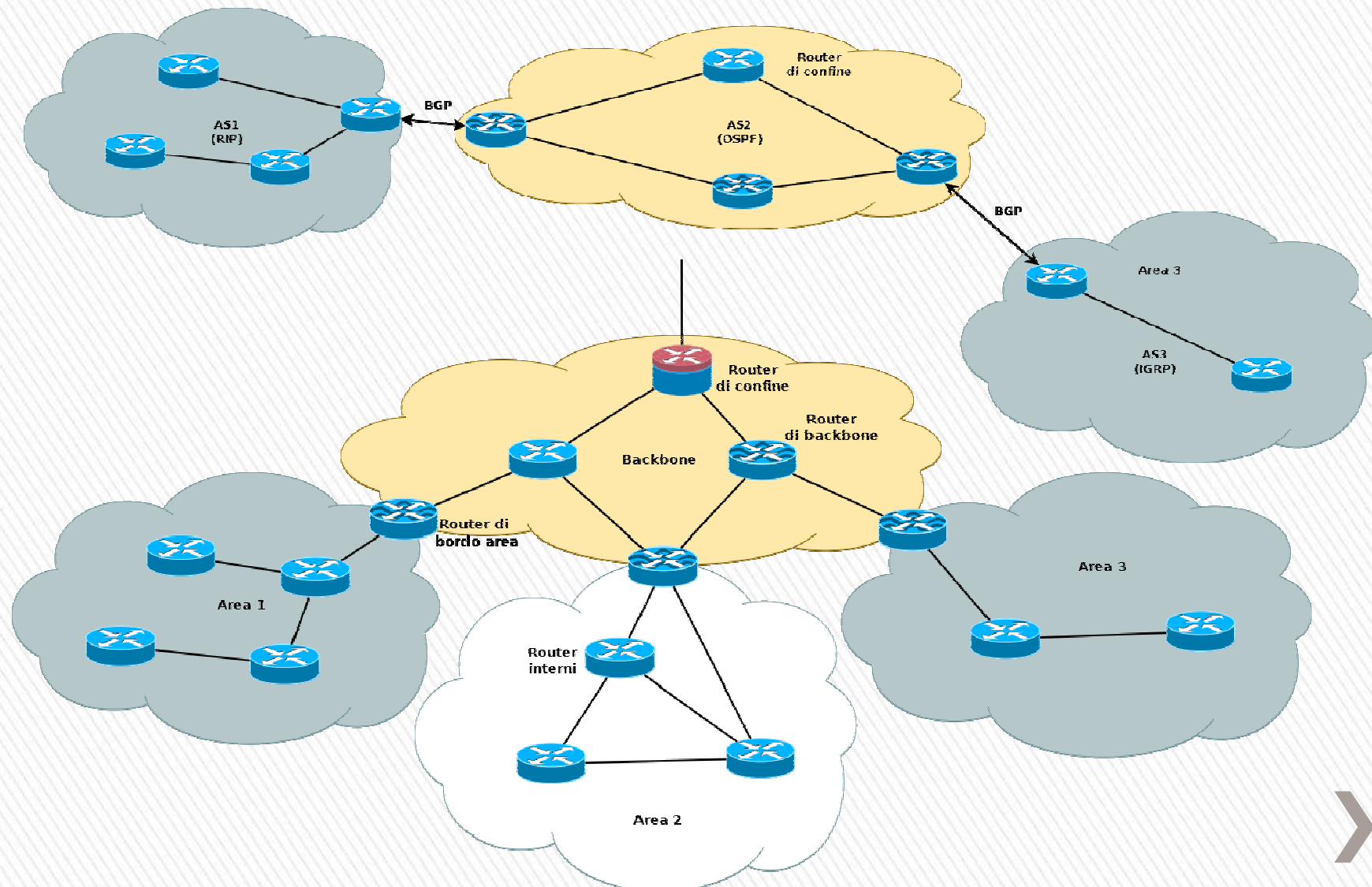


PROTOCOLLI DI ROUTING

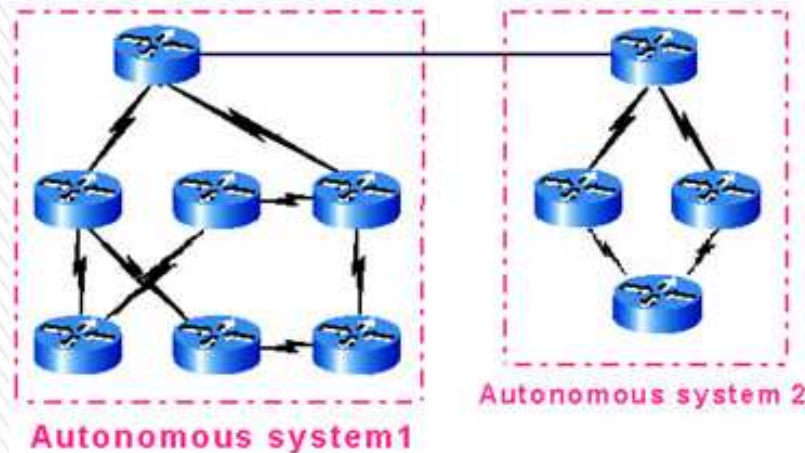


Un protocollo di routing è un processo di comunicazione tra i router per scambiarsi informazioni utilizzate per formare la **tabella di routing** (routing table).

INTERNETWORK

Per realizzare una internetwork, dobbiamo unire più reti insieme, ciascuna delle quali può essere gestita in modo indipendente rispetto alle altre.

Questo ha portato alla suddivisione dell'internetwork in aree diverse. Ogni zona che appartiene allo stesso dominio amministrativo prende il nome di **Autonomous System (AS)**.



L'instradamento attraverso le reti avviene sulla base di particolari tabelle dette **tabelle di routing** (caricate sul router). All'aumentare del numero dei Router, cresce la complessità di tali tabelle e delle procedure per la determinazione dei percorsi migliori. La scelta di suddividere l'insieme delle reti in **AS**, ha permesso di semplificare le procedure abbinate al processo di routing.

Tutti i router del medesimo **AS** condividono lo stesso protocollo di routing. Ricordiamo che lo scopo di un **protocollo di routing** (Routing Protocol), è quello di mantenere dinamicamente le routing table.

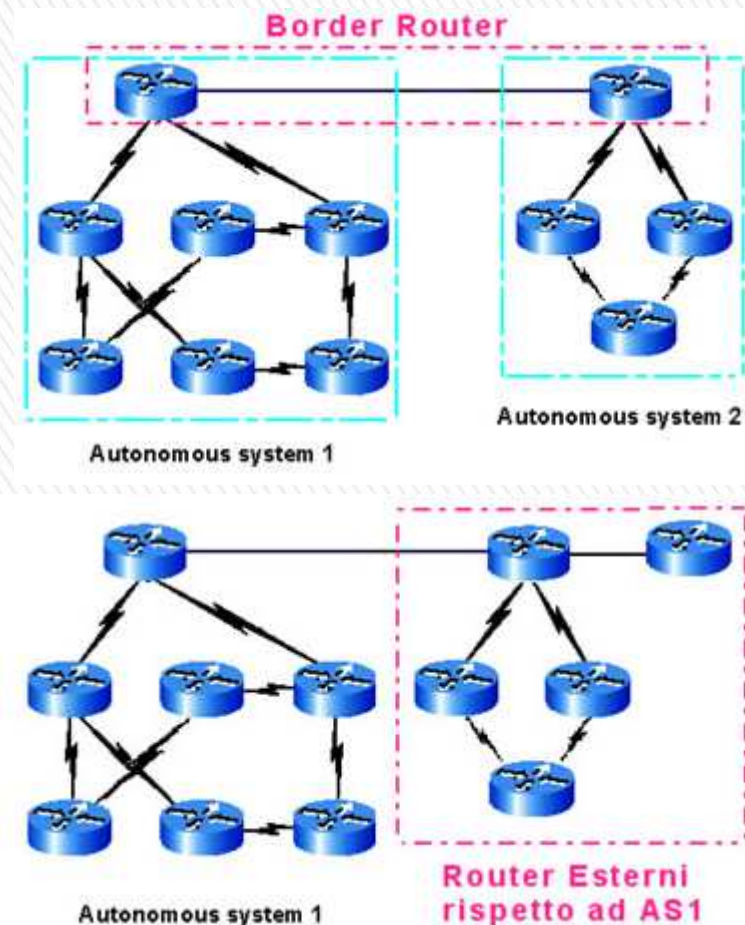
AUTONOMOUS SYSTEM

Si rende a questo punto importante **collegare** i vari **Autonomous System** fra loro mediante l'uso di router in grado di instradare pacchetti fra aree diverse. Possiamo suddividere i router nelle 3 seguenti categorie:

Interior Router: responsabili della trasmissione informazioni all'interno di un **AS**, per cui non hanno una diretta connessione con una qualsiasi rete esterna.

Border Router: (Router di confine) realizza la connessione fra AS diversi. Può essere considerato quindi come il punto di ingresso e di uscita verso altri **AS**.

Exterior Router (Router esterno) appartengono a questa classe tutti i router che sono esterni rispetto all'**AS** che stiamo considerando.

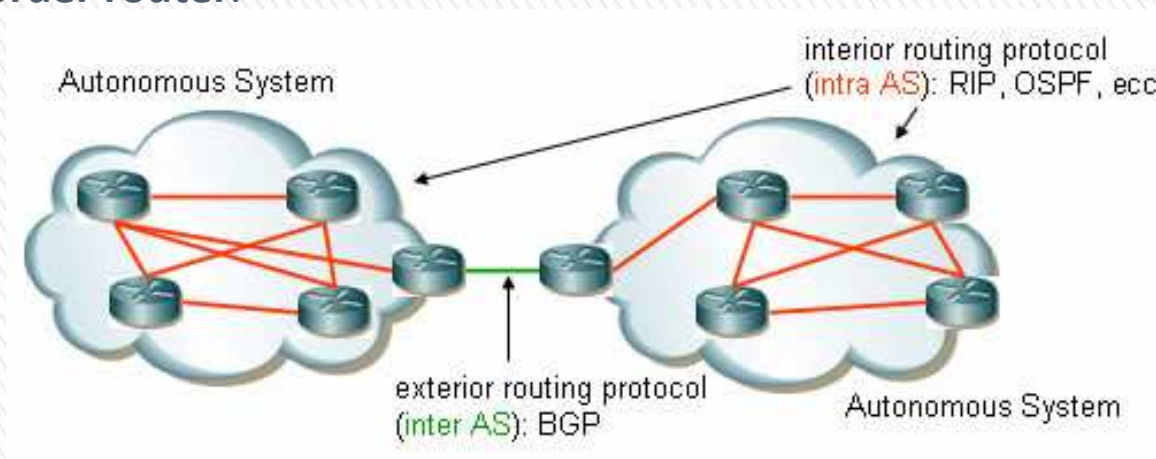


Le reti sono quindi raggruppate in **Autonomous System** (gruppi di reti controllati e gestiti da un'unica entità amministrativa). Gli **Autonomous System** sono identificati tramite un numero intero, univoco a livello mondiale, assegnato dalla stessa autorità che rilascia gli indirizzi Internet (**IANA** → <http://www.iana.org/>).

Gli **interior router** scambiano le informazioni di instradamento tramite l' **Interior Gateway Protocol (IGP)** mentre i **border router** utilizzano un **Exterior Gateway Protocol (EGP)**.

Nel caso di routing interno all'AS (**intradomain routing**) i router interni devono conoscere solo i segmenti di rete appartenenti allo stesso AS. Non è quindi necessario che mantengano informazioni su reti esterne. Qualora servisse comunicare con altri **AS** si limitano a mandare i pacchetti relativi ai router di confine.

Questa organizzazione ha dei notevoli vantaggi, poiché riduce le dimensioni della routing table in quanto le network esterne non sono presenti direttamente in tale tabella se non attraverso i **border router**.



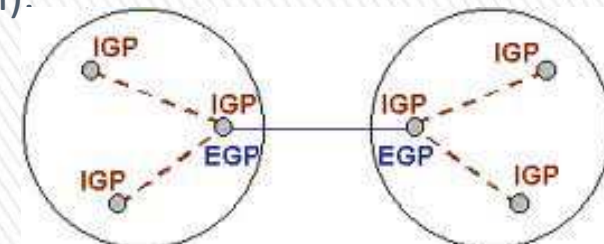
PRINCIPALI PROTOCOLLI DI ROUTING

Protocolli IGP:

- **RIP:** E' un protocollo utilizzato all'interno di un **AS** per l'instradamento dinamico. **RIP** e' l'acronimo di **Routing Information Protocol** ed e' un protocollo relativamente semplice appartenente alla famiglia di protocolli di tipo "**distance vector**".
- **OSPF:** è l'acronimo di **Open Shortest Path First**. **OSPF** appartiene alla classe **IGP** e pertanto opera internamente a ciascun **Autonomous System (AS)**. Il protocollo, essendo di tipo "**link state**", utilizza l'**algoritmo SPF** per calcolare il percorso più breve verso la destinazione. L'**algoritmo SPF** si basa principalmente sulla larghezza di banda; il percorso a costo più basso viene inserito nella **routing table** (o **forwarding database**). **OSPF** è protocollo pubblico (quindi non proprietario).

L'**OSPF** raccoglie le informazioni dai router vicini circa lo stato dei link ed invia i suoi dati agli altri. Questo scambio di informazioni permette ai router di creare la **topology table** o **link state database**.

L'**OSPF** è un protocollo da utilizzare in reti estese e scalabili (ampliabili). Vengono usati in alternativa ai protocolli "distance vector" quando i limiti di questi ultimi li rendono inapplicabili. Ad esempio se voglio utilizzare metriche basate sulla larghezza di banda o quando voglio, in reti estese, avere una maggior convergenza del protocollo (con reti ampie il RIP può avere una convergenza di alcuni minuti).



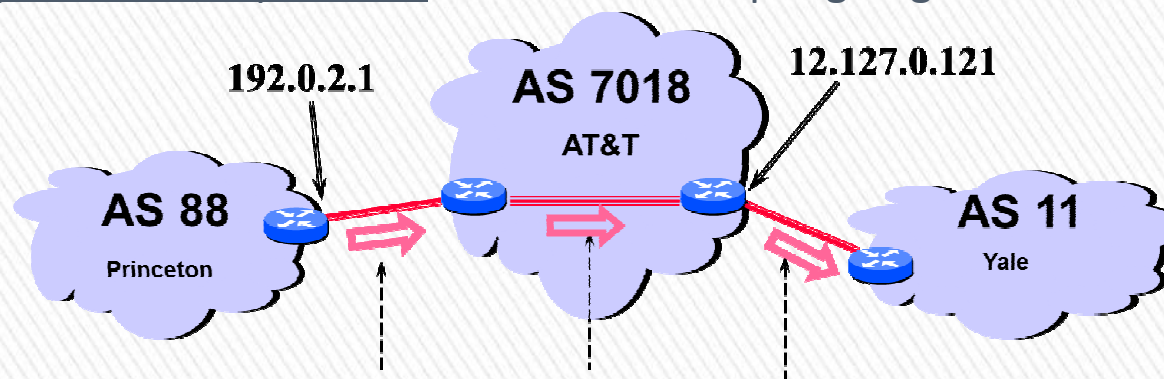
Protocolli EGP:

- **BGP:** Il **Border Gateway Protocol** è un protocollo di instradamento che agisce nel 'cuore' della rete Internet. Il **BGP** usa tabelle di reti IP, o prefissi, che forniscono informazioni sulla raggiungibilità dei diversi AS. Si tratta di un protocollo di routing a indicazione di percorso ("**path vector**"), che non usa metriche di carattere tecnico ma prende le decisioni di instradamento basandosi su regole definite in ciascun rete. *La versione corrente, BGP-4, è definita nella specifica RFC 4271.*

Tutti i **provider Internet (ISP)** sono obbligati a usarlo. **BGP** è quindi il protocollo di routing usato nell'internet pubblica ed è l'unico (visto che si è costretti ad usarlo!) di tipo **exterior**.

I protocolli **EGP**, a differenza degli **IGP**, non hanno metodi per scoprire l'esistenza di router "vicini" ma devono essere definiti manualmente.

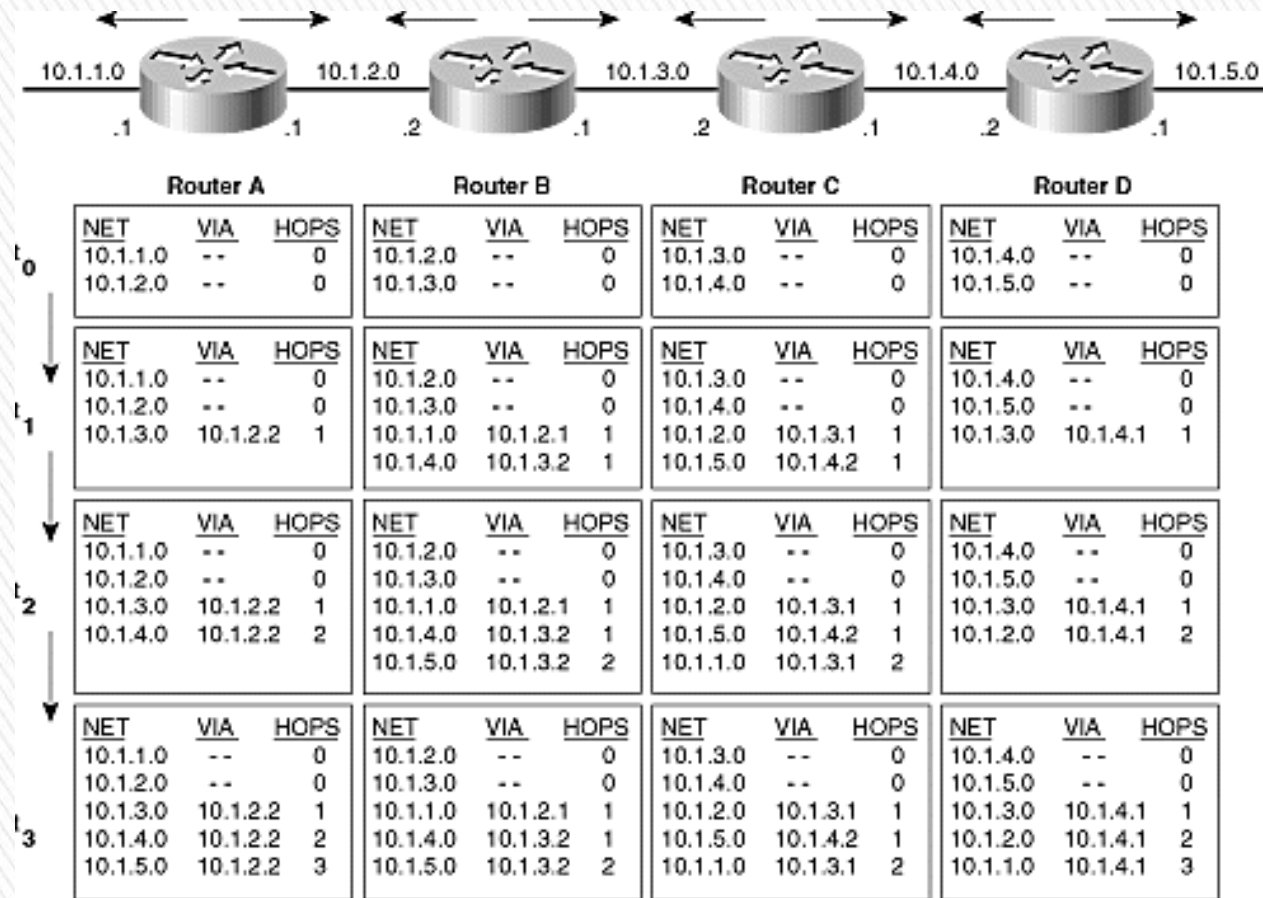
Il **BGP** è un protocollo d'instradamento decentralizzato (comunica solo con i router a cui è direttamente collegato). Essendo un algoritmo di tipo "**path vector**" non viene fornito il numero di hop ma l'intero percorso da attraversare per giungere ad una determinata destinazione.



PROTOCOLLI DI ROUTING: CLASSIFICAZIONE PER TIPI DI ALGORITMO:

DISTANCE VECTOR:

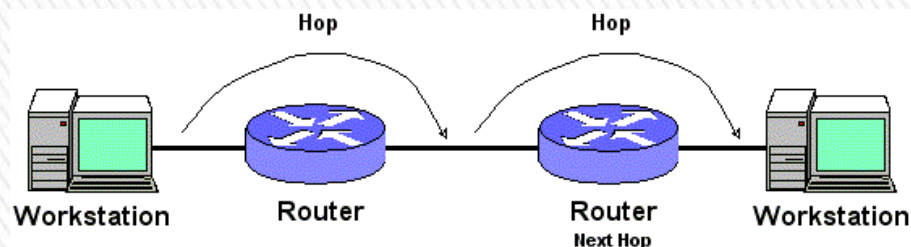
I protocolli “**distance vector**” si basano sul fatto che ciascun router tiene traccia della distanza e della direzione (vector) di tutte le possibili destinazioni nell'ambito dell'internetwork.



Calcolo della metrica in un protocollo di tipo "DISTANCE VECTOR":

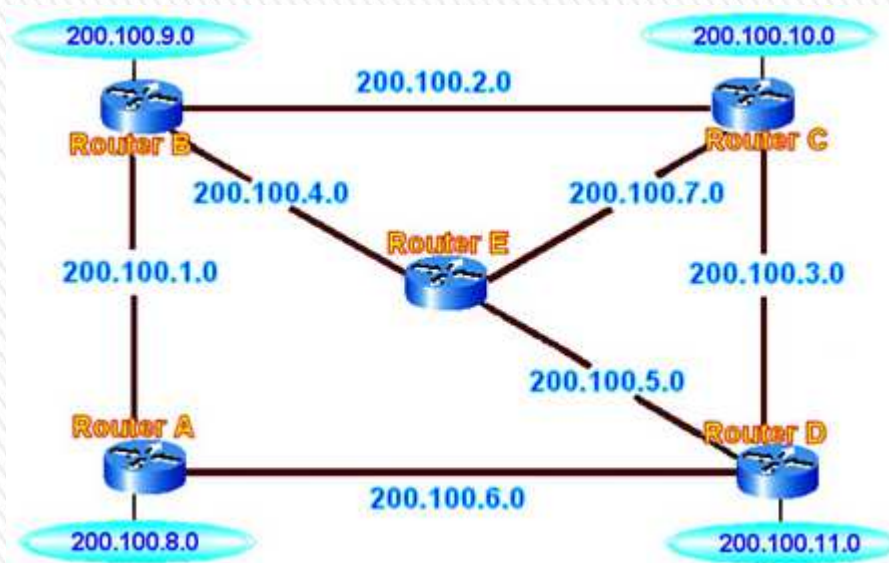
Soffermiamoci ora ad analizzare, rimanendo nell'ambito di un protocollo **distance vector**, in quale maniera si possono calcolare le distanze riportate nelle tabelle di routing. Attribuiamo al risultato di tale calcolo il nome di **metrica**. Quest'ultima può essere valutata considerando una singola caratteristica del percorso o più caratteristiche combinate insieme. Esaminiamone alcune nel dettaglio:

- **Ritardo.** Si riferisce al tempo che viene impiegato affinché un pacchetto venga inviato lungo una connessione. Questo parametro può dipendere, sia dalla velocità trasmissiva del mezzo fisico utilizzato, sia dalla congestione presente sul collegamento.
- **Banda.** Maggiore è la banda assegnata ad un certo collegamento, minore sarà il valore attribuito alla distanza.
- **Hop Count.** Questa variabile considera il numero di router (salti) che un pacchetto deve *attraversare* per arrivare alla destinazione desiderata.
- **Carico.** Si riferisce all'utilizzo delle risorse hardware di una rete, infatti se un router non è in grado di processare in maniera corretta le informazioni a causa di sue limitate capacità di calcolo, verrà scelta una rotta alternativa.



Esempio di funzionamento di un protocollo di routing di tipo "DISTANCE VECTOR":

Facciamo l'assunzione di considerare in questo esempio una metrica di tipo **hop count** e che tutti i router della rete sottostante vengano inizializzati nello stesso istante. In un primo momento ciascun router costruirà una routing table contenente solo la rete locale di appartenenza più quelle direttamente connesse. Ad esempio il **router A** avrà la seguente tabella:



NETWORK	NEXT HOP	HOP COUNT
200.100.8.0		0
200.100.6.0		0
200.100.1.0		0

Si osservi che essendo le reti elencate in tabella direttamente connesse al **Router A**, l'**hop count** assegnato sarà zero e il campo **next hop** vuoto, ad indicare che non sono necessari passaggi intermedi attraverso altri router per raggiungerle.

Ciascuna tabella di routing verrà successivamente inviata a tutti i router adiacenti. Prendendo in esame il **Router A**, dopo aver ricevuto l'update dagli altri Router, aggiornerà la propria tabella aggiungendovi le reti limitrofe (vedi indicazione sottostante).

Tabella Router A alla 1° iterazione

NETWORK	NEXT HOP	HOP COUNT
200.100.8.0		0
200.100.6.0		0
200.100.1.0		0
200.100.11.0	Router D	1
200.100.5.0	Router D	1
200.100.4.0	Router B	1
200.100.2.0	Router B	1
200.100.3.0	Router D	1
200.100.9.0	Router B	1

Questo procedimento, precedentemente accennato, viene iterato fra tutti i Router adiacenti ad intervalli di 60 secondi. Ad ogni aggiornamento vengono aggiunte le reti che non sono ancora presenti in tabella (*sconosciute!*) incrementando di 1 l'**hop count**. Se invece la rete risulta già presente, viene scelta quella con metrica minore.

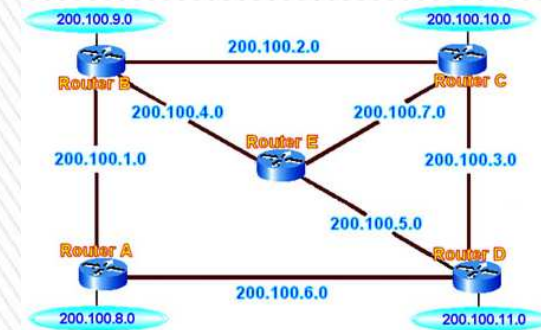


Tabella Router C alla 2° iterazione

NETWORK	NEXT HOP	HOP COUNT
200.100.2.0		0
200.100.7.0		0
200.100.3.0		0
200.100.10.0		0
200.100.8.0	Router B	2
200.100.1.0	Router B	1
200.100.9.0	Router B	1
200.100.4.0	Router B	1
200.100.2.0	Router B	1

Questo protocollo non permette di avere una conoscenza della rete nella sua globalità, ma soltanto dei salti successivi (**next hop**) che permetteranno di inoltrare i pacchetti verso la destinazione finale.

Questa caratteristica contraddistingue i protocolli di tipo **"Distance Vector"** da quelli di tipo **"link state"** i quali invece riescono a costruirsi delle tabelle di routing che tengono conto della topologia dell'intera rete.

Il routing loop nel “DISTANCE VECTOR”:

Nel protocollo del tipo **distance vector** ogni router ha una conoscenza solo locale della rete, cioè sa che per raggiungere una certa destinazione deve passare attraverso un particolare router, ma non sa cosa succede dopo quel router.

Le variazioni della topologia possono portare a dei problemi di instabilità come il “**routing loop**” che sono il classico esempio di situazione instabile. Un **loop** si verifica ad esempio in questa situazione:

Si supponga avvenga un'interruzione sul **link L2** tra il **router B** e **C**:



- Il **router B** setta tutti le entry (dette anche **DV** o Distance Vector) relative alle destinazioni raggiungibili sulla **linea L2** ad un costo infinito (quindi irraggiungibili);

Tabella di instradamento di B dopo la caduta del link		
B → ?	Link	Cost=hops
B	local	0
A	1	1
C	2	inf.

- **Caso A:** il **router B** trasmette il nuovo Distance Vector (**DV**) al **router A**. Quest'ultimo aggiorna la propria routing table e la rete si stabilizza;

Distance Vector di A		
A → ?	Link	Cost=hops
A	local	0
B	1	1
C	1	inf.

➤ **Caso B:**

1. il **router A** trasmette la sua tabella (nella tabella di **A** il costo attuale per raggiungere **C** è a 2) prima che B abbia propagato l'errore. Nel **DV** di **B** il costo per raggiungere **C** (però via A!) verrà aggiornato a **3**.

Distance Vector proveniente da A

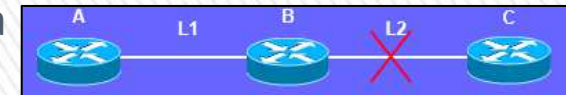
A → ?	Link	Cost=hops
A	local	0
B	1	1
C	1	2



Tabella di instradamento di B dopo l'aggiornamento

B → ?	Link	Cost=hops
B	local	0
A	1	1
C	1	3

2. Il **router B** annuncia ad A che C è raggiungibile (mediante A!) con costo 3. Il **router A** aggiorna quindi la sua tabella, ponendo **C** raggiungibile mediante **B** con costo **4**. Con questa impostazione un pacchetto dati per **C** verrebbe 'intrappolato' tra **A** e **B**.



Distance Vector proveniente da B

B → ?	Link	Cost=hops
B	local	0
A	1	1
C	1	3



Tabella di instradamento di A dopo l'aggiornamento

A → ?	Link	Cost=hops
A	local	0
B	1	1
C	1	4

Questo transitorio continuerebbe all'infinito (**count to infinity**) se non vi fosse un valore massimo che pone fine a questa situazione dichiarando una destinazione irraggiungibile se il numero di **hop** supera questo valore massimo.

Nei protocolli del tipo **distance vector**, la causa dei loop è strettamente legata alla visione limitata che i router hanno della rete.

L'unico metodo per impedire che si creino i loop e che un pacchetto rimanga imprigionato è quella di porre un limite al **hop-count**. Tutte le destinazioni che distano più di **hop-count-limit** sono considerate irraggiungibili. Ciò permette al protocollo di convergere ad una situazione stabile.

Il valore di **hop-count-limit** in RIP è 15; quindi ogni destinazione lontana 16 salti è considerata irraggiungibile. Questo pone un limite alla dimensione massima della rete. In particolare, ipotizzando una distribuzione abbastanza uniforme dei nodi della rete, si può usare il RIP per reti al massimo di 1000 nodi.

Caratteristiche dei protocolli di tipo "DISTANCE VECTOR":

Vantaggi:

- semplice da implementare;
- non richiede grandi capacità elaborative e memoria occupata sul router.

Svantaggi:

- possono innescarsi dei loop a causa di particolari variazioni della topologia;
- Lenta convergenza (arrivo ad uno stato stabile);
- vedono la rete dalla prospettiva dei vicini. E' quindi difficile capirne e prevederne il comportamento su reti grandi. Nessun nodo ha una mappa della rete;
- Metrica basata sui salti e non sulla banda;
- Il **hop-count-limit** limita la dimensione della rete: il numero di hop ammesso dovrà essere inferiore all'hop-count-limit.
- Inviano l'intera tabella di routing per cui ho un alto consumo di banda

LINK STATE:

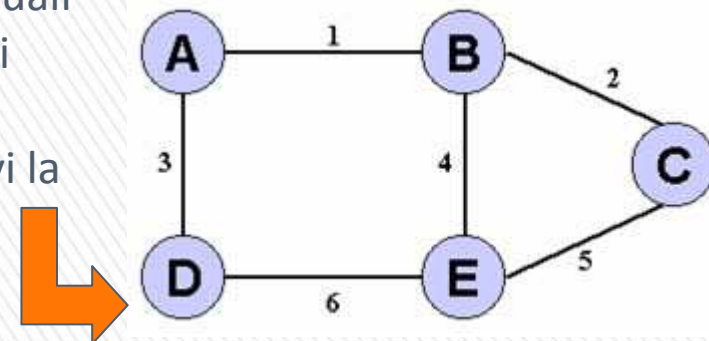
I protocolli **link state** hanno un approccio diverso rispetto ai distance vector per raccogliere le informazioni necessarie ad effettuare il routing dinamico e per calcolare i percorsi di destinazione.

Una caratteristica importante di questi protocolli è quella di mantenere una visione completa della topologia di rete, cioè ciascun router è a conoscenza dell'esistenza di qualsiasi altro router della rete e di come sono connessi fra loro.

I protocolli di tipo **Link State** sono infatti basati sul concetto di "mappa distribuita": tutti i nodi posseggono una copia della mappa della rete, che viene regolarmente aggiornata.

Il protocollo **link state** mantiene un database complesso per immagazzinare le informazioni sulla topologia, le quali vengono usate per calcolare i percorsi verso le possibili destinazioni. In questo database ciascun record rappresenta un link nella rete. Come esempio si osservi la rete a lato e la tabella di instradamento associata.

Da	A	Link	Distanza
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1



Ciascuna entry contiene un identificatore di interfaccia, la destinazione, il numero di link e la distanza o metrica.

Da	A	Link	Distanza
A	B	1	1
A	D	3	1

Con queste informazioni ogni nodo può facilmente calcolare il percorso più breve verso tutti gli altri nodi. Poiché tutti i nodi contengono lo stesso database ed eseguono lo stesso algoritmo di route-generation, i percorsi sono coerenti e non si verificano loops.

Per la determinazione del percorso ciascun router utilizza un algoritmo del tipo SPF (shortest path first) ovvero viene scelto quello con il percorso più breve fra tutti i possibili per una determinata destinazione. L'algoritmo utilizzato è l'algoritmo di Dijkstra.

I pacchetti, inviati dai router, che consentono la costruzione della mappa della rete, sono detti **Link State Packet (LSP)**.

Vediamo nel dettaglio come opera il protocollo di tipo Link state:

- Nella fase di startup il router rileva i segmenti che sono ad esso connessi direttamente.
- Ciascun router invia delle informazioni (LSA) per far conoscere agli altri router quali sono le proprie connessioni. Questi LSA vengono scambiati fra tutti i router dell'internetnetwork in pacchetti **LSP**.
- Ciascun router elabora l'LSA proveniente dagli altri e si costruisce il proprio database.

Caratteristiche dei protocolli di tipo “LINK STATE”:

Vantaggi:

- Le informazioni vengono inviate in maniera non periodica ma al variare della topologia (***triggered updates***) quindi pochi aggiornamenti.
- Usa lo Shortest Path per la determinazione del cammino minimo;
- visione totale della rete (registrata nelle “topology table”);
- convergenza veloce;
- non suscettibili di routing loops;
- consumano poca banda.

Svantaggi:

- Complessi da configurare;
- Richiede molta memoria e grandi capacità elaborative
- richiedono una struttura gerarchica che suddivida in aree più piccole e gestibili la rete riducendo la dimensione della “**topology table**”;

PATH VECTOR:

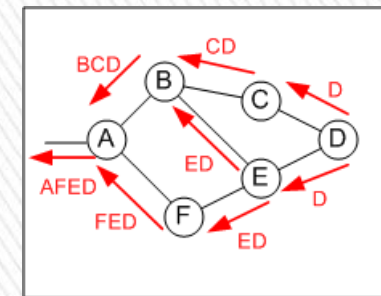
Il protocollo **PATH VECTOR** è una variante del Distance Vector che permette la memorizzazione dell'intero percorso tra due AS con immediato riconoscimento dei loop e del conteggio a infinito (**count to infinity**). Inoltre non obbliga tutti gli **AS** ad usare la stessa metrica.

Nel **path vector** i nodi si scambiano informazioni sui percorsi anziché sui costi.

- il vettore che ogni router manda ai vicini contiene l'intero cammino che il pacchetto deve seguire
- il router ignora tutti i cammini dove compare lui stesso (evitando così i loop)
- richiede un numero maggiore di informazioni da scambiare
- Ogni nodo applica le proprie regole

Ogni entry nella routing table contiene: la rete di destinazione, il prossimo router e il percorso per arrivare a destinazione

Esempio: Vediamo un esempio di applicazione dell'algoritmo "**Path vector**". Obiettivo è trovare il percorso migliore per raggiungere l'AS **D**. Ogni AS segnala il suo percorso preferenziale per raggiungere **D** (le frecce indicano il percorso della segnalazione). Ad esempio quando il nodo **B** riceve le due segnalazioni **CD** ed **ED** seleziona tra queste quella che reputa migliore (ad esempio **CD**) e informa i router successivi sul suo percorso preferenziale verso **D** (ovvero **BCD**). La selezione del percorso avviene sulla base di regole locali definite su ogni AS (esempio evitare il nodo E perchè lento ed inaffidabile, dare precedenza ai percorsi più brevi, etc.).



LINK UTILI:

<http://www.informaticaetlc.it/blog/tag/distance-vector>

http://www.dacrema.com/Informatica/Autonomous_System.htm

<http://deisnet.deis.unibo.it/Didattica/CorsiCE/RetiLB/>

