Può attraversare fino a 15 router dati i limiti sulla metрика
- *La distanza 16 indica che una rete non è raggiungibile*
- Ogni router è connesso a una o più reti (*directly connected networks*)
- Per esse la metрика è impostata a 0.

---

---

---

---

---

---

---

---

## Tabella di routing

- Ogni riga della tabella di routing contiene le seguenti informazioni:

- Indirizzo di destinazione
- Metрика associata con la destinazione
- Indirizzo del "next router" ( next hop router)
- Un "recently updated" flag
- Numerosi timer



Informazioni del distance vector

---

---

---

---

---

---

---

---

## Timer

- Il protocollo RIP gestisce i seguenti timer:
  - Routing update timer (default 30 s): intervallo di tempo per l'invio degli annunci
  - Route invalid timer (default 90 s): intervallo di tempo dopo il quale una route è dichiarata irraggiungibile (distanza posta ad infinito)
  - Route flush timer (default 270 s): intervallo di tempo dopo il quale la route è cancellata dalla routing table
  - Triggered updates: sono inviate con un ritardo casuale compreso tra 1 e 5 secondi, per evitare intasamenti della rete e per far si' di poter eventualmente comunicare piu' cambi di route con un messaggio solo.

---

---

---

---

---

---

---

---

## RIP Update

- Ogni RIP Update contiene il distance vector del router mittente
- I router memorizzano l'ultimo distance vector ricevuto per ogni interfaccia di rete
- Il routing è tipo dinamico
- Un router modifica la propria tabella di routing se:
  - Cade una linea attiva
  - Riceve un distance vector da un nodo adiacente diverso dall'ultimo memorizzato
- Il calcolo consiste nella fusione (*merge*) di tutti i distance vector delle linee attive
- Se la tabella risulta diversa da quella precedente, il router invia ai nodi adiacenti un nuovo distance vector

---

---

---

---

---

---

---

---

## RIP Update

- I messaggi di routing update sono "imbustati" all'interno di messaggi di protocollo di trasporto UDP attraverso la porta 520.
- Quando un router RIP riceve un pacchetto IP esso controlla se contiene una busta UDP con porta 520.
- In caso affermativo il pacchetto viene passato al processo di routing.
- In caso negativo il pacchetto viene passato al processo di inoltro che consulta la tabella di routing per decidere su quale interfaccia inoltrare il pacchetto ricevuto.

---

---

---

---

---

---

---

---

## RIP versione 1

- Un'unica netmask per ciascuna rete IP (*classful routing*)

---

---

## RIP versione 2

- Consente l'utilizzo delle netmask
- Un'unica netmask per ciascuna rete IP (*classful routing*)

---

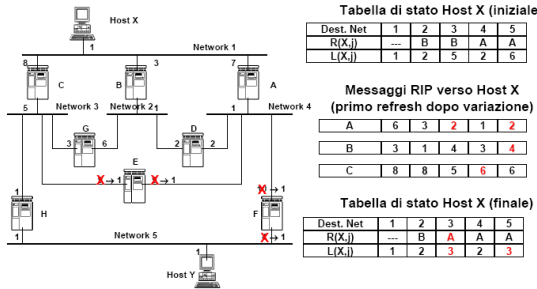
---

---

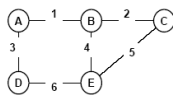
---

---

## Esempio: Inizializzazione



## Esempio : Inizializzazione



- Condizione iniziale
  - Routing table vuote
- Metrica
  - Distanza

Routing Table

Destinazione	A	B	C	D	E
Distanza	0	?	?	?	?
Link	local	?	?	?	?

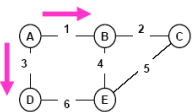
Destinazione	A	B	C	D	E
Distanza	?	0	?	?	?
Link	?	local	?	?	?

Destinazione	A	B	C	D	E
Distanza	?	?	0	?	?
Link	?	?	local	?	?

Destinazione	A	B	C	D	E
Distanza	?	?	?	0	?
Link	?	?	?	local	?

Destinazione	A	B	C	D	E
Distanza	?	?	?	?	0
Link	?	?	?	?	local

## Esempio : Inizializzazione



- Step 2:
  - A emette un messaggio verso B e D

Address	A	---	---	---	---
Metric	0	---	---	---	---

Routing Table

Destinazione	A	B	C	D	E
Distanza	0	?	?	?	?
Link	local	?	?	?	?

Destinazione	A	B	C	D	E
Distanza	1	0	?	?	?
Link	1	local	?	?	?

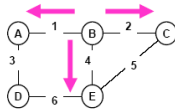
Destinazione	A	B	C	D	E
Distanza	?	?	0	?	?
Link	?	?	local	?	?

Destinazione	A	B	C	D	E
Distanza	1	?	?	0	?
Link	3	?	?	local	?

Destinazione	A	B	C	D	E
Distanza	?	?	?	?	0
Link	?	?	?	?	local



## Esempio



- Step 3:
  - B emette un messaggio verso A, C e E

<b>B</b>	Address	A	B	---	---	---
	Metric	1	0	---	---	---

Routing Table

<b>A</b>	Destinazione	A	B	C	D	E
	Distanza	0	1	?	?	?
	Link	local	1	?	?	?

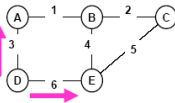
<b>B</b>	Destinazione	A	B	C	D	E
	Distanza	1	0	?	?	?
	Link	1	local	?	?	?

<b>C</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	0	?	?
	Link	2	2	local	?	?

<b>D</b>	Destinazione	A	B	C	D	E
	Distanza	1	?	?	0	?
	Link	3	?	?	local	?

<b>E</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	?	?	0
	Link	4	4	?	?	local

## Esempio



- Step 4:
  - D emette un messaggio verso A e E

<b>D</b>	Address	A	---	---	D	---
	Metric	1	---	---	0	---

Routing Table

<b>A</b>	Destinazione	A	B	C	D	E
	Distanza	0	1	?	1	?
	Link	local	1	?	3	?

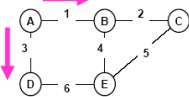
<b>B</b>	Destinazione	A	B	C	D	E
	Distanza	1	0	?	?	?
	Link	1	local	?	?	?

<b>C</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	0	?	?
	Link	2	2	local	?	?

<b>D</b>	Destinazione	A	B	C	D	E
	Distanza	1	?	?	0	?
	Link	3	?	?	local	?

<b>E</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	?	1	0
	Link	4	4	?	6	local

## Esempio



- Step 5:
  - A emette un messaggio verso B e D

<b>A</b>	Address	A	B	---	D	---
	Metric	0	1	---	1	---

Routing Table

<b>A</b>	Destinazione	A	B	C	D	E
	Distanza	0	1	?	1	?
	Link	local	1	?	3	?

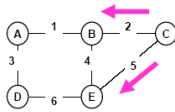
<b>B</b>	Destinazione	A	B	C	D	E
	Distanza	1	0	?	2	?
	Link	1	local	?	1	?

<b>C</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	0	?	?
	Link	2	2	local	?	?

<b>D</b>	Destinazione	A	B	C	D	E
	Distanza	1	?	?	0	?
	Link	3	?	?	local	?

<b>E</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	?	1	0
	Link	4	4	?	6	local

## Esempio



- Step 6:  
- C emette un messaggio verso B e E

C	Address	A	B	C	---	---
	Metric	2	1	0	---	---

Routing Table

A	Destinazione	A	B	C	D	E
	Distanza	0	1	?	1	?
	Link	local	1	?	3	?

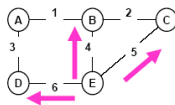
B	Destinazione	A	B	C	D	E
	Distanza	1	0	1	2	?
	Link	1	local	2	1	?

C	Destinazione	A	B	C	D	E
	Distanza	2	1	0	?	?
	Link	2	2	local	?	?

D	Destinazione	A	B	C	D	E
	Distanza	1	2	?	0	?
	Link	3	3	?	local	?

E	Destinazione	A	B	C	D	E
	Distanza	2	1	1	1	0
	Link	4	4	5	6	local

## Esempio



- Step 7:  
- E emette un messaggio verso B, C e D

E	Address	A	B	C	D	E
	Metric	2	1	1	1	0

Routing Table

A	Destinazione	A	B	C	D	E
	Distanza	0	1	?	1	?
	Link	local	1	?	3	?

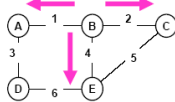
B	Destinazione	A	B	C	D	E
	Distanza	1	0	1	2	1
	Link	1	local	2	1	4

C	Destinazione	A	B	C	D	E
	Distanza	2	1	0	2	1
	Link	2	2	local	5	5

D	Destinazione	A	B	C	D	E
	Distanza	1	2	2	0	1
	Link	3	3	6	local	6

E	Destinazione	A	B	C	D	E
	Distanza	2	1	1	1	0
	Link	4	4	5	6	local

## Esempio



- Step 8:  
- B emette un messaggio verso A, C e E

B	Address	A	B	C	D	E
	Metric	1	0	1	2	1

Routing Table

A	Destinazione	A	B	C	D	E
	Distanza	0	1	2	1	2
	Link	local	1	1	3	1

B	Destinazione	A	B	C	D	E
	Distanza	1	0	1	2	1
	Link	1	local	2	1	4

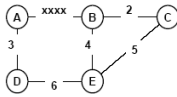
C	Destinazione	A	B	C	D	E
	Distanza	2	1	0	2	1
	Link	2	2	local	5	5

D	Destinazione	A	B	C	D	E
	Distanza	1	2	2	0	1
	Link	3	3	6	local	6

E	Destinazione	A	B	C	D	E
	Distanza	2	1	1	1	0
	Link	4	4	5	6	local

### Esempio: guasto di un link

Supponiamo che si verifichi un guasto nel link che collega A a B



- Condizione iniziale
  - rete a regime
  - guasto del ramo AB
- Metrica
  - Distanza

Routing Table

Destinazione	A	B	C	D	E
Distanza	0	inf	inf	1	inf
Link	local	1	1	3	1

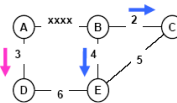
Destinazione	A	B	C	D	E
Distanza	inf	0	1	inf	1
Link	1	local	2	1	4

Destinazione	A	B	C	D	E
Distanza	2	1	0	2	1
Link	2	2	local	5	5

Destinazione	A	B	C	D	E
Distanza	1	2	2	0	1
Link	3	3	6	local	6

Destinazione	A	B	C	D	E
Distanza	2	1	1	1	0
Link	4	4	5	6	local

### Esempio: guasto di un link



- Step 1
  - Messaggio di A verso D

Address	A	B	C	D	E
Metric	0	inf	inf	1	inf

Address	A	B	C	D	E
Metric	inf	0	1	inf	1

Routing Table

Destinazione	A	B	C	D	E
Distanza	0	inf	inf	1	inf
Link	local	1	1	3	1

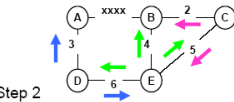
Destinazione	A	B	C	D	E
Distanza	inf	0	1	inf	1
Link	1	local	2	1	4

Destinazione	A	B	C	D	E
Distanza	inf	1	0	2	1
Link	2	2	local	5	5

Destinazione	A	B	C	D	E
Distanza	1	inf	2	0	1
Link	3	3	6	local	6

Destinazione	A	B	C	D	E
Distanza	inf	1	1	1	0
Link	4	4	5	6	local

### Esempio: guasto di un link



- Step 2
  - Messaggio di C verso B, E

Address	A	B	C	D	E
Metric	inf	1	0	2	inf

Address	A	B	C	D	E
Metric	1	inf	2	0	1

Address	A	B	C	D	E
Metric	inf	1	1	1	0

Routing Table

Destinazione	A	B	C	D	E
Distanza	0	inf	3	1	2
Link	local	1	3	3	3

Destinazione	A	B	C	D	E
Distanza	inf	0	1	2	1
Link	1	local	2	4	4

Destinazione	A	B	C	D	E
Distanza	inf	1	0	2	1
Link	2	2	local	5	5

Destinazione	A	B	C	D	E
Distanza	1	2	2	0	1
Link	3	6	6	local	6

Destinazione	A	B	C	D	E
Distanza	2	1	1	1	0
Link	6	4	5	6	local

## Esempio: guasto di un link

- Step 3
  - Messaggio di A verso D

A	Address	A	B	C	D	E
	Metric	0	inf	3	1	2

B	Address	A	B	C	D	E
	Metric	inf	0	1	2	1

D	Address	A	B	C	D	E
	Metric	1	2	2	0	1

E	Address	A	B	C	D	E
	Metric	2	1	1	1	0

### Routing Table

A	Destinazione	A	B	C	D	E
	Distanza	0	3	3	1	2
	Link	local	3	3	3	3

B	Destinazione	A	B	C	D	E
	Distanza	3	0	1	2	1
	Link	4	local	2	4	4

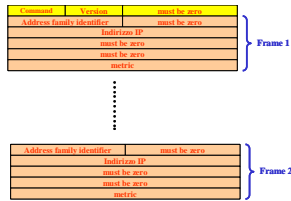
C	Destinazione	A	B	C	D	E
	Distanza	3	1	0	2	1
	Link	5	2	local	5	5

D	Destinazione	A	B	C	D	E
	Distanza	1	2	2	0	1
	Link	3	6	6	local	6

E	Destinazione	A	B	C	D	E
	Distanza	2	1	1	1	0
	Link	6	4	5	6	local

## RIP v 1 – Formato del routing update

- Il messaggio di routing update può essere composta da 1 a 25 frame descrittori della route.
- Qualora la route da annunciare richieda più di 25 frame le rimanenti vengono inserite in un messaggio successivo.
- Ogni messaggio è imbastito in una trama UDP con l'indirizzo (255,255,255,255) come destinazione.



- Command:** indica se è una richiesta o una risposta. Se è una richiesta il router invia la sua routing table.
- Address family identifier** indica il protocollo utilizzato, per cui possono essere utilizzati anche protocolli diversi da IP. Nel caso di protocollo IP il campo contiene il valore 2.

## RIP v.1

- Il limite maggiore del RIP v.1 è rappresentato dal fatto che tale algoritmo si basa sull'ipotesi che una destinazione sia identificata soltanto mediante il prefisso naturale (non sono considerate le netmask)

## RIP v.2

- L'algoritmo RIP è stato superato da algoritmi successivi basati sul concetto di link state, ad esempio OSPF.
- Tuttavia anche se gli algoritmi link state sono superiori in prestazioni RIP mantiene un notevole interesse per alcuni vantaggi che è in grado di offrire, quali:
  - il piccolo overhead in termini di occupazione di banda su piccole reti
  - la semplicità di implementazione, configurazione e gestione del protocollo
  - i bassi tempi richiesti a questo scopo.
- Per questi motivi è stata sviluppata una versione del protocollo, indicata con RIP 2, che cerca di eliminare alcuni inconvenienti di RIP v.1.
- La versione RIP 2, definita dalla RFC 1723, adotta l'algoritmo di routing distance, ma definisce in modo diverso il messaggio di routing update

---

---

---

---

---

---

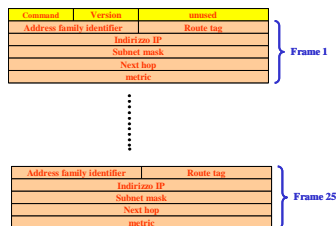
---

---

## Messaggio routing update in RIP 2

- Nel messaggio è contenuto oltre all'indirizzo IP anche la subnet mask

- **Command:** lo stesso significato che in RIP 1
- **Address family identifier:** come in RIP; se il campo assume il valore xFFFF è presente una password di autenticazione
- **Route tag:** serve per distinguere le route di tipo interno (apprese dai messaggi RIP) da altri protocolli di routing eventualmente di tipo exterior quale il BGP




---

---

---

---

---

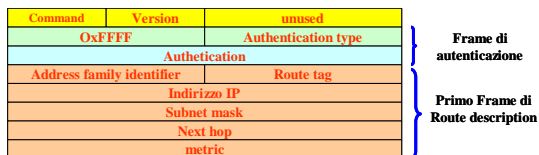
---

---

---

## Messaggio routing update in RIP 2 con autenticazione

- RIP 2 permette di autenticare i messaggi di routing inserendo a questo scopo alcune informazioni di autenticazioni nella testata del messaggio



- Il campo authentication type indica il tipo di algoritmo di autenticazione che sarà utilizzato.
- I messaggi RIP 2 sono imbustati in un pacchetto IP destinato all'indirizzo multicast 224.0.0.9

---

---

---

---

---

---

---

---

## IGRP (Interior Gateway Routing Protocol)

- Il protocollo IGRP e' un protocollo di routing sviluppato dalla CISCO a metà '80
- Negli anni '90 è stata sviluppata una versione migliorata (enhanced)
- Anche questo protocollo, così come il protocollo RIP, si basa sull'algoritmo distance vector; tuttavia vi sono alcuni aspetti che lo rendono, rispetto al RIP, più efficiente.
- In particolare, tali aspetti possono essere identificati in:
  - Metriche più sofisticate
  - Supporto del multipath routing
  - Migliore stabilità
- Inoltre, rispetto a RIP, IGRP utilizza una frequenza di update delle route inferiore (90 s).

## Metriche IGRP

- IGRP utilizza un insieme di parametri per determinare la metrica nella tabella di routing, al contrario di RIP che utilizza soltanto l'hop count.
- In particolare IGRP utilizza quattro parametri:
  - B – banda
  - D – ritardo
  - R – affidabilità
  - L – carico
- La variazione dei coefficienti permette di privilegiare determinati parametri a scapito di altri (es. banda piuttosto che ritardo)

## Metriche IGRP

- **Banda:** Minima sul percorso
- **Ritardo:** somma sul percorso
- **Affidabilità:** peggiore sul percorso
- **Carico:** più alto sul percorso
- Il gestore può pesare diversamente le 4 metriche mediante 5 parametri (k1, k2, k3, k4, k5) a seconda del tipo di servizio (ToS=Type of Service)

## EIGRP ( Enhanced Interior Gateway Routing Protocol)

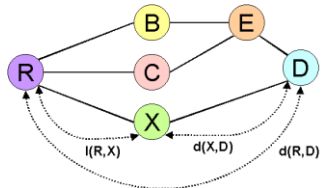
- Il protocollo EIGRP (Enhanced IGRP) e' stato sviluppato da CISCO a partire dalla release software 9.21 sulle basi del protocollo IGRP, rispetto al quale sono stati introdotti i seguenti miglioramenti:
  - Convergenza più rapida
  - Maschere a lunghezza variabile
  - Minore traffico di routing
  - Trasferimento di dati affidabile (Reliable Transfer Protocol)
  - Supporto multiprotocollo
  - Loop Free (ma si possono verificare dei Black Hole)
- EIGRP non prevede updates periodici;
- Ogni router mantiene una copia delle routing table di tutti i router adiacenti .

## EIGRP

- L'algoritmo è composto da due parti:
  - Selezione del router vicino accettabile
  - Attivazione del processo di diffusione
- Il principio di funzionamento dell'algoritmo e' il seguente: quando un router riceve un routing update verifica se il nuovo costo è minore del precedente e, in questo caso, aggiorna la route, propaga la nuove informazione ai vicini e termina.
- Il principio generale e' che un cambio di route di tipo "peggiorativo" viene preso in considerazione solo se proviene dallo stesso router che attualmente viene usato per raggiungere quella destinazione.
- Se il nuovo costo è maggiore, verifica se esiste un "vicino accettabile": se questo non esiste attiva il processo di diffusione che consiste nel propagare l'informazione "ho dei problemi di connettività su un link" ai vicini, fino a quando qualcuno è in grado di segnalare un nuovo cammino.

## EIGRP

- Ogni router R mantiene per ogni destinazione D e per ogni suo router adiacente le seguenti informazioni:
  - $d(X, D)$ : distanza tra ogni router adiacente e la destinazione
  - $l(R, X)$ : costo del link tra R ed X



- Il router R sceglie come percorso migliore per trasferire i pacchetti a D il router adiacente X per cui:
 
$$\min [ l(R,X) + d(X,D) ]$$

## EIGRP

- Si supponga, a questo punto, che il router R riceva da un router adiacente K un messaggio indicante che il costo del link  $l(R,K)$  è cambiato oppure che il costo verso una destinazione  $d(K,D)$  è cambiato.
- Per la regola del "costo minore verso una destinazione":
  - se il nuovo costo verso la destinazione è minore del precedente in possesso di R, si adotta come next hop il router K e si aggiorna le route
  - se il nuovo costo è superiore a quello in possesso di R:
    - Se K non è il router X (che era il precedente next hop verso D) scarta l'informazione
    - Se K è il router X applica la regola del "vicino accettabile" descritta nella slide successiva

---

---

---

---

---

---

---

---

---

---

## EIGRP • Regola del vicino accettabile

- Cerca tra tutti i router adiacenti X se esiste un router K tale che la sua distanza dalla destinazione D sia minore della distanza precedentemente in possesso di R:  $d(K,D) < d(R,D)$
- Questo significa che il nuovo router K è in grado di raggiungere la destinazione D attraverso un percorso alternativo senza passare da R e quindi il costo  $d(K,D)$  non dipende dal costo  $d(R,D)$
- Se non esiste alcun vicino adiacente che soddisfa la relazione il router R deve attivare il processo di "Diffusione"

---

---

---

---

---

---

---

---

---

---

## OSPF ( Open Shortest Path First)

- OSPF è basato sull'algoritmo di link state e rappresenta attualmente l'algoritmo più diffuso per le reti basate sul TCP/IP
- Definito dall'IETF con la RFC 1247 (1991) e la RFC 1583 (1994), ne sono state definite varie versioni, di cui la più recente è definita da RFC 2328
- OSPF è in grado di gestire reti di grandi dimensioni, ma è alquanto più complesso del RIP.
- Il dominio di routing è rappresentato da Autonomous System ( AS). Un AS viene diviso in aree per poter realizzare una soluzione gerarchica.
- Ogni area è caratterizzata mediante un numero di 4 ottetti ( spesso scritti in esadecimale con i caratteri separati da punto).
- OSPF opera in modo indipendente in ogni area e prevede meccanismi per lo scambio di informazioni tra le aree.

---

---

---

---

---

---

---

---

---

---



## OSPF

### Obiettivi di OSPF

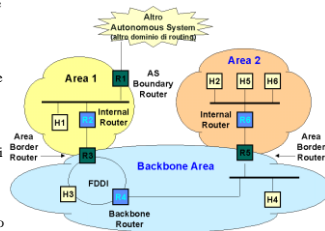
- Superare i limiti del RIP
- Avere metriche più efficienti (migliori possibilità di descrivere la rete)
- Introdurre delle gerarchie (per maggior scalabilità)
- Separazione informazioni interne ed esterne (all'AS)
- Supporto subnetting variabile (CIDR)
- Sicurezza

## Area unica

- Quando la rete è costituita da meno di 50 router si può realizzare un'area unica
  - bisogna considerare anche la complessità della rete ed il numero di maglie presenti
  - se i router hanno basse prestazioni sarebbe consigliabile avere aree con non più di 20 router
- Se la rete è costituita da un numero contenuto di router, ma l'azienda è in grande espansione conviene strutturarla con backbone e aree
- Incapsulato direttamente in IP, cioè un pacchetto OSPF ha una normale intestazione IP
  - il valore del campo protocol dell'intestazione IP (89 per OSPF) serve a distinguere questi pacchetti da altri

## Terminologia

- *Autonomous System (AS)* : dominio di routing
- *AS boundary router (ASBR)*: router collegato a router appartenenti ad altri AS
- *Link State Advertisement (LSA)*: : diffondono informazioni sullo stato dei link
  - equivalenti ai tradizionali LSP
  - non sono pacchetti
- *Backbone*: area di transito tra le altre aree
- *Backbone router* : router che è nel backbone
- *Area border router (ABR)*: backbone router che si affaccia su più aree
  - esegue una copia dell'algoritmo per ogni area
- *Internal router*: router che fa parte di un'area diversa dal backbone
- **Vicini**: due router che sono connessi alla medesima rete e possono comunicare direttamente punto-punto o punto-multipunto
- **Adiacenti**: due router che si scambiano informazioni di routing

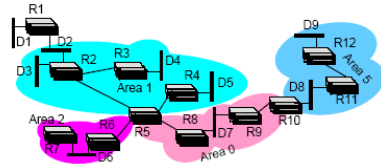


## Visione della rete

- In OSPF, come in ogni algoritmo del tipo link state, ogni router costruisce la propria visione della rete da cui viene ricavata la tabella di routing.
- Poiché in OSPF il routing è di tipo gerarchico i router non hanno tutti la stessa mappa della rete
- Tutti i router che si trovano in una certa area hanno la stessa mappa della rete e quindi gli stessi LSA

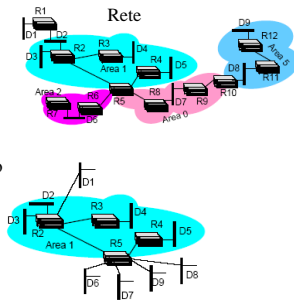
## Esempio

Rete completa



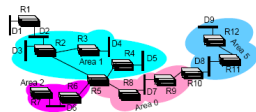
## Visione della rete da parte dei router in area 1

- R5= Area Border Router per l'area 1
- L'area è strutturata ad albero



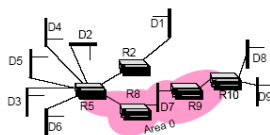
## Visione della rete da parte di un Backbone Router R5

Rete



- R5= Backbone router

- Tutte le destinazioni delle aree 1 e 2 sono viste raggiungibili mediante R5
- Le destinazioni dell'area 5 sono collegate a R10



## OSPF

- Ogni router OSPF viene identificato univocamente in tutto l'AS da una sequenza di 32 bit, chiamata **router ID** ( ad esempio l'indirizzo associato a una delle interfacce del router, generalmente quella con indirizzo minore)
- Tutti i router inviano periodicamente su tutte le loro interfacce pacchetti chiamati **hello packet** che include l'identità del router.
- Lo scambio di hello packet consente ai router adiacenti di conoscere l'uno l'identità dell'altro.
- La trasmissione sulle LAN avviene mediante un indirizzo multicast riservato
- Le informazioni topologiche sono inviate periodicamente mediante **LSA (Link State Advertisement)**, che svolgono le funzioni dei link state packet dell'algoritmo link state.
- Un pacchetto in OSPF può trasmettere varie LSA, contrariamente a quanto accade nel link state.

---

---

---

---

---

---

---

---

## OSPF: Link state

- I protocolli "Link State" sono adatti a reti di grandi dimensioni
- Principi base:
  - i router hanno la responsabilità di contattare i router vicini e acquisire la loro identità (pacchetti *Hello*)
  - i router emettono i *link state packets (LSP)* che contengono la lista delle reti connesse al router (vicini) ed i loro costi associati
  - gli LSP sono trasmessi a tutti gli altri router (*flooding*)
  - tutti i router hanno lo stesso insieme di dati e quindi possono costruire la stessa mappa della rete (database topologico)
  - le mappe di rete sono utilizzate per determinare i cammini migliori e quindi l'instradamento

---

---

---

---

---

---

---

---

## OSPF: Link state

- Gli LSP sono emessi
  - quando un router contatta un nuovo router vicino
  - quando un link si guasta
  - quando il costo di un link varia
  - periodicamente ogni fissato intervallo di tempo
- La rete trasporta gli LSP mediante la tecnica del *flooding*
  - un LSP è rilanciato da un router su tutte le sue interfacce tranne quella da cui è stato ricevuto
  - gli LSP trasportano dei riferimenti temporali (time stamp) o numeri di sequenza per evitare il rilancio di pacchetti già rilanciati consentendo un corretto riscontro dal ricevente

---

---

---

---

---

---

---

---

## Tecnica del flooding

- Assicura che tutti i router di una rete
  - riescano a costruire un database contenente lo stato della rete
  - abbiano le stesse informazioni sullo stato dei link
- Alla ricezione di un LSP:
  - un router esamina i campi di un LSP: link identifier, metrica, time stamp o numero di sequenza
  - se il dato non è contenuto nel database, viene memorizzato e l'LSP è rilanciato su tutte le interfacce del router tranne quella di ricezione
  - se il dato ricevuto è più recente di quello contenuto nel database, il suo valore è memorizzato e l'LSP è rilanciato su tutte le interfacce del router tranne quella di ricezione
  - se il dato ricevuto è più vecchio di quello contenuto nel database, viene rilanciato un LSP con il valore contenuto nel database esclusivamente sull'interfaccia di arrivo dell'LSP
  - se i due dati sono della stessa età non viene eseguita alcuna operazione

---

---

---

---

---

---

---

---

## Topologia della rete

- Ogni router mantiene un database che riflette i dati aggiornati sulla topologia della rete (Link State Database o Database topologico)
- La topologia di rete è rappresentata come un grafo orientato
- I nodi rappresentano
  - router
  - Network
    - reti di transito: non contengono host (sorgenti e/o destinazioni)
    - reti non di transito
- I rami rappresentano
  - collegamenti diretti tra nodi di tipo router
  - collegamenti tra nodi di tipo router e nodi di tipo network

---

---

---

---

---

---

---

---

## Costruzione del grafo

- Nel grafo rappresentativo della rete:
  - due router collegati da una linea punto-punto sono connessi da due rami orientati (uno per ogni verso)
  - più router connessi alla stessa rete sono rappresentati da nodi (router node) connessi ciascuno da due rami orientati (uno per ogni verso) al nodo rappresentativo della rete (network node)
  - una rete connessa ad un singolo router è rappresentata da un nodo di tipo stub
  - un host direttamente connesso ad un router è rappresentato con un ramo
  - sistemi autonomi esterni sono rappresentati da nodi stub ed il costo dei rami è determinato dal protocollo EGP

---

---

---

---

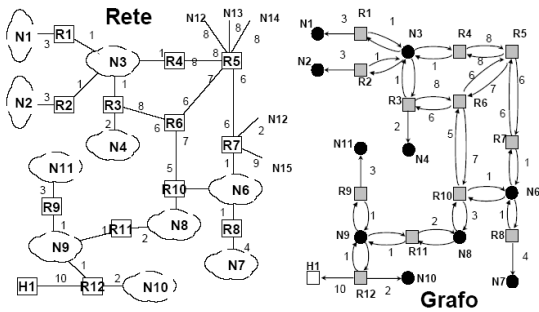
---

---

---

---

## Esempio

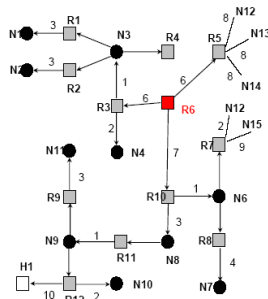


## Spanning tree e tabella di routing

- Ogni router calcola lo spanning tree a partire dal grafo rappresentativo della rete mediante l'algoritmo di Dijkstra

Spanning Tree e Routing Table in R6

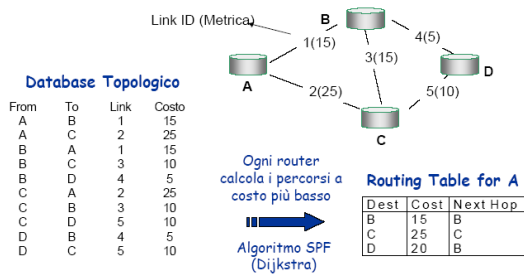
Destin.	N. H.	Dist.	Destin.	N. H.	Dist.
N1	R3	10	N10	R10	13
N2	R3	10	N11	R10	14
N3	R3	7	H1	R10	21
N4	R3	8	R5	R5	6
R1	R3	7	R7	R10	8
N6	R10	8	N12	R10	10
N7	R10	12	N13	R5	14
N8	R10	10	N14	R5	14
N9	R10	11	N15	R10	17



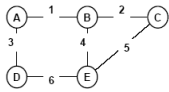
## Costruzione tabelle di routing

- Nodi vicini si riconoscono attraverso messaggi di "Hello"
- Una volta riconosciutisi, instaurano rapporti di adiacenza
- Nodi adiacenti si scambiano le informazioni sulla topologia dell'intera rete in loro possesso
- A regime tutti i nodi hanno una visione completa (ed uguale) della topologia di tutta la rete (*Database Topologico*)
- A partire dal database topologico ogni router costruisce la propria tabella di routing

## Costruzione tabelle di routing



## Esempio



Database Topologico (link bidirezionali)

Da	A	link	Dist
A	B	1	1
A	D	3	1
B	C	2	1
B	E	4	1
D	E	6	1
E	C	5	1

### Routing Table

<b>A</b>	Destinazione	A	B	C	D	E
	Distanza	0	1	2	1	2
	Link	local	1	1	3	1

<b>B</b>	Destinazione	A	B	C	D	E
	Distanza	1	0	1	2	1
	Link	1	local	2	1	4

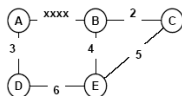
<b>C</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	0	2	1
	Link	2	2	local	5	5

<b>D</b>	Destinazione	A	B	C	D	E
	Distanza	1	2	2	0	1
	Link	3	3	6	local	6

<b>E</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	1	1	0
	Link	4	4	5	6	local

## Esempio

- Guasto del ramo AB



### Routing Table

<b>A</b>	Destinazione	A	B	C	D	E
	Distanza	0	inf	inf	1	inf
	Link	local	1	1	3	1

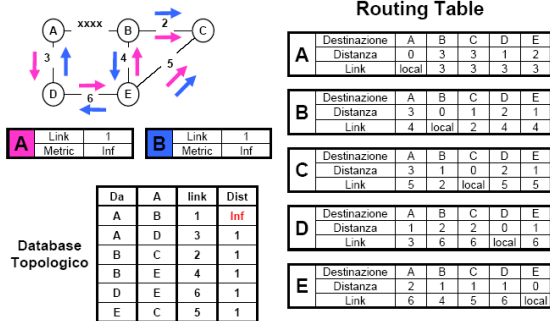
<b>B</b>	Destinazione	A	B	C	D	E
	Distanza	inf	0	1	inf	1
	Link	1	local	2	1	4

<b>C</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	0	2	1
	Link	2	2	local	5	5

<b>D</b>	Destinazione	A	B	C	D	E
	Distanza	1	2	2	0	1
	Link	3	3	6	local	6

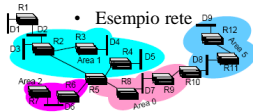
<b>E</b>	Destinazione	A	B	C	D	E
	Distanza	2	1	1	1	0
	Link	4	4	5	6	local

## Esempio



## Inoltro dei pacchetti

- Esempio: un calcolatore sulla LIS D3 invia un pacchetto a un calcolatore sulla LIS D9
- La prima decisione di routing viene presa dal mittente. Poiché il destinatario si trova su una LIS diversa invia al pacchetto al suo router di default.
  - Nell'esempio: router R2
- R2 ha una propria tabella di routing costruita in base ad informazioni connetture negli LSA:
  - Nell'esempio: ha scelto come next hop verso D9 il router R5, per cui invia il pacchetto a R5.
- R5 è un area border router e quindi è parte della backbone area da cui ha ricevuto le informazioni per costruire la rete.
  - Nell'esempio: poiché dalla mappa della rete D9 si trova a valle di D10 e poiché in base alla mappa il next hop verso D9 è R8, R5 inoltra il pacchetto a R8.
- I pacchetti sono inoltrati nell'area 0 verso il router R10.
- L'area border router R10, essendo parte dell'area 5, conosce i dettagli della topologia di tale area.
  - Nell'esempio: R10 sa che D9 è collegato a R12 e sceglie come next hop R11
- Il pacchetto viene trasportato nell'area 5 fino a R12, che lo consegna al destinatario tramite i servizi del livello 2 (data link).



## Comunicazioni tra router OSPF

- Gli LSA che OSPF fa scambiare tra i router sono inviati all'interno di pacchetti IP.
- Il protocollo OSPF è in realtà composto da tre "sottoprotocolli" con un header comune:
  - Protocollo HELLO (verifica dei link attivi).
  - Protocollo EXCHANGE (inizializzazione del DB)
  - Protocollo FLOODING (Aggiornamento dei DB)
- Per verificare se due router hanno le stesse informazioni (sono sincronizzati) si confrontano:
  - Il numero di LSA
  - La somma dei checksum

## Link state packet

- Sono pacchetti IP contenenti
  - Protocol number 89
  - TTL impostato a 1 (sono sempre “intercettati” dal protocollo OSPF del router direttamente collegato)
- Tre possibili indirizzi di destinazione (si usa il multicast se presente):
  - ALLDRouters (tutti i router “designati”)
  - ALLOSPFRouter (tutti i router OSPF)
  - IP del vicino (caso line punto-punto)
- La frammentazione è eseguita solo se indispensabile; per evitare la frammentazione si limita la dimensione a 1500 byte (per gli LSA).

---

---

---

---

---

---

---

---

## Formato messaggi OSPF

- Il protocollo OSPF e' implementato direttamente sopra il protocollo IP e consiste di tre sottoprotocolli: hello, exchange e flooding.
- Tutti i messaggi OSPF hanno una testata comune.

Testata Comune dei Protocolli OSPF

Versione	Tipo	Lunghezza Pacchetto
ID Router		
ID Area		
Checksum	Tipo Autenticazione	
Autenticazione		
Autenticazione		

- Versione: 2
- Tipo: tipo di messaggio
- Lunghezza: lunghezza in byte del messaggio
- ID router: e' l'identificativo IP del router che invia il messaggio;
- l'Area ID e' l'identificativo di area, ove il valore 0 e' riservato per l'area backbone.
- Checksum e' quello normale di IP, calcolato su tutto il messaggio.
- Tipo di autenticazione: può avere soltanto due valori : 0 e 1.
  - 0 - nessuna autenticazione
  - 1 - autenticazione semplice, a password di 8 caratteri.
- Serve in realta' solo a proteggere da errori e imprevisti.

---

---

---

---

---

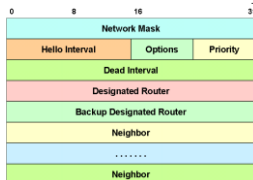
---

---

---

## Protocollo HELLO

- Questo protocollo ha due compiti:
  - verificare che i link siano operativi
  - eleggere il router designato ed il backup



- **Network Mask:** netmask associata all'interfaccia da cui viene emesso il pacchetto
- **Hello Interval:** comunica ogni quanti secondi viene emesso un pacchetto di Hello
- **Options:** vengono definiti solo gli ultimi 2 bit
  - **E:** se il router è in grado di inviare e ricevere route esterne; è pari a 0 se l'interfaccia appartiene ad una stub area
  - **T:** se il router è in grado di gestire il routing TOS
- **Priority:** serve per l'elezione del Designated Router e viene settato da management. Ciascun router e' configurato con una 'priorita', che puo' variare tra 0 e 255. Viene eletto DR il router che ha la priorità più alta; quindi un router con priorità 0 non potrà mai diventare DR.
- **Dead Interval:** intervallo di tempo di validità dei pacchetti di Hello ricevuti, oltrepassato il quale gli Hello ricevuti vengono ritenuti decaduti
- **DR, BDR:** indirizzo del Designated Router - BDR (0 se non sono ancora stati definiti)
- **Neighbor:** lista di router\_ID da cui è stato ricevuto il pacchetto di Hello negli ultimi Dead\_Interval secondi

---

---

---

---

---

---

---

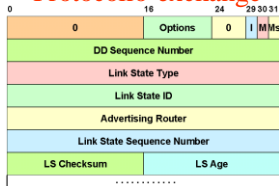
---



## Protocollo exchange

- Il protocollo Exchange e' asimmetrico; il primo step del protocollo consiste nel selezionare un "master" e uno "slave" e solo di seguito i due routers si scambieranno la descrizione dei propri database.
- Usato per la sincronizzazione iniziale dei database dei router, che avviene fra una coppia di router adiacenti.
- Scambio di messaggi "Database Description" in modalit  *master-slave* (chi invia per primo si elegge *master*, il ricevente   lo *slave*).
- Ogni messaggio   identificato da tre flag: I (*initialize*), M (*more*) e MS (*master/slave*) che regolano la procedura di sincronizzazione.
- Ogni messaggio include LS record del database del router, ed un *acknowledgement* dei pacchetti di EXCHANGE ricevuti nella direzione opposta

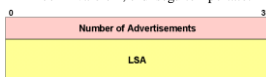
## Protocollo exchange



- Options: e' equivalente al pacchetto Hello
- E: se il router   in grado di inviare e ricevere route esterne;   pari a 0 se l'interfaccia appartiene ad una stub area
- T: se il router   in grado di gestire il routing TOS
- I: Initialize
- M: More
- MS: Master - Slave (1= Master)
- DD SN: numero di sequenza del pacchetto DD
- I campi successivi (che possono essere ripetuti) sono la descrizione dell'header di un LSA, ed hanno quindi lo stesso significato.

## Protocollo di flooding

- Il protocollo flooding viene utilizzato per diffondere (processo di forwarding) a tutta la rete il nuovo stato di un link.
- Questi aggiornamenti vengono inviati, attraverso il pacchetto di "Link State Update", nel caso di:
  - un cambiamento di stato del link
  - allo scadere di un timer (normalmente 60 min)
- Il pacchetto "Link State Update", che caratterizza il campo "Type" dell'header comune con il valore 4, e' di seguito riportato:



- Number of Advertisement:   il numero di LSA che vengono trasportati dal pacchetto in esame in quanto e' possibile trasportare pi  LSA
- LSA:   il Link State vero e proprio

## Confronti tra RIP e OSPF

- RIP è un protocollo di tipo **Distance Vector**
- Utilizza l'algoritmo di **Bellmann-Ford**
- Ogni router informa solamente i suoi vicini sulla propria tabella di routing
- I cambiamenti sul routing vengono propagati **periodicamente** causando un'occupazione di banda (circa ogni 30 sec.)
- **Lenta** convergenza
- RIP viene utilizzato in reti di **piccole dimensioni**

- OSPF è un protocollo di tipo **link state**
- Utilizza l'algoritmo di **Dijkstra**
- Ogni router conosce lo stato di tutta la rete
- I cambiamenti sul routing vengono propagati **istantaneamente** (nel momento in cui avvengono) attraverso la tecnica del *"flooding"*
- **Migliore convergenza** rispetto al RIP
- OSPF utilizza una **struttura gerarchica**
- OSPF si adatta bene a reti di **grandi dimensioni**

---

---

---

---

---

---

---

---

---

---

## EGP (Exterior Gateway Protocol)

- Primo protocollo tra AS
- risale ai primi anni ottanta (RFC 827)
- EGP è caratterizzato da tre funzionalità principali:
  - **neighbor acquisition**
    - verificare se esiste un accordo per diventare vicini
  - **neighbor reachability**
    - monitorare le connessioni con i vicini
  - **network reachability**
    - scambiare informazioni sulle reti raggiungibili da ciascun vicino
- EGP è simile ad un protocollo distance vector
  - le informazioni inviate ai vicini sono sostanzialmente informazioni di raggiungibilità
  - non sono specificate le regole per definire le distanze
  - la distanza minima può non essere il criterio migliore da seguire

---

---

---

---

---

---

---

---

---

---

## BGP ( Border Gateway Protocol)

- BGP è stato concepito come sostituto di EGP
- Oggi è in uso la versione 4 (RFC 1771)
  - consente l'uso del CIDR
- I router BGP appartenenti ad AS adiacenti si scambiano informazioni attraverso connessioni di livello affidabile TCP (porta 179)
  - le comunicazioni sono affidabili
  - funzionalità di controllo degli errori demandate allo strato di trasporto BGP più semplice
- BGP è un protocollo di tipo **path vector**
  - nel vettore dei percorsi si elencano tutti gli AS da attraversare per andare da una sorgente ad una destinazione
  - risolve il problema dei percorsi ciclici
  - più consono a definire le politiche di routing tra AS rispetto alla semplice distanza

---

---

---

---

---

---

---

---

---

---

## BGP

- Per il calcolo dei percorsi viene utilizzato il protocollo Path Vector ( che è una variante del distance vector)
- Mentre nel distance vector viene indicato solo il costo di un cammino, nel path vector per ogni destinazione IP è fornita la **sequenza di Autonomous System (AS)** da attraversare.
- Per la descrizione di un percorso verso una destinazione vengono utilizzati dei campi detti attributi.
- I più importanti sono la "**lista degli AS attraversati**" e la "**lista delle reti raggiungibili**"
- A questo scopo, BGP utilizza due ottetti nei quali una serie di opportuni flag permettono di discriminare il significato dei vari attributi.
- Supporta autenticazione

---

---

---

---

---

---

---

---

## BGP

0	1	2	3	4	8	15
O	T	P	E	0	Attribute Type Code	

- **O (well-known/optional)**: indica se l'attributo e' opzionale (O=1) oppure se e' ben conosciuto.
- **T (transitive/local)**:
- **transitive**: (T=1) è un parametro che può essere aggiornato o interpretato da altri router (ad esempio la banda del link più piccolo sul percorso) è deve quindi essere trasmesso ai router successivi
- **local**: parametro che assume un significato solo all'interno di un certo AS; non deve essere propagato
- **P (partial)**: settato se qualche router sul percorso non è stato in grado di comprendere il significato di quel particolare attributo (ad es uno opzionale); significa che l'attributo è stato solo parzialmente valutato
- **E (length)**: indica se la lunghezza del campo LENGTH è codificata con 1 ottetto (E = 0) oppure 2

---

---

---

---

---

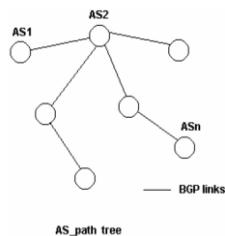
---

---

---

## BGP

- BGP utilizza i messaggi scambiati tra i border router per costruire un grafo di AS.
- In genere costruisce un albero: un **AS path tree**




---

---

---

---

---

---

---

---