Università di Genova
Facoltà di Ingegneria

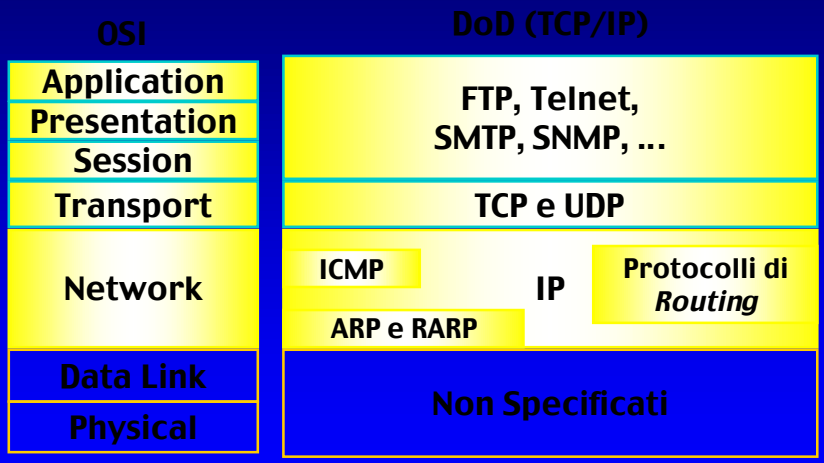
Telematica

6. TCP/IP - Introduzione ed IP

Prof. Raffaele Bolla

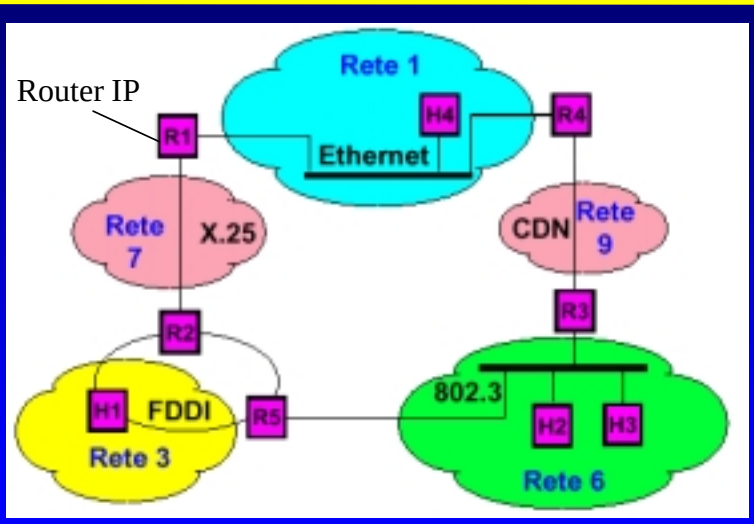
dist

Architettura



6.2

Struttura



6.3

Internetworking Protocol (IP)

- È il livello Network di TCP/IP ed è il protocollo principale di questa architettura
- Offre un servizio non connesso.
- Semplice protocollo di tipo Datagram.
- E` specificato nel RFC (Request For Comments) 791.
- La versione attuale è la 4 (IPv4), anche se quella successiva è già stata completamente definita come 6 (IPv6).

6.4

IP - Datagram

- I pacchetti viaggiano su percorsi indipendenti
- *Out of order delivery*
- Gestione della banda difficoltoso
  - riservare e garantire banda
  - rifiutare connessioni (*Call Acceptance Control*)
- Meno complesso: non richiede negoziazione né lato utente, nè all'interno della rete
- Robusto: si adatta a variazioni di traffico,topologia, guasti
- Adatto al traffico dati (*bursty*)

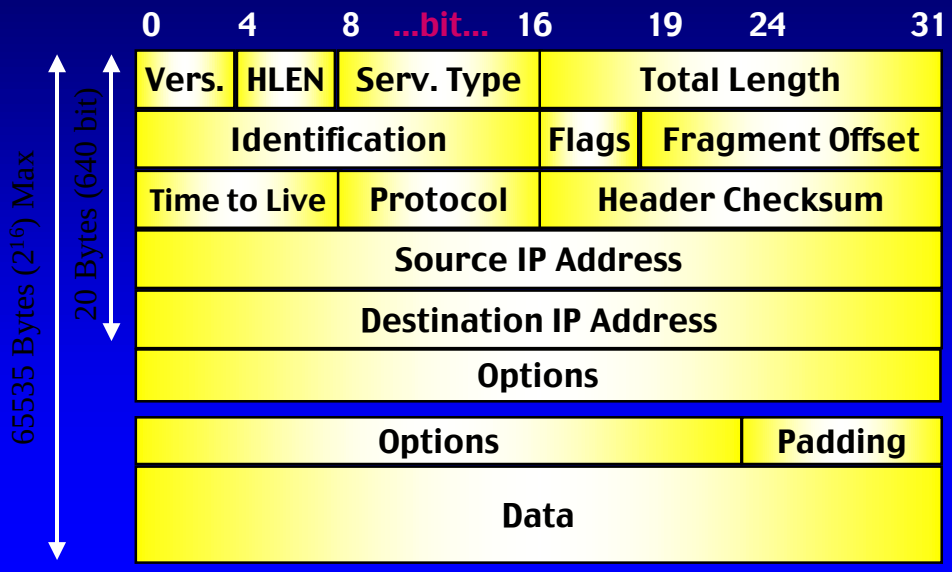
6.5

IP

- Gestione indirizzi a 32 bit a livello di rete e di *host*
- *Routing*
- Frammentazione e riassemblaggio dei pacchetti
- Rivelazione (ma non correzione) di errori (sull'intestazione)

6.6

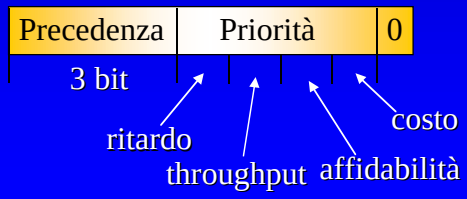
IP Datagram



6.7

IP Datagram

- Versione
- Header Lenght
  - lunghezza dell'header, in blocchi di 4 byte (max 64 byte)
- Service Type
  - tipo di servizio, attualmente non usato ma importante in prospettiva.



6.8

IP Datagram

- Total Lenght (16 bit)
  - lunghezza globale del pacchetto corrente (non quello prima della frammentazione), max 2^{16}-1 = 65.535 ottetti (576 ottetti il minimi che un router deve saper gestire senza frammentazione).
- Identification (16 bit)
  - ID univoco del pacchetto (costante nel caso di frammentazione), necessario per la deframmentazione
- Flags (3 bit)
  - 0: posto a zero
  - DF: Don't Fragment
  - MF: More Fragment (0 sull'ultimo frammento)

6.9

IP Datagram

- Fragment Offset (13 bit)
  - In multipli di 8 byte, quindi è in grado di rappresentare fino a 65 535 byte.
- Time To Live (8 bit)
  - contatore decrementato ad ogni hop (una unità in meno per ogni secondo di attesa nel router).
- Protocol (8 bit)
  - TCP (6), UDP (17), ICMP, ...
- Header Checksum
- Source - Destination IP Address

6.10

IP Datagram

Copy	Class	Number
1 bit	2 bit	5 bit

- Il *copy bit* decide se le opzioni vanno copiate in tutti i pacchetti in caso di frammentazione
- Classe 0 denota controllo della rete o del datagram, classe 2 *debugging* o misure.
- Le opzioni possibili sono:
  - *Loose and strict source routing*
  - *Record/trace route*
  - *Timestamp*

6.11

IP Datagram

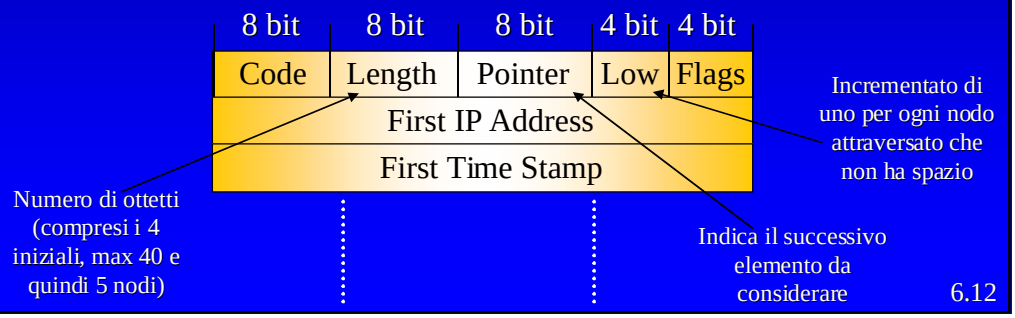
Opzione Timestamp

Tre tipi di comportamento:

Flag = 0: Registra solo i *timestamp*

Flag = 1: Registra *timestamp* e gli indirizzi IP

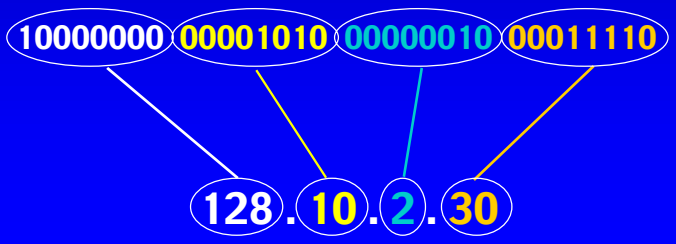
Flag = 3: La lista degli indirizzi viene inserita dalla sorgente, i *timestamp* vengono inseriti solo dai nodi attraversati presenti nella lista



6.12

Indirizzi IP

- Gli indirizzi IP sono indirizzi univoci, assegnati da una autorità centrale, e hanno una lunghezza di 32 bit.
- Tali indirizzi sono composti da due parti:
  - l'indirizzo della rete (netid)
  - l'indirizzo del *host* (hostid)
- L'indirizzo è legato alle interfacce di rete.



6.13

Indirizzi IP

Classe A - /8

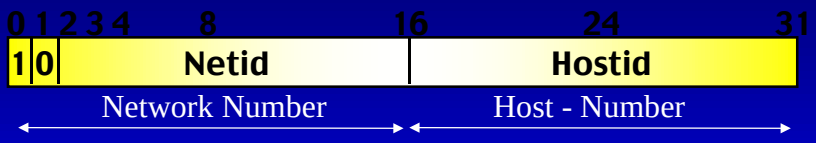
Netid validi	Hostid validi
1.x.x.x - 127.x.x.x	**x.0.0.1 - x.255.255.254**
Max num. di reti	Max numero di host
126 (2^7-2)	**16.777.214** (2^{24}-2)
(non va contata la rete 127 e la 0)	(non va contato l'ind. tutti 0 e quello tutti 1)

Considerando che lo spazio complessivamente disponibile è di 4.294.967.296 indirizzi (2^{32}) le reti di classe A coprono circa il 50% dello spazio di indirizzamento disponibile)

6.14

Indirizzi IP

Classe B - /16



Netid validi
128.1.x.x - 191.254.x.x
Max num. di reti
16.384 (2^{14})

Hostid validi
x.x.0.1 - x.x.255.254
Max numero di host
65.534 (2^{16}-2)

Comprendono circa 1.073.741.824 (2^{30}) indirizzi, la classe B rappresenta circa il 25% dello spazio di indirizzamento complessivo

6.15

Indirizzi IP

Classe C - /24

Netid validi
192.0.1.x - 223.255.254.x
Max num. di reti
2.097.152 (2^{21})

Hostid validi
x.x.x.1 - x.x.x.254
Max numero di host
254 (2^{8}-2)

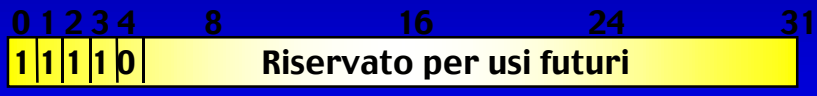
Comprendono circa 536.870.912 (2^{29}) indirizzi, la classe C rappresenta circa il 12,5% dello spazio di indirizzamento complessivo

6.16

Indirizzi IP

Classe D (224.0.0.1 - 239.255.255.254)

Classe E (240.0.0.1 - 255.255.255.254)



6.17

Indirizzi IP particolari

This Host (solo come ind. sorgente nel *bootstrap*)	Tutti 0	
Host on this net. (solo come ind. sorgente nel *bootstrap*)	Tutti 0	Hostid
Limited broadcast (solo come ind. destinazione)	Tutti 1	
Directed broadcast (solo come ind. destinazione)	Netid	tutti 1
Loopback (non deve venir propagato dai router)	127	qualunque numero

6.18

Indirizzi IP - Netid particolari

- Alcuni netid sono riservati per essere usati su reti private
- Non sono "annunciati" su Internet, quindi non sono raggiungibili direttamente

172.16.0.0 – 172.31.255.255	16 indirizzi di classe B
10.0.0.0 – 10.255.255.255	1 network di classe A
192.168.0.0 – 192.168.255.255	256 indirizzi di classe C

6.19

IP

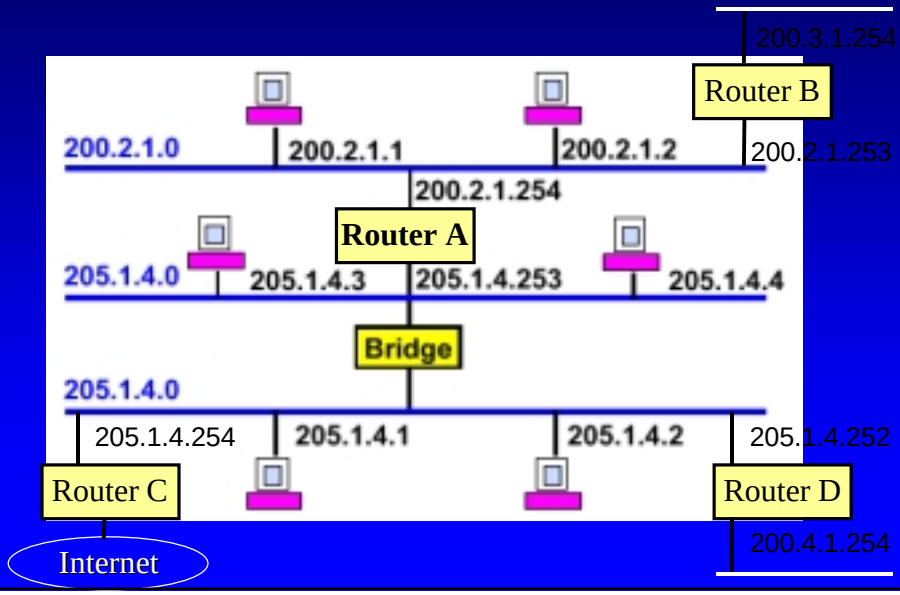
Instradamento

- IP assume una corrispondenza biunivoca tra reti fisiche (intese come dominii di *broadcasting* a livello 2) e logiche.
  - Si osservi che
    - » le realizzazioni moderne ammettono
      - più reti logiche sulla stessa rete fisica.
      - più reti fisiche nella stessa rete logica (Proxy ARP)
    - » Una conseguenza del fatto che l'indirizzo contenga l'identificatore di una rete è che quando una macchina viene fisicamente spostata il suo indirizzo deve essere modificato.

6.20

IP - Instradamento

Esempio



6.21

IP - Instradamento

Reti fisiche e reti logiche

- Punto-punto
  - le interfacce possono essere "*unnumbered*" (es.: linee dedicate o dial-up)
- Multiaccesso con possibilità di *broadcast*
  - gli host possono comunicare direttamente senza passare per router intermedi (es. : le LAN)
- Multiaccesso senza possibilità di broadcast
  - gli host possono comunicare direttamente senza passare per router intermedi (es. : reti a pacchetto commutate)

6.22

IP

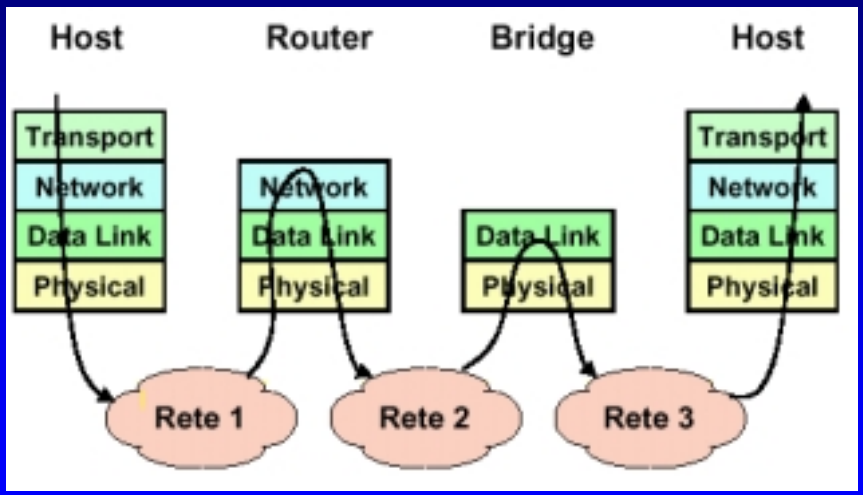
Instradamento

- L'instradamento viene quindi realizzato:
  - all'interno di una rete logica, in modo implicito, direttamente dalla stazione sorgente
    - » ossia deve essere realizzato dal livello 2 e quindi si deve fornire la "traduzione" dell'indirizzo IP di destinazione in indirizzo di livello due (ARP -RARP)
  - Tra reti logiche diverse è gestito esplicitamente dai router, ossia è il router che deve ricevere il pacchetto e che quindi instradarlo verso la sottorete opportuna
    - » La stazione sorgente deve quindi indirizzare il pacchetto al router, questo implica che:
      - Deve essercene almeno uno direttamente connesso alla rete fisica
      - La stazione deve conoscerne l'indirizzo (default router)

6.23

IP

Instradamento



6.24

IP - Instradamento

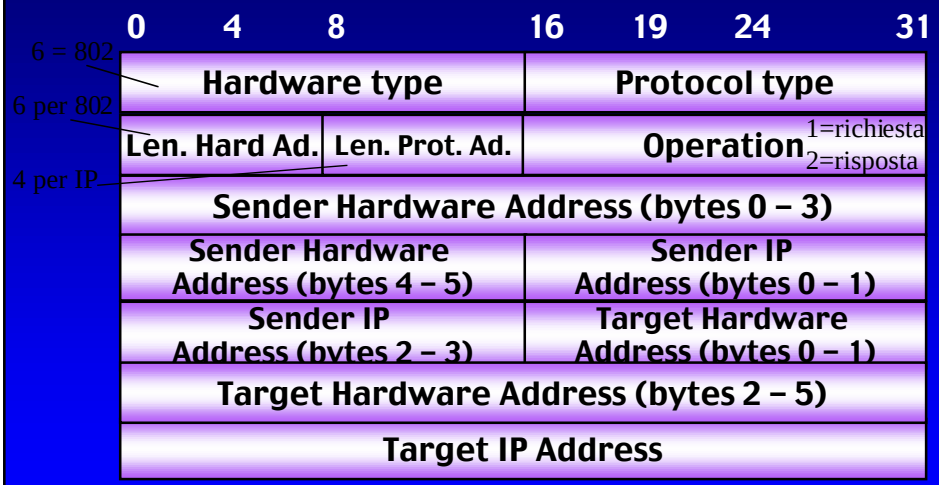
(R)ARP

- I protocolli ARP (*Address Resolution Protocol*) e RARP (*Reverse Address Resolution Protocol*) servono per definire in modo automatico le corrispondenze fra indirizzi di livello 2 ed indirizzi IP e viceversa.
  - ARP viene usato tutte le volte che una stazione vuole inviare un pacchetto ad un'altra stazione sulla sottorete, di cui conosce solo l'indirizzo IP.
  - RARP viene usato dalle stazioni non dotate di memoria di massa (*diskless*) per reperire il propio indirizzo IP all'avvio (*bootstrap*).
- Si appoggiano direttamente sui protocolli di livello 2 della sottorete e non su IP.

6.25

IP - Instradamento

ARP



6.26

IP - Instradamento

ARP

- La stazione A manda in *broadcast* un pacchetto ARP contenente l'indirizzo IP di cui vuol conoscere il corrispondente indirizzo di livello 2.
- La stazione B che riconosce il proprio ind. IP risponde fornendo il suo indirizzo di livello 2.
- Con il primo pacchetto ARP la stazione A fornisce anche il proprio indirizzo di livello 2, così che B può risponderle senza usare un *broadcast*.
- La corrispondenza resta memorizzata in una memoria di cache per un certo periodo.

6.27

IP - Instradamento

Instradamento nei router

- Nei router l'intradamento è realizzato tramite una tabella di instradamento (*Routing Table,* RT) del tipo:

Destinazione (rete o host)	**Indirizzo di invio**
200.2.1.0	Invio diretto sulla interf. 1
205.1.4.0	Invio diretto sulla interf. 2
200.3.1.0	200.2.1.253
200.4.1.0	205.1.4.252
default	205.1.4.254

6.28

IP - Instradamento

Instradamento nei router

- Due sono quindi gli aspetti di cui si compone l'instradamento:
  - Esecutivo: la scelta della direzione di uscita tramite la tabella
  - Algoritmico: la compilazione/aggiornamento della tabella
- Il secondo aspetto si realizza tramite
  - il calcolo del percorso migliore eseguito secondo un qualche algoritmo
  - Lo scambio di informazioni fra i router per eseguire tale calcolo

6.29

IP - Instradamento

Instradamento nei router

- Per rendere l'instradamento efficiente si deve mantenere le RT di piccole dimensione.
- Tabelle grandi:
  - Richiedono più tempo per l'individuazione della corretta direzione di uscita (*next hop*)
  - Sono di difficile gestione in fase di calcolo e di aggiornamento.
- La suddivisione net e host, crea una gerarchia che ha l'obiettivo di ridurre la dimensione delle RT.
- Lo stesso vale per la presenza del "default router"

6.30

IP - Instradamento

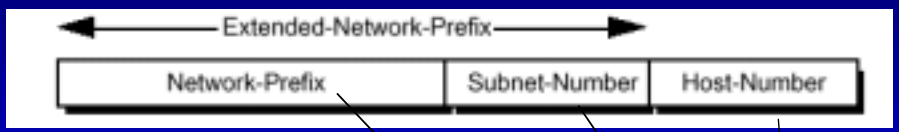
Routing Table

- Tipiche informazioni contenute nelle RT per ciascune delle reti destinazione sono
  - Indirizzo della rete destinazione
  - Maschera di *subnet*
  - Indirizzo IP del successivo *router* da attraversare (*next hop*) o sul fatto che la destinazione è direttamente raggiungibile
  - Porta di uscita del "*next hop*"
  - Metrica (anche più di una)
  - Identificatore della sorgente dell'instradamento (manuale, locale, ICMP, uno degli algoritmi di instradamento)

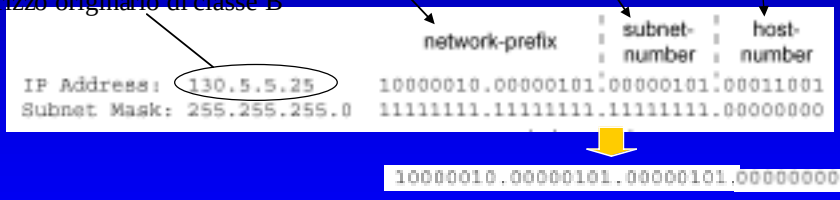
6.31

IP - Instradamento

Subnetting



Indirizzo originario di classe B



Notazione concisa

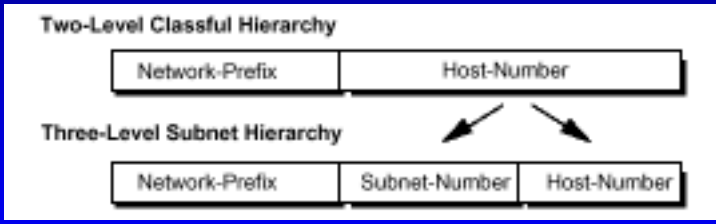
130.5.5.25/24 10000010.00000101.00000101.00011001

6.32

IP - Instradamento

Subnetting

- Il *subnetting* aggiunge un livello di gerarchia all'indirizzamento, contribuendo a ridurre la dimensione delle RT.

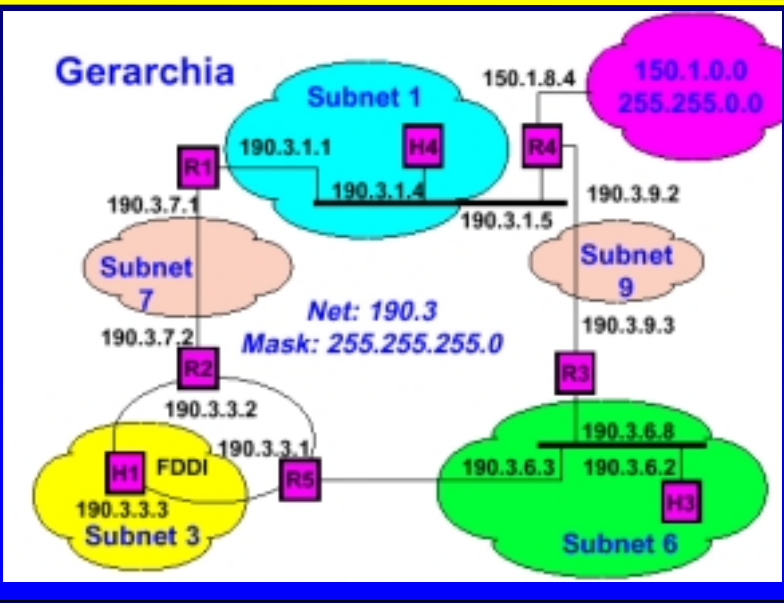


- All'interno di una classe di indirizzi la destinazione è ora individuata dalla coppia (*IP-address, Netmask*)

6.33

IP - Instradamento

Subnetting



6.34

IP - Instradamento

Subnetting

- Il router R5 avrà una tabella del tipo

Destinazione		next hop
190.3.6.0/24	255.255.255.0	diretta
190.3.3.0/24	255.255.255.0	diretta
190.3.1.0/24	255.255.255.0	190.3.3.2
190.3.7.0/24	255.255.255.0	190.3.3.2
190.3.9.0/24	255.255.255.0	190.3.6.8
150.1.0.0/16	255.255.0.0	190.3.6.8
Default		190.3.6.8

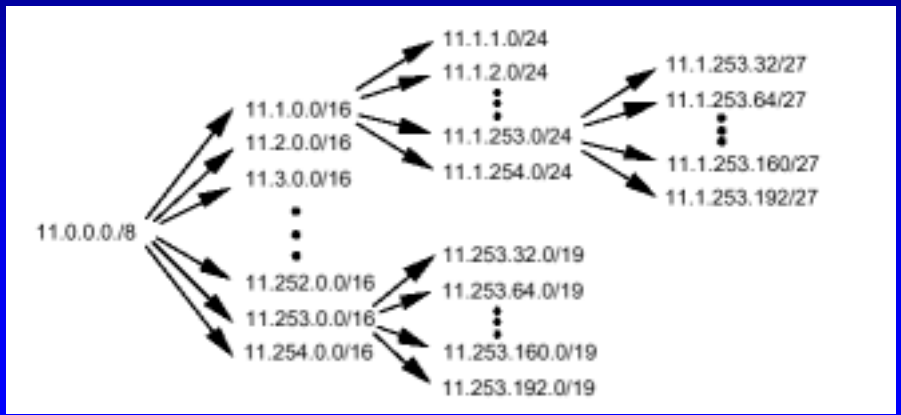
- Gli host dovranno conoscere il proprio ind. IP, la Netmask e l'indirizo del router di default. ad es. per l'host H3 avrà:
  - Ind. IP = 190.3.6.2, NetMask = 255.255.255.0 (ff.ff.ff.0), default router = 190.3.6.8.

6.35

IP - Instradamento

Variable Length Subnetting

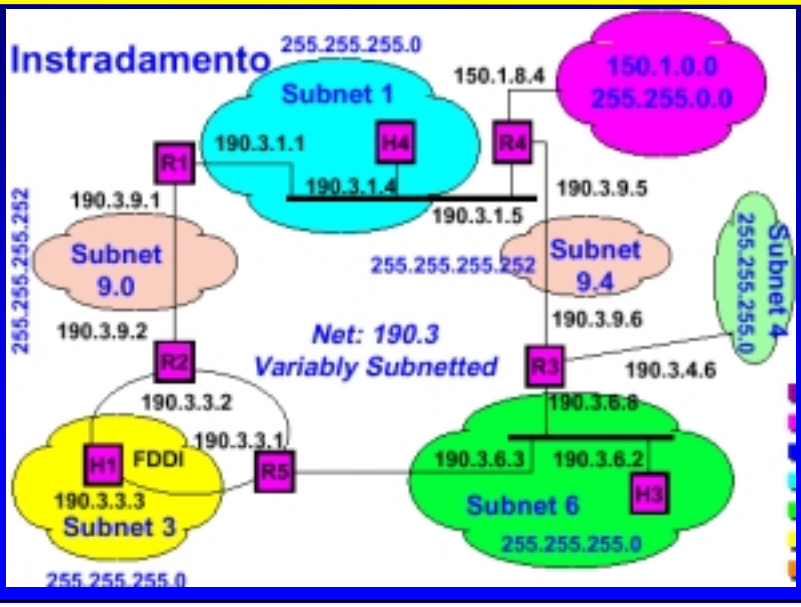
- All'interno di una singola classe si possono realizzare gerarchie multiple:



6.36

IP - Instradamento

Esempio



6.37

IP - Instradamento

Variable Length Subnetting

- La maschera variabile permette di sfruttare meglio lo spazio di indirizzamento e ridurre ulteriormente le RT.
- In presenza di più di una scelta per una destinazione si sceglie quella con *subnet mask* più lunga
- Si consideri ad es. la RT di R5

Destinazione		*Next hop*
190.3.6.0/24	255.255.255.0	diretta
190.3.3.0/24	255.255.255.0	diretta
190.3.9.0/24	255.255.255.0	190.3.3.2
190.3.9.4/30	255.255.255.252	190.3.6.8
190.3.4.0/24	255.255.255.0	190.3.6.8
190.3.0.0/16	255.255.0.0	190.3.3.2
150.1.0.0/16	255.255.0.0	190.3.6.8
Default		190.3.6.8

6.38

IP - Instradamento

Proxy ARP

- E' una tecnica che permette la corrispondenza di più reti fisiche ad una sola rete o sottorete logica .
- Un *router* risponde agli ARP verso indirizzi IP che sa non appartenere a quella rete fisica con il proprio indirizzo di livello 2.
- L'appartenenza o meno ad una rete viene ricavata attraverso un *subnetting* che solo il *router* è obbligato a conoscere.

6.39

IP - Instradamento

Classless InterDomain Routing (CIDR)

- La crescita degli utenti nella rete ha velocemente portato verso l'esaurimento lo spazio di indirizzamento disponibile.
- La ragione principale è legata al fatto che in molte situazioni le reti di classe C sono troppo piccole, quindi viene richiesto un indirizzo di classe B di cui però va sprecato gran parte dello spazio di indirizzamento.
- Per cui si è definito un meccanismo di "supernetting" che consiste nel accorpare indirizzi di classe C contigui in un unico spazio di indirizzamento creando suddivisioni netid-hostid ad hoc.

6.40

IP - Instradamento

Classless InterDomain Routing (CIDR)

- Il CIRD trasforma lo spazio di indirizzamento della classe C in un unico spazio "senza classe" che viene suddiviso usando come "quanti" le reti di classe C con un meccanismo di subnetting.
- I router in grado di gestire tale meccanismo, operano usando la coppia indirizzo IP - netmask per identificare il "*next-hop*"
- Quindi "annunciano" la coppia ed in presenza della netmask ignorano la definizione di classe C (effettuano un *supernetting*)

6.41

IP - Instradamento

Classless InterDomain Routing (CIDR)

- Si supponga che ad un ISP sia stato assegnato il blocco di indirizzi 206.0.64.0/18.
- Questo blocco rappresenta 16.384 IP *address* che possono essere interpretati come 64 reti da x/24.
- Se un cliente richiede 800 *host addresses*, invece che assegnargli una classe B (e perdere 64.700 indirizzi) o quattro Classi C (introducendo 4 nuove reti nelle RT di Internet), l' ISP può assegnare al cliente il blocco 206.0.68.0/22, con 1.024 indirizzi IP

```
ISP:          206.0.64.0/18 11001110.00000000.01000000.00000000
Client:       206.0.68.0/22 11001110.00000000.01000100.00000000
Class C #0: 206.0.68.0/24 11001110.00000000.01000100.00000000
Class C #1: 206.0.69.0/24 11001110.00000000.01000101.00000000
Class C #2: 206.0.70.0/24 11001110.00000000.01000110.00000000
Class C #3: 206.0.71.0/24 11001110.00000000.01000111.00000000
```

6.42

IP - Instradamento

Classless InterDomain Routing (CIDR)

CIDR prefix-length	Dotted-Decimal	# Individual Addresses	# of Classful Networks
/13	255.248.0.0	512 K	8 Bs or 2048 Cs
/14	255.252.0.0	256 K	4 Bs or 1024 Cs
/15	255.254.0.0	128 K	2 Bs or 512 Cs
/16	255.255.0.0	64 K	1 B or 256 Cs
/17	255.255.128.0	32 K	128 Cs
/18	255.255.192.0	16 K	64 Cs
/19	255.255.224.0	8 K	32 Cs
/20	255.255.240.0	4 K	16 Cs
/21	255.255.248.0	2 K	8 Cs
/22	255.255.252.0	1 K	4 Cs
/23	255.255.254.0	512	2 Cs
/24	255.255.255.0	256	1 C
/25	255.255.255.128	128	1/2 C
/26	255.255.255.192	64	1/4 C
/27	255.255.255.224	32	1/8 C

6.43

IP - Instradamento

Network Address Translation (NAT)

- Consiste nell'usare nella rete una delle classi assegnate all'uso privato e quindi non "annunciate" e far tradurre al router di accesso ad Internet gli indirizzi verso l'esterno (usando uno spazio di indirizzi "ufficiale" ridotto) in modo dinamico.
- Esistono dei limiti di questo metodo:
  - Per UDP e TCP si deve ricalcolare il *checksum*
  - FTP ha il numero IP scritto in ASCII nel suo interno, cambiarlo può cambiare la lunghezza del pacchetto e avere effetti sul TCP.
  - ICMP ha l'indirizzo IP nella parte dati
  - Tutte le applicazioni che trasportano l'indirizzo IP possono aver problemi.

6.44

Instradamento

- Requisiti
  - Minimizzare lo spazio occupato dalle RT per:
    - » Velocizzarela commutazione
    - » Semplificare i router (meno cari)
    - » Ridurre l'informazione necessaria all'aggiornamento
  - Minimizzare il traffico di controllo
  - Robustezza, ossia evitare:
    - » Cicli
    - » Buchi neri
    - » Oscillazioni
  - Ottimizzare i percorsi (dal punto di vista della distanza, del ritardo, del costo economico, …)

6.45

Instradamento

Alternative

- Centralizzato o **distribuito** (o isolato)
- Basato sulla sorgente o ***"hop-by-hop"***
- **Deterministico** o stocastico
- **Singolo percorso** o multi-percorso
- Dipendente dallo stato (**dinamico**) o indipendente dallo stato (**statico**)
- ***Distance Vector*** o ***Link State***.

6.46

Instradamento

Distance vector

- Ogni nodo (router) conosce l'identità di tutti nodi della rete e i nodi a lui direttamente connessi (vicini).
- Ogni nodo mantiene un *Distance Vector*, ossia una lista di coppie (destinazione, costo) per tutte le possibili destinazioni.
- Il costo è la somma stimata dei costi sui singoli link sul percorso "più corto" (*shortest path*) verso quella destinazione.
- Ogni nodo inizializza i costi relativi a destinazioni "lontane" ad una valore alto, convenzionalmente indicato infinito.

6.47

Instradamento

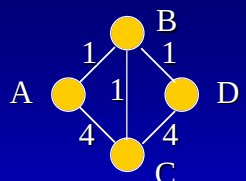
Distance vector

- Periodicamente ogni nodo invia ai propri vicini il proprio DV.
- Quando un *router* A riceve un DV da B (suo vicino), verifica quali sarebbero i costi per le varie destinazioni usando come transito B; per le destinazioni in cui tali costi risultano minori di quelli attuali, sostituisce il costo vecchio con quello calcolato e lo stesso fa con il next-hop nella RT.
- Questo modo di procedere corrisponde all'applicare in forma distribuita ed asincrona l'algoritmo di Bellman-Ford.

6.48

Instradamento

Distance vector



Situazione Iniziale

	A	B	C	D
A	0	1	4	∞
B	1	0	1	1
C	4	1	0	4
D	∞	1	4	0

1	Costo per raggiungere B da A
+	
1 0 1 1	Costi da B agli altri
=	
2 1 2 2	Costi passando per B
0 1 4 ∞	Costi attuali in A
min	
0 1 2 2	Nuovo DV
- - B B	*Next hop*

6.49

Instradamento

Distance vector

- Questo procedimento corrisponde a realizzare in modo distribuito e asincrono l'algoritmo di Bellman-Ford, ossia ogni nodo i esegue l'iterazione

$$D_i \leftarrow \min_{j \in N(i)} \{d_{ij} + D_j\}$$

  (dove $N(i)$ è l'insieme dei nodi adiacenti ad i), usando le stime D_j più recenti ricevute dai vicini e trasmettendo D_i ai propri vicini.
- Non è necessaria una inizializzazione con particolari valori di D_j.

6.50

Instradamento

Distance vector

Sia A il numero di archi e N quello dei nodi

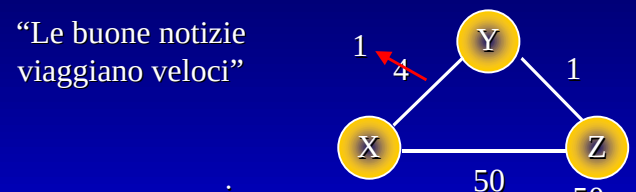
- Nel caso peggiore, l'algoritmo di Bellman-Ford centralizzato compie N-1 iterazioni, ciascuna su N-1 nodi, con al più N-1 alternative per nodo, il che porterebbe a complessità **O(N^3)**.
- Si può mostrare che la complessità è **O(mA)**, con m numero di iterazioni per la convergenza. Questo porta una complessità generalmente compresa fra **O(N^2)** e **O(N^3).**
- Nel caso distribuito, se le iterazioni fossero eseguite in modo sincrono (simultaneamente ad ogni nodo), scambiando ad ogni iterazione i risultati con i vicini, partendo dalle condizioni iniziali $D_i^0 = \infty$ per tutti gli $i \neq 1$ e $D_1^0 = 0$, l'algoritmo convergerebbe in al più N-1 passi.

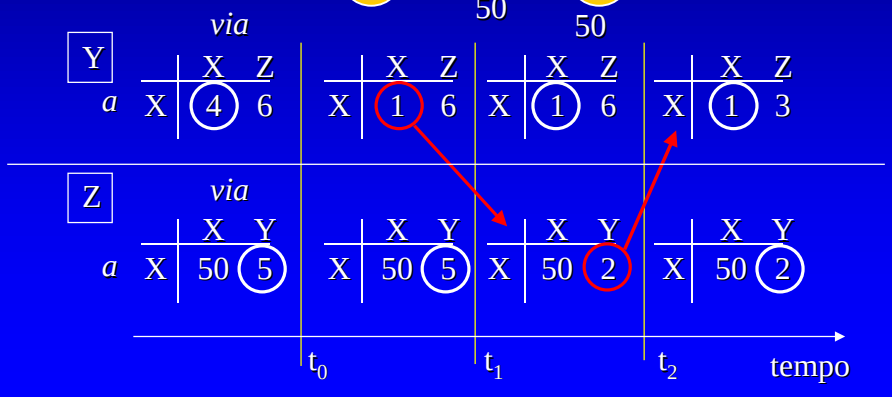
6.51

Instradamento

Distance vector

"Le buone notizie viaggiano veloci"



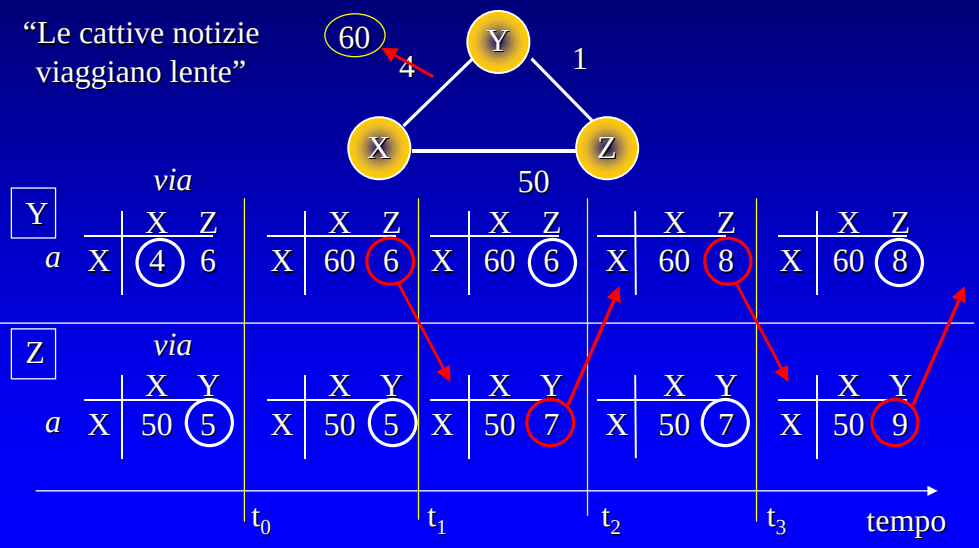


6.52

Instradamento

Distance vector

"Le cattive notizie viaggiano lente"

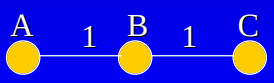


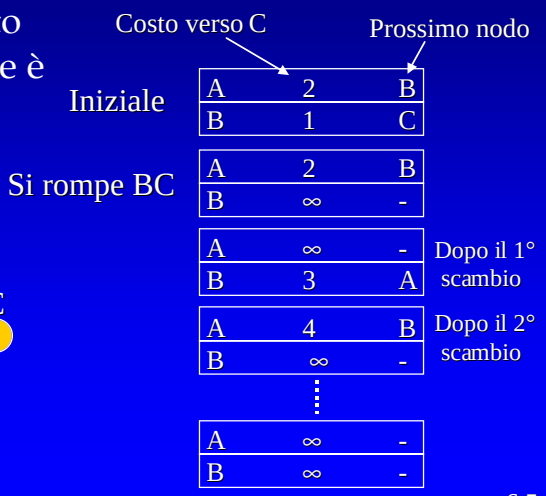
6.53

Instradamento

Distance vector

- Questo tipo di algoritmo ha un problema legato all'aggiornamento che è chiamato ***Count-to-infinity*** :





6.54

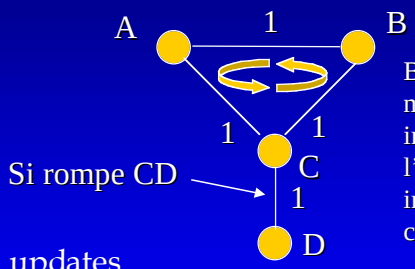
Instradamento
Distance vector

- **Ci sono diverse possibili soluzioni al count to infinity**
  - ***Path vector***
    - » oltre al costo si trasmette il percorso (*path-vector*), in questo modo i nodi possono capire quando non esiste più un percorso valido verso una certa destinazione. (BGP)
  - *Split horizon*
    - » Non viene passato il costo per una certa destinazione ad un vicino se questi è il *next hop* per quella destinazione.
    - » Una versione più complessa detta s*plit horizon with poisonous reverse*, invece di non passare costi passa un costo infinito, questo a volte accelera la convergenza. (RIP)
    - » Nel caso precedente, A non invia a B un costo (o lo invii infinito) verso C. Il ciclo quindi non si crea.

6.55

Instradamento
Distance vector

» Se ho ,più nodi coinvolti nella rottura direttamente non si riesce a bloccare il conto ad inf..



B e A hanno sempre mandato a C un costo infinto verso D, ma non l'uno all'altro; questo innesca un ciclo che coinvolge A, B e C.

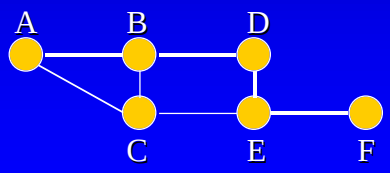
- Triggered updates
  - » Normalmente, per evitare un numero eccessivo di aggiornamenti delle tabelle e di traffico di controllo, si limita il ritardo minimo fra due aggiornamenti consecutivi (per es. 30 sec.). Nel caso di collegamento caduto, gli aggiornamenti sono fatti immediatamente (riduce il tempo di convergenza). (RIP)

6.56

Instradamento

Distance vector

- *Source tracing*
  - » Insieme al costo i nodi si scambiano anche il nodo da attraversare immediatamente prima della destinazione. Con questa informazione aggiuntiva è possibile ricavare direttamente dalla tabella il percorso complessivo e quindi, mantenendo la RT più piccola si ottiene lo stesso del *path vector*.



RT di A

Dest.	Succ.	Ultimo
A	-	
B	B	A
C	C	A
D	B	B
E	B	D
F	B	E

6.57

Instradamento

Link-state

- La filosofia del link state routing è quella di distribuire a tutti i nodi della rete l'intera sua topologia ed i costi di ogni *link* che la compone.
- Con questa informazione ogni *router* è in grado ci calcolarsi i propri percorsi ottimi verso ogni destinazione.
- Se tutti vedono gli stessi costi e tutti usano lo stesso algoritmo, i percorsi saranno liberi da cicli.
- Quindi sono due gli aspetti caratterizzanti questo metodo
  - Il modo in cui la topologia della rete viene diffusa fra i nodi.
  - Il modo in cui ogni nodo calcola i percorsi ottimi.

6.58

Instradamento

Link-state - Dijkstra (Cont.)

- Nel caso LS ogni nodo applica l'algoritmo di Dijkstra.
- A diffirenza dell'algoritmo di Bellman-Ford che itera sul numero di archi attraversati da un percorso, l'algoritmo di Dijkstra itera sulla lunghezza del percorso.
- Nel caso peggiore la sua complessita è **$O(N^2)$**, in media si colloca intorno a **$O(A \log A)$** con A numero degli archi .

6.59

Instradamento

Link-state - Dijkstra

- Sia P un insieme di nodi e D_i la distanza minima "stimata" dal nodo 1.
- Fissando inizialmente
  $P = \{1\}$, $D_1 = 0$, $D_j = d_{j1}$ per tutti i $j \neq 1$
- I passi dell'algoritmo di Dijkstra sono

  1 Trova $i \notin P$ tale che
  $$D_i = \min_{j \notin P} \{D_j\}$$
  e poni $P \leftarrow P \cup \{i\}$.
  Se P contiene tutti i nodi l'algorimo è completato.

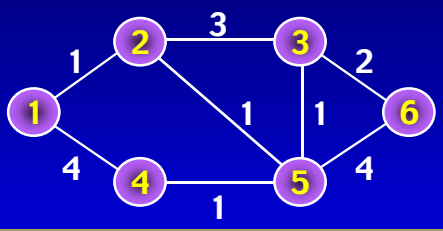
  2 Per tutti i $j \notin P$, poni $D_j \leftarrow \min\{D_j, d_{ji} + D_i\}$
  e torna al passo 1

6.60

Instradamento

Link-state - Dijkstra

Bellman–Ford



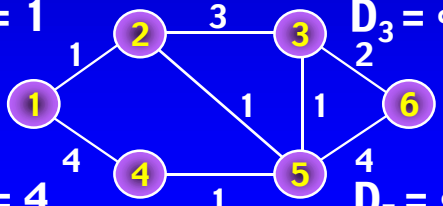
$d_{ij} = d_{ji}$

$\forall$ (i, j)

P = {1, 2}

Dijkstra

$D_2 = 1$ $D_3 = \infty$ $D_6 = \infty$ $D_4 = 4$ $D_5 = \infty$



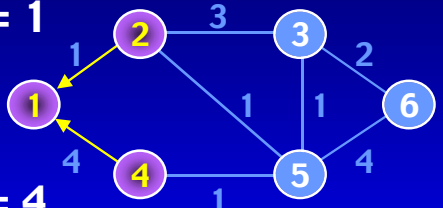
6.61

Instradamento

Link-state - Dijkstra

Bellman–Ford

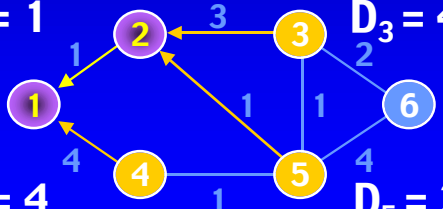
$D_2^1 = 1$ $D_4^1 = 4$



Dijkstra

$D_2 = 1$ $D_3 = 4$ $D_6 = \infty$ $D_4 = 4$ $D_5 = 2$

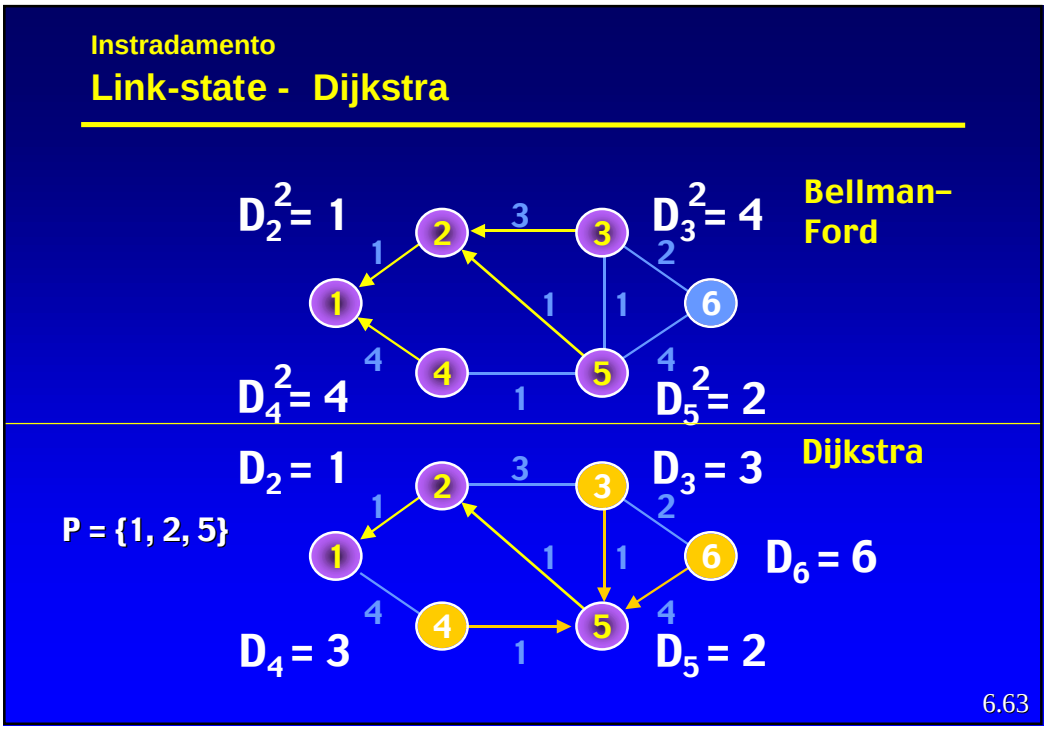
P = {1, 2}



6.62

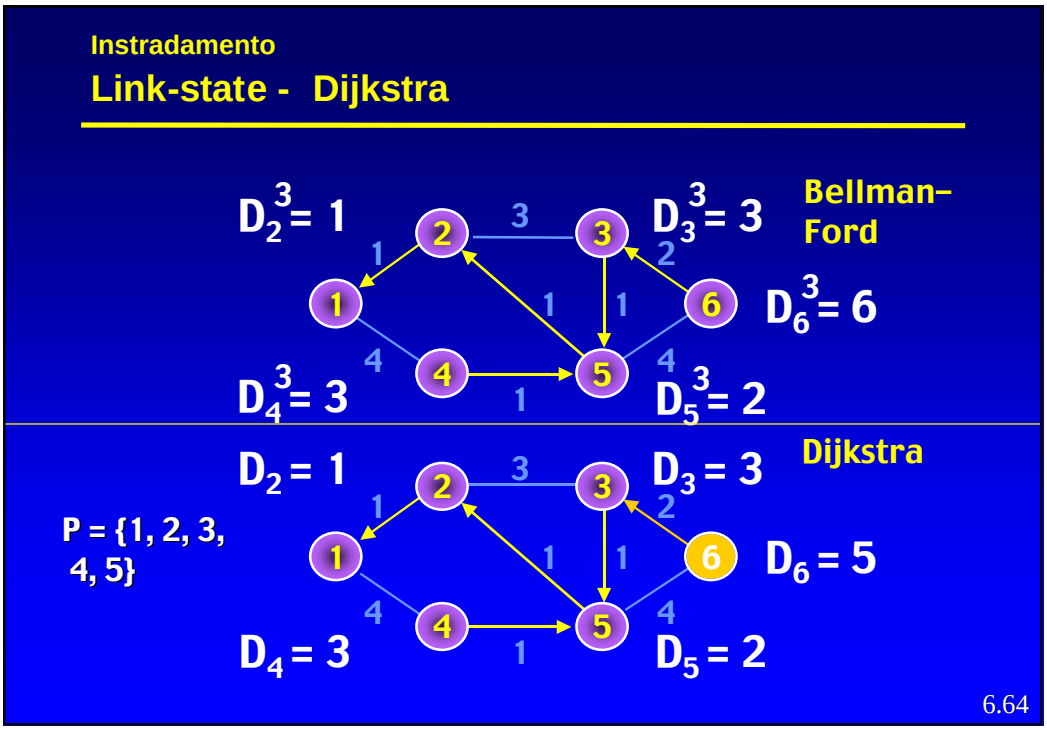
Instradamento

Link-state - Dijkstra



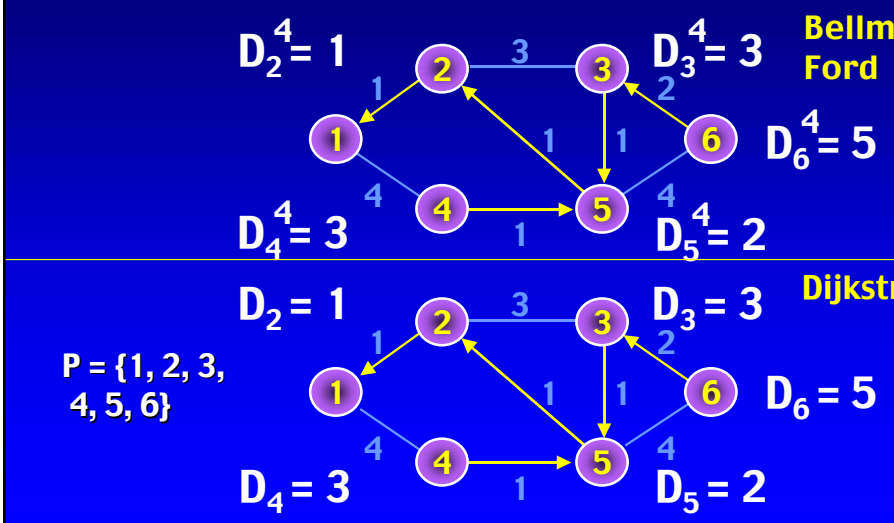
Instradamento

Link-state - Dijkstra



Instradamento

Link-state - Dijkstra

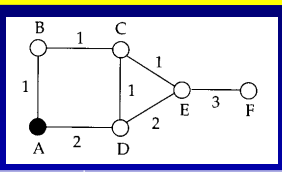


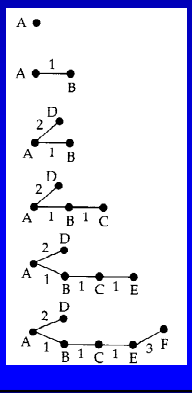
6.65

Instradamento

Link-state - Dijkstra

C(B,1) significa che C è raggiunto tramite B, con costo 1





Definitivo (P)	Temporaneo	Commenti
A	B(A,1),D(A,2)	La radice e i suoi vicini
A, B(A,1)	D(A,2), C(B,2)	Aggiunto C
A, B(A,1), D(A,2)	E(D,4), C(B,2)	Scartato C(D,3)
A, B(A,1), D(A,2), C(B,2)	E(C,3)	Scartato D(D,4)
A, B(A,1), D(A,2), C(B,2), E(C,3)	F(E,6)	
A, B(A,1), D(A,2), C(B,2), E(C,3), F(E,6)	Vuoto	Stop

6.66

Instradamento
Link-state

- Disseminazione della topologia
  - Ogni nodo crea un insieme di ***Link-State-Packet*** (LSP) che descrivono le sue linee in uscita.
  - Ogni LSP contiene l'indirizzo del nodo, quello dei nodi vicini, ed il costo delle linee verso i nodi vicini.
  - Ogni LSP viene distribuito a tutti i nodi tramite un *controlled flooding*
    - » Ogni nodo che riceve un LSP lo memorizza in un database e invia una copia su tutte le proprie linee in uscita, tranne quella da cui l'ha ricevuto. Si può dimostrare che nessun LSP passa due volte per lo stesso *link* lungo la stessa direzione, quindi un LSP viene distribuito in al più 2L invii, dove L è il numero dei *link*.

6.67

Instradamento
Link-state

- Numero di sequenza
  - Per poter decidere se un LSP ricevuto è significativo (contiene una informazione più recente di quella attualmente nel nodo) ogni LSP deve contenere un numero di sequenza progressivo.
  - Il numero di sequenza ha valore locale per ogni tipo di LSP (identificato da coppia ordinata di nodi collegati da una linea)
  - Ogni volta che un nodo riceve un LSP più vecchio di quello in memoria, lo elimina senza propagarlo.

6.68

Instradamento
Link-state

- Le sequenze realmente utilizzabili sono di lunghezza finita e quindi soggetta ad "avvolgersi" (*wrapping*) bloccando l'aggiornamento.
- *Wrapped sequence number*
  - Per evitare il problema si può prendere una sequenza molto grande (32 bit => 4.295.967295) e decidere che quando due numeri distano troppo, il più piccolo sia anche il più giovane. Per es. supponendo che N sia lunghezza della sequenza, allora ***a*** è più vecchio di ***b*** se
    - » $a < b$ e $|b - a| < N/2$
      oppure se
    - » $a > b$ e $|b - a| \geq N/2$

6.69

Instradamento
Link-state

- La presenza di un numero di sequenza pone il problema dell'inizializzazione della sequenza quando un nodo si (ri)attiva. Due sono i meccanismi adottati
  - Invecchiamento (*Aging*)
  - *Lollipop sequence space*

6.70

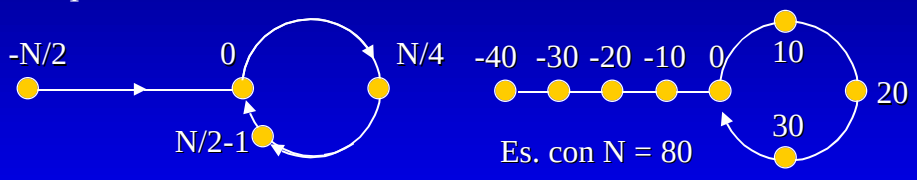
Instradamento
Link-state

- Invecchiamento
  - Prevede l'inserimento di un campo di di anzianità nel LSP, che viene inizializzato ad un valore (MAX_AGE) dal creatore del pacchetto.
  - Ogni nodo copia in un contatore C_AGE il valore MAX_AGE e lo decrementa periodicamente.
  - Quando in un *router* C_AGE raggiunge zero, la corrispettiva informazione viene eliminata dal DB e viene generato un LSP con anzianità zero, per forzare la stessa operazione sugli altri *router*.
  - E' difficile fissare un valore ottimale per MAX_AGE (troppo corto: scade prima di essersi propagato; troppo lungo: un nodo che riparte deve attendere a lungo perché i nuovi pacchetti diventino significativi)

6.71

Instradamento
Link-state

- *Lollipop sequence space*
  - Si tratta di una sequenza che si "avvolge" in modo particolare



  - In questo caso **a** è più vecchio di **b** se:
    - » $a < 0$ e $a < b$ o
    - » $a > 0, a < b$ e $|b - a| < N/4$, o
    - » $a > 0, b > 0, a > b$ e $|b - a| > N/4$

6.72

Instradamento
Link-state

- *Lollipop sequence space*
  - Quando un nodo riceve un LSP con un numero di sequenza più vecchio di quello nel DB, lo comunica a chi gli ha inviato il pacchetto fornendo anche l'ultimo valore di sequenza memorizzato.
  - Un nodo che riparte genera sempre un numero di sequenza più vecchio degli altri e quindi i nodi vicini gli inviano l'ultimo valore da lui usato da cui può ripartire aggiungendogli 1.
  - In pratica i *router* vicini si comportano come una sorta di memoria distribuita.

6.73

Instradamento
Link-state

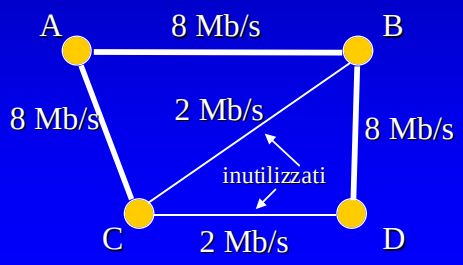
- Ci sono alcune altre considerazioni di criticità da fare
  - Se la rete di partiziona per la caduta di una o più linee e le singole parti evolvono indipendentemente, quando si ricollegano possono crearsi problemi (*loop*). (Soluzione: scambio fra nodi vicini di parti di DB)
  - Se invece di una linea, si rompe un nodo, non c'è nessuno che propaga l'informazione. (Soluzione: pacchetti di Hello e anzianità massima degli LSP nel DB)
  - Bisogna proteggere gli LSP da "corruzioni" casuali o volute.

6.74

Instradamento

Metriche e dinamica

- Metriche statiche
  - In genere si tratta di valori inversamente proporzionali alla capacità del *link*. La staticità fa si che le linee a minor velocità tendano ad essere sotto-utilizzate.



Pesi
8 Mb/s = 1
2 Mb/s = 4

6.75

Instradamento

Metriche e dinamica

- Dinamiche
  - Le metriche dipendenti dal traffico sono sicuramente più efficaci, ma comportano alcuni problemi.
  - Consideriamo la sperimentazione avvenuta su ARPAnet, dove in origine si era usata una metrica proporzionale alla lunghezza delle code di uscita dei *router*, per fare alcune osservazioni:
    - » La lunghezza (metrica) derivava da una media su un orizzonte (10 s). La durata dell'orizzonte è critica:
      - Corta: troppi transienti;
      - Lunga: rete converge lentamente;
      - La durata ottima non è omogenea sulla rete: dipende dalle capacità dei *link*

6.76

Instradamento

Metriche e dinamica

» La dinamica del costo non deve essere alta: altrimenti alcuni percorsi vengono completamente ignorati

» La lunghezza della coda è usata come "predittore" della situazione futura del *link*: ma linee con code lunghe non verranno scelte nel futuro e quindi si "scaricheranno" (specialmente quelle ad alta capacità) e viceversa.

» La mancanza di restrizioni fra valori successivi dei costi può generare oscillazioni significative.

» Il ricalcolo quasi-sincrono delle tabelle tende a raccogliere traffico su alcune linee.

6.77

Instradamento

Metriche e dinamica

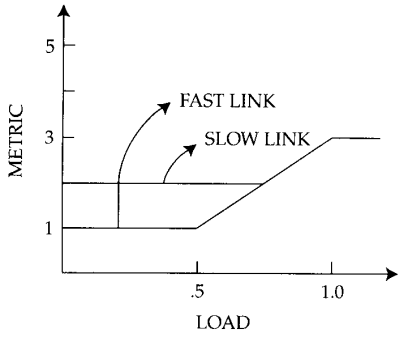
- La soluzione scelta per ARPAnet è stata:
  - Metrica mista capacità-coda dove a carico basso prevale la capacità, carico alto la coda.
  - Costi con una dinamica ridotta: valori da 1 a 3.
  - Massima variazione permessa fra due successivi ricalcoli: 1/2.

6.78

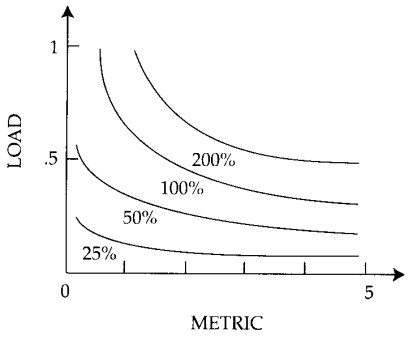
Instradamento

Metriche e dinamica

Mappa della metrica



Risposta della rete

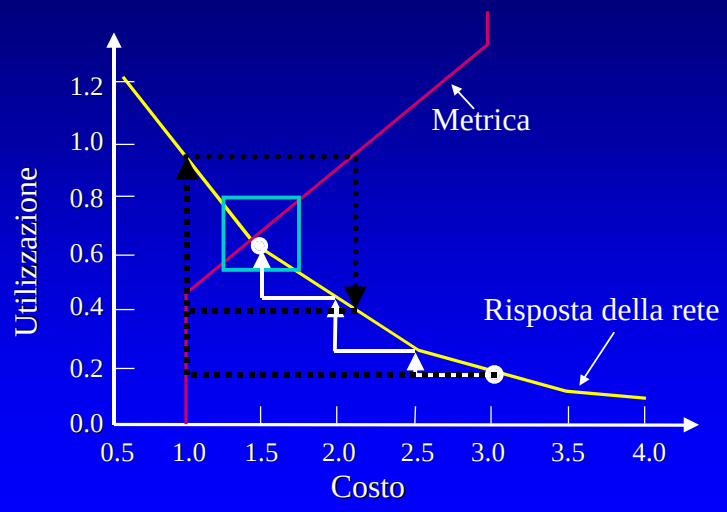


Carico medio in funzione del valore del costo su una singola linea

6.79

Instradamento

Metriche e dinamica



6.80

Instradamento

DVR e LSR

- Il confronto DV *Routing* (DVR) e LS *Routing* (LSR) è complesso:
  - In genere si tende a supporre che gli LSR convergano più rapidamente dei DVR, in pratica la velocità di convergenza dipende molto dalla topologia della rete e dalle caratteristiche del traffico.
  - I DVR sono non escludono la presenza di cicli a priori, ma con le opportune modifiche gli possono evitare efficacemente.
  - Gli LSR, per contro, sono più complessi, devono fare uno sforzo significativo per mantenere i DB congruenti generando anche un traffico di controllo più elevato ed hanno RT più grandi.
  - Gli LSR possono usare più metriche diverse contemporaneamente.

6.81

Instradamento

Gerarchia

Ci sono due ragioni importanti per le quali nelle reti di una certa dimensione si tende ad usare meccanismi di instradamento gerarchici:

- **La scalabilità**
  - Per un numero di nodi elevato (WAN), indipendentemente dal tipo di algoritmo, la complessità dell'instradamento e la dimensione delle RT diventano comunque eccessive.
  - Per esempio, nel caso LS, con tanti archi quanti nodi, si ha una complessità di circa O(NlogN) ed una RT con con dimensione O(N), quindi

# nodi	RT	Calcoli
1000	1000	O(3000)
1.000.000	1.000.000	O(6.000.000)

- **L'autonomia amministrativa**

6.82

Instradamento

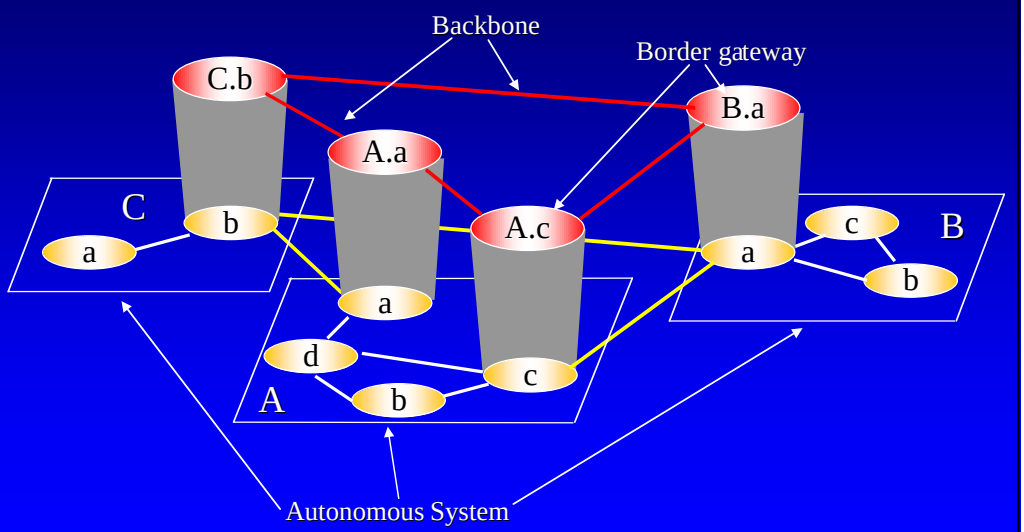
Gerarchia

- Internet distingue tre livelli gerarchici principali:
  - Sottoreti o singoli domini di *broadcast*
    - » All'interno dei quali l'instradamento fa uso dell'ARP.
  - *Autonomous System* (AS)
    - » In cui i protocolli di instradamento prendono il nome di ***Interior Gatway Protocol*** (IGP) e sono: RIP, IGRP e OSPF.
  - *Backbone*
    - » In cui i protocolli di instradamento prendono il nome di ***Exterior Gateway Protocol*** (EGP) e sono: EGP e BGP.

6.83

Instradamento

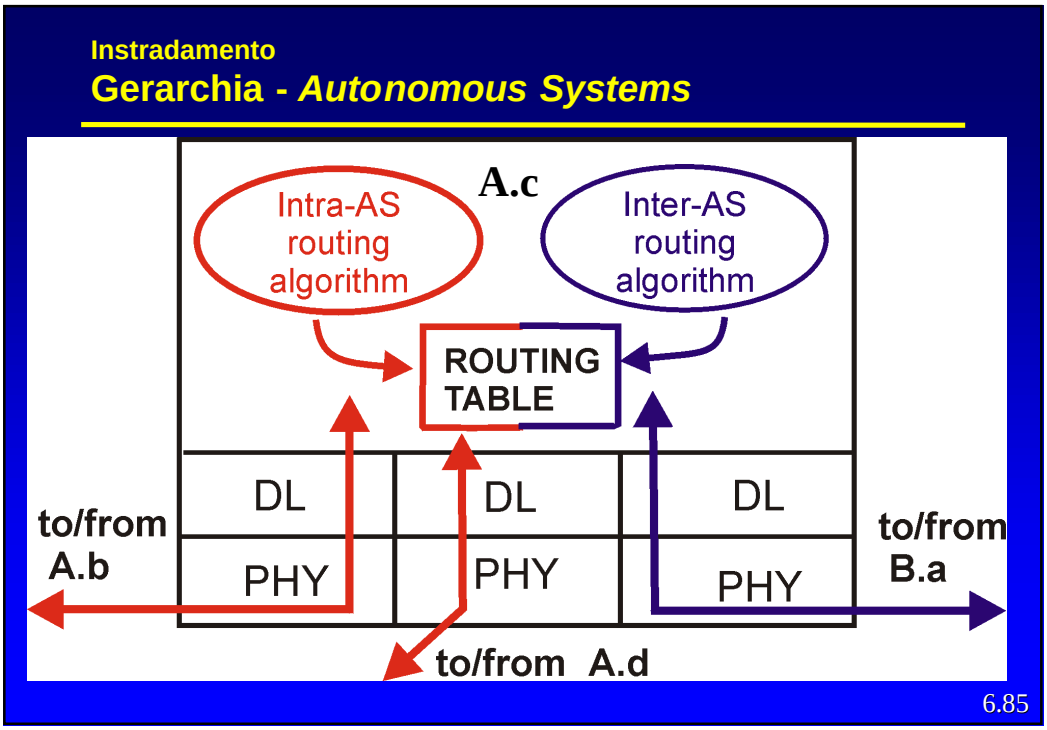
Gerarchia - *Autonomous Systems*



6.84

Instradamento

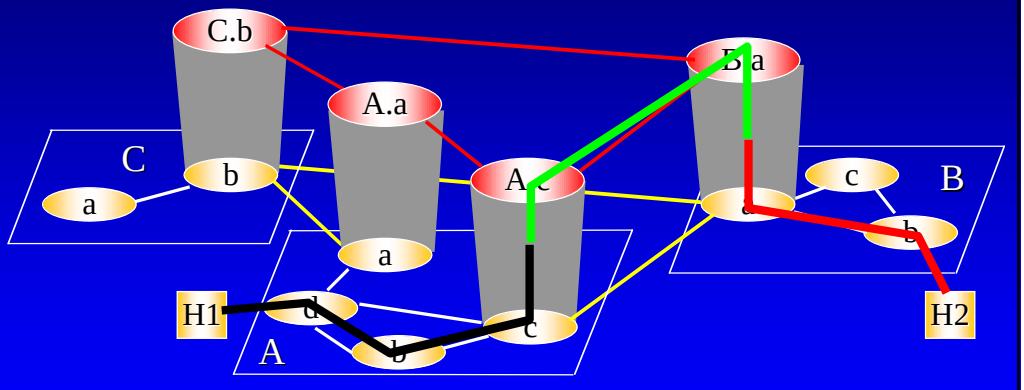
Gerarchia - *Autonomous Systems*



6.85

Instradamento

Gerarchia - *Autonomous Systems*



6.86

Instradamento
Gerarchia

- Il partizionamento è realizzato grazie alla gerarchizzazione degli indirizzi.
- Per area ogni livello si hanno "pochi" nodi.
- Potenzialmente ogni livello può usare algoritmi diversi
- La gerarchia non è stretta ossia il collegamento un area di un livello e il livello superiore può avvenire tramite più nodi (detti *Border Gatway*, BG)
- Ci sono dei router che partecipano all'instradamento di livelli differenti.
- Alcuni indirizzi possono non essere omogenei con lo spazio di indirizzamento dell'area/livello (questo diminuisce l'efficacia della gerarchia).

6.87

Instradamento
Gerarchia

- I diversi livelli non possono nascondersi reciprocamente tutte le informazioni:
  - Ad es., per poter calcolare l'instradamento più opportuno, un nodo del livello 3 deve conoscere i costi per raggiungere i nodi del livello superiore.
  - Allo stesso modo, un nodo di livello 4 deve conoscere i costi verso i nodi del livello 3.
  - Queste conoscenze sono fornite tramite LSP particolari (detti *external records* e *summary records*) che contengono solo le destinazioni ed i costi per raggiungerle (non la topologia). In pratica le reti dei livelli superiori/inferiori vengono rappresentate come se i loro nodi fossero direttamente collegati ai BG.

6.88

Instradamento

Routing Informatio Protocol (RIP)

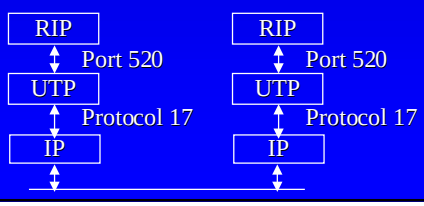
- E' un IGP originariamente progettato dalla Xerox per la propria rete, sviluppato dall'Università di Berkley per la propria implementazione di TCP/IP, e standardizzato con RFC 1058 nel 1988.
- E' un DVR, usa una metrica statica:il costo di un percorso è il numero di hop ossia di linee che esso attraversa.
- Utilizza lo *split horizon with poisonous reverse*, e *i triggered update*.
- Aggiorna la RT ogni 30 s. e elimina ogni vettore non aggiornato per 180 s consecutivi

6.89

Instradamento

Routing Informatio Protocol (RIP)

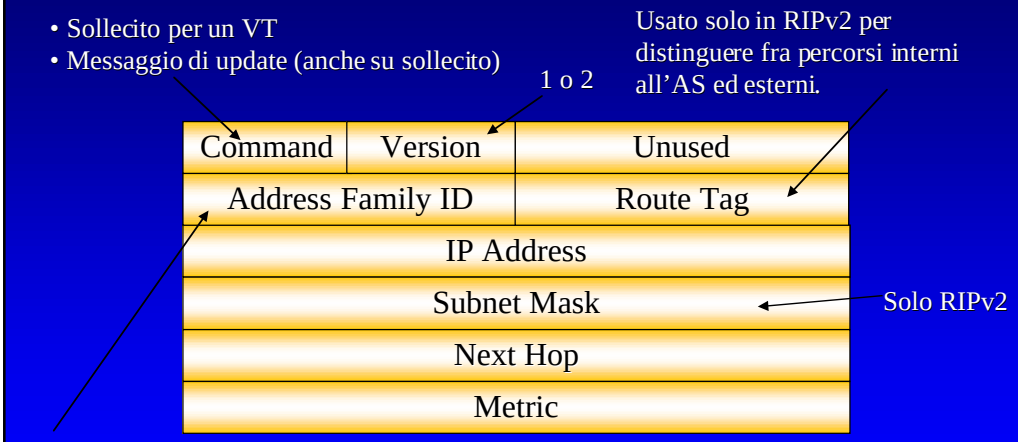
- Ha come valore massimo del costo 15, 16 corrisponde ad infinito. Quindi non permette reti con percorsi con più di 15 *router* attraversati.
- La limitazione di cui sopra è legata al fatto che per reti più grandi è troppo lento a convergere (non alla dimensione del campo costo).
- Ne esistono due versioni, la seconda (RIPv2) consente l'uso del CIDR.
- Lo scambio di informazioni avviene attraverso un protocollo di livello 4 (UTP)



6.90

Instradamento

Routing Informatio Protocol (RIP)



6.91

Instradamento

Interior gateway Routing Protocol (IGRP)

- E' nuovamente un DVR, ma di tipo proprietario; infatti è stato sviluppato dalla CISCO verso la metà degli anni '80 ed è disponibile solo sui suo prodotti.
- Usa una metrica dinamica e sofisticata (considera ritardo, banda, affidabilità, lunghezza del pacchetto ed il carico).
- Permette la suddivisione del carico su più linee (multipercorso).

6.92

Instradamento

Open Shortest Path First (OSPF)

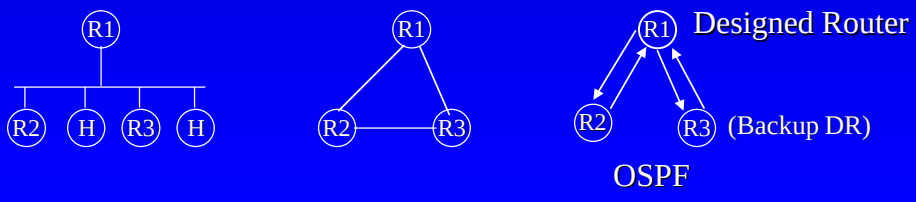
- Nasce nel 1990 con l'RFC 1247 per sostituire il RIP
- E' stato realizzato per rispondere a diverse esigenze:
  - "Open" ossia aperto e non proprietario
  - Multi-metrica (anche dinamiche)
  - Capace di autoadattarsi a cambi topologici
  - *Rounting* diversi per tipi di servizi differenti
  - Bilanciamento dei flussi (multi-percorso)
  - Gerarchico
  - Protezione da aggiornamenti fallaci
  - *Tunneling*

6.93

Instradamento

Open Shortest Path First (OSPF)

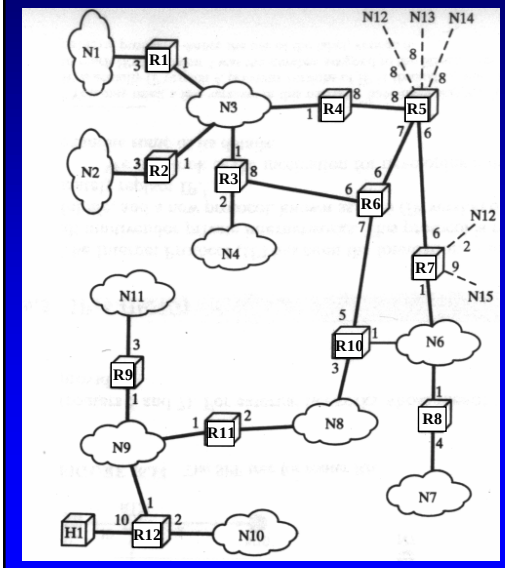
- Supporta tre tipi di connessione e reti
  - Punto - punto fra router
  - Reti multiaccesso con *broadcast* (LAN)
  - Reti multiaccesso senza *broadcast* (WAN a pacchetto)
- Nel caso di LAN a cui sono connessi più router identifica un *router* di riferimento (*designed router*) per ridurre il traffico di LSP sulla LAN.

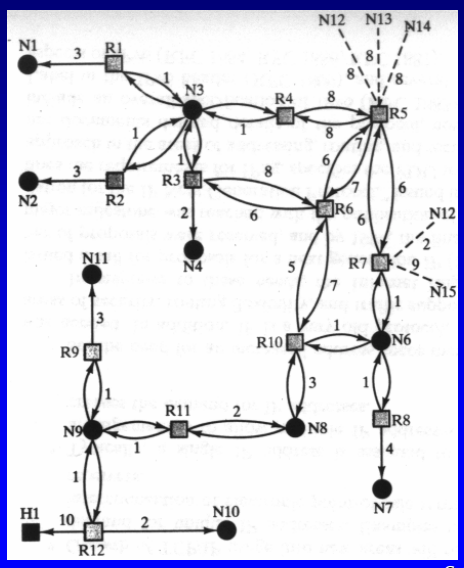


6.94

Instradamento

Open Shortest Path First (OSPF)





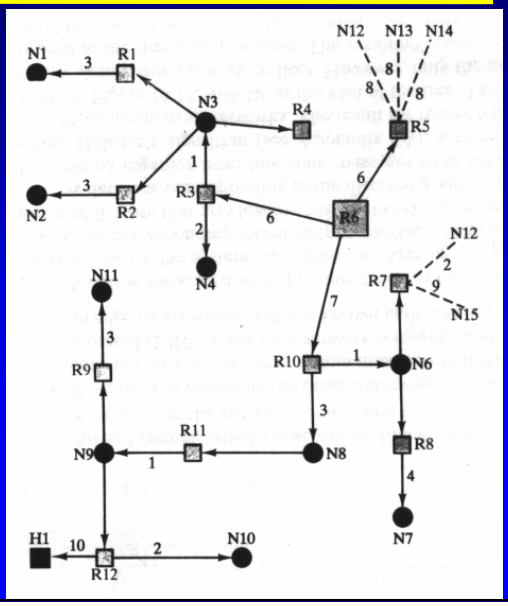
6.95

Instradamento

Open Shortest Path First (OSPF)

TABLE 16.5 Routing Table for RT6.

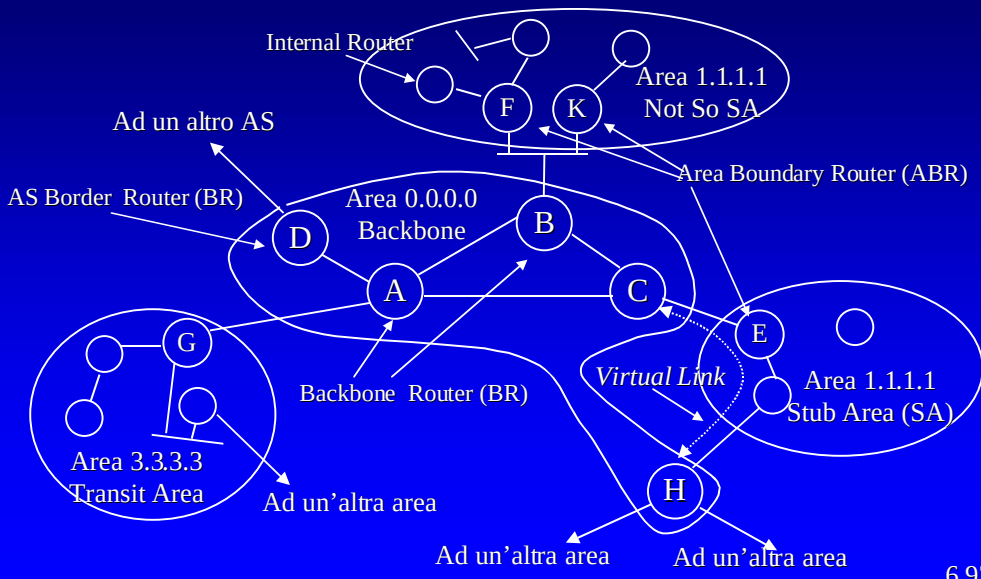
Destination	Next hop	Distance
N1	RT3	10
N2	RT3	10
N3	RT3	7
N4	RT3	8
N6	RT10	8
N7	RT10	12
N8	RT10	10
N9	RT10	11
N10	RT10	13
N11	RT10	14
H1	RT10	21
RT5	RT5	6
RT7	RT10	8
N12	RT10	10
N13	RT5	14
N14	RT5	14
N15	RT10	17



6.96

Instradamento

Open Shortest Path First (OSPF)



6.97

Instradamento

Open Shortest Path First (OSPF)

- Le *Stub Area* non propagano informazioni interne o esterne ed accedono al *backbone* tramite un *router* di *default*.
- L'instradamento fra due aree viene realizzato in tre parti:
  - Il percorso nell'area sorgente fra la sorgente stessa ed un *Area Border Router*.
  - Il percorso fra le due aree tramite il backbone
  - Il percorso nell'area destinazione fra l'ABR che riceve il pacchetto dal *backbone* e la destinazione.
- In pratica si forza un instradamento a stella in cui il backbone rappresenta il centro stella.

6.98

Instradamento

EGP- "EGP"

- Al più vecchio dei protocolli EGP è stato assegnato lo stesso nome che distingue la categoria: EGP.
- E' un protocollo di stile DV che però non propaga costi ma solo informazioni di raggiungibilità.
- Non è in grado di evitare cicli e quindi non può essere usato in tipologie magliate ma solo ad albero.
- La sua struttura di riferimento è composta da "*Core Router*" (CR) collegati fra loro ad albero.
- Ogni AS può essere collegato ad un unico CR e quindi ogni CR fa da centro stella per un gruppo di AS

6.99

Instradamento

EGP - Border Gateway Protocol (BGP)

- E' il protocollo EGP relativamente recente, definito dal RFC 1654.
- La versione in uso attualmente è la 4 (BGP4).
- Permette la cooperazione fra *router* di AS diversi (chiamati *gateway*) per la realizzazione dell'instradamento fra AS.
- Per lo scambio di informazioni fra i nodi usa il TCP (porta 179).

6.100

Instradamento

EGP - Border Gateway Protocol (BGP)

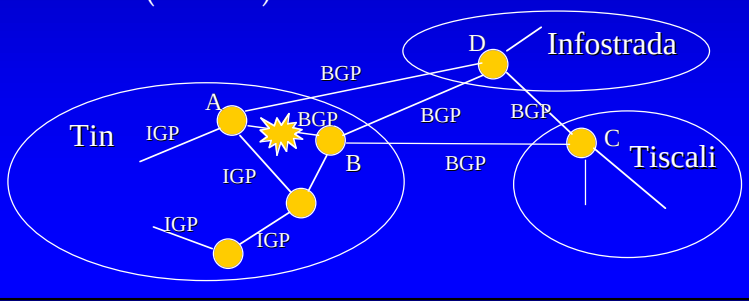
- Opera in tre passi:
  - Identificazione dei nodi adiacenti (*neighbor*)
  - Raggiungibilità dei nodi adiacenti
  - Raggiungibilità delle reti
- Utilizza un algoritmo DV, ed in particolare usa un *Path Vector*.
- Distingue tre tipi di reti
  - *Stub:* che hanno un'unica connessione con il *backbone* e non possono venir usate come transito
  - *Multiconnected*: che potenzialmente possono essere usate per transito (se lo permettono)
  - *Transit*: costruite per realizzare il transito.

6.101

Instradamento

EGP - Border Gateway Protocol (BGP)

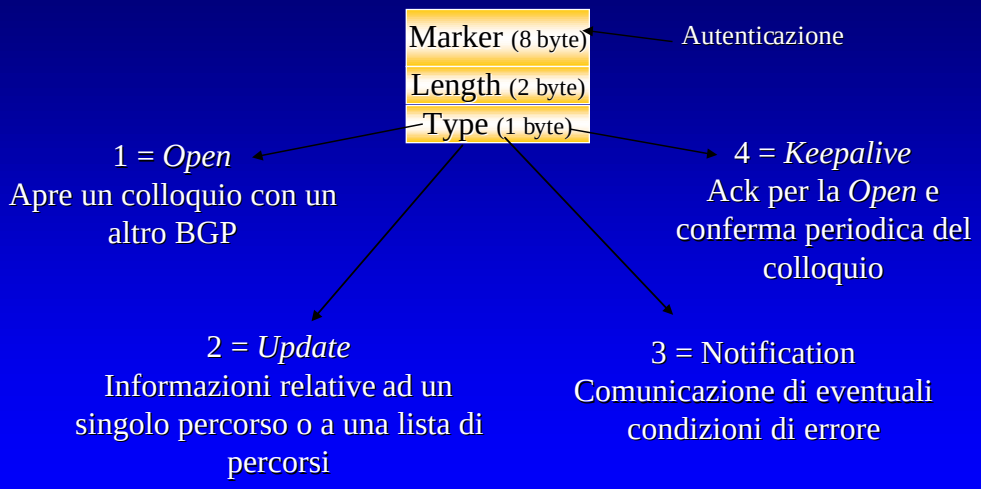
- Anche se tutti i *router* all'interno di un AS sono fra loro cooperativi, non è detto che i nodi che interconnettono due AS si "fidino" l'uno dell'altro.
- Questo accade in quanto gli AS sono in genere controllati o posseduti da organizzazioni diverse e quindi l'instradamento deve dipendere anche dagli accordi fra i diversi AS (transito).



6.102

Instradamento

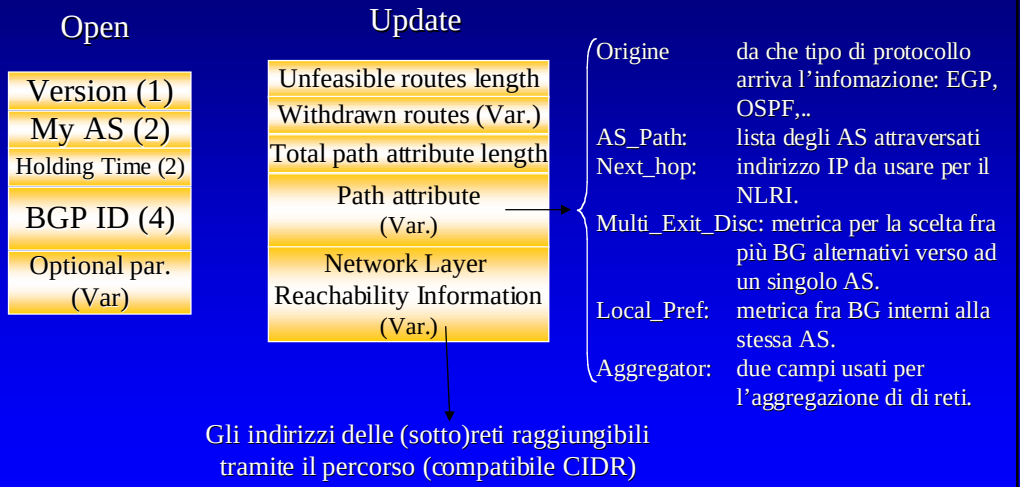
EGP - Border Gateway Protocol (BGP)



6.103

Instradamento

EGP - Border Gateway Protocol (BGP)



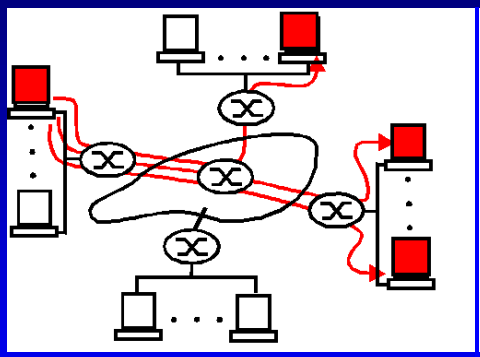
6.104

Instradamento
Multicast

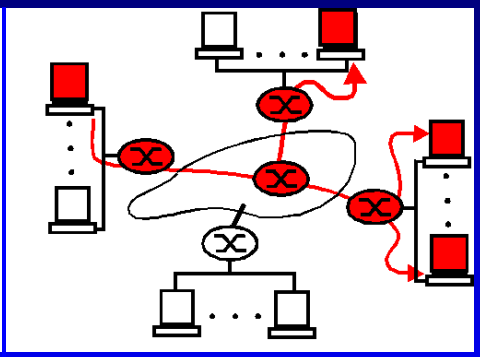
- Si tratta di dare supporto alle trasmissioni in cui un singolo pacchetto debba venir consegnato a molte destinazioni.
- Esempi di casi in cui è richiesto un *multicast* sono molti e coinvolgono
  - Trasferimenti dati (ad es. aggiornamenti di software)
  - *Broadcast* media (audio, video, testo)
  - Applicazioni condivise (*whiteboard*, teleconferenza, …)
  - Aggiornamento di dati (quotazioni di borsa)
  - Giochi interattivi
  - Localizzazione di risorse (Server, stampanti, …).

6.105

Instradamento
Multicast



Multicast realizzato tramite invio di più pacchetti *unicast*

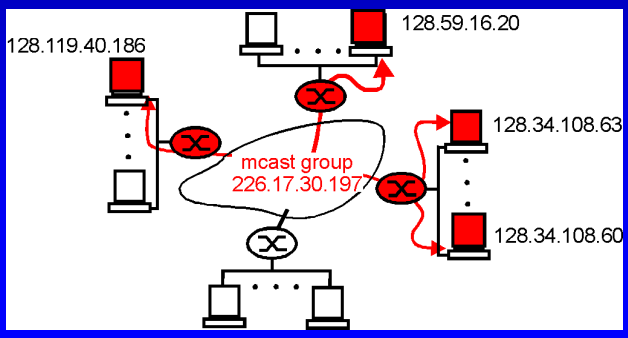


Multicast realizzato con invio di un singolo pacchetto duplicato nei *router*

6.106

Instradamento
Multicast

- Se si usa un singolo pacchetto duplicato all'occorrenza dai *router*, bisogna che tale pacchetto trasporti l'indirizzo di tutte le destinazioni.
- In alternativa si può astrarre l'indirizzo dalla destinazioni e creare un indirizzo del gruppo (o indirizzo *multicast*)



6.107

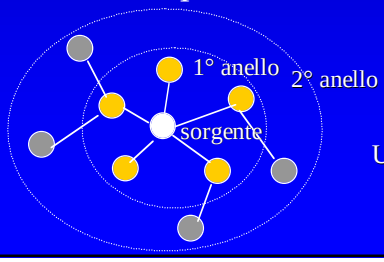
Instradamento
Multicast

- Elemento base del *multicast* è quindi il concetto di **gruppo**
  - Rappresenta l'associazione fra un insieme di trasmettitori ed un insieme di ricevitori. Ogni ricevitore del gruppo riceve i pacchetti inviati ad qualunque trasmettitore del gruppo stesso.
  - Il gruppo concettualmente esiste indipendentemente dalla presenza di elementi componenti.
  - Nella pratica un gruppo nasce nel momento in cui il primo elemento si aggrega e termina la sua esistenza quando tutti gli elementi si sono dissociati.
- Due sono quindi gli aspetti significativi:
  - **La gestione dei gruppi**
  - **La disseminazione dell'informazione (*routing* vero e proprio.)**

6.108

Instradamento
Multicast

- Si osservi che la potenza del *muticast* è anche legata proprio al disaccoppiamento fra trasmettitori e ricevitori, che permette ad un trasmettitore di localizzare il ricevitore senza conoscerne l'indirizzo specifico.
- Per esempio, un calcolatore che cerca un *print server*, può richiederlo tramite un "*well-know*" indirizzo multicast e utilizzare un meccanismo di tipo "*expanding ring*" per identificare il *server* più vicino.



Usando il *Time To Live* (TTL) si ricerca in anelli di dimensioni crescenti

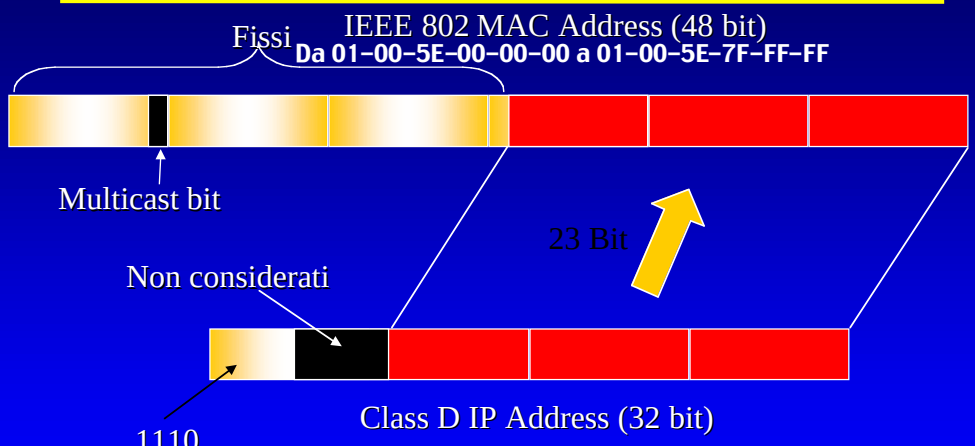
6.109

Instradamento
Multicast

- Per quanto concerne la realizzazione del *multicast*, si possono distinguere due ambiti:
  - *Broadcast* LAN (quindi senza attraversare *router*)
  - WAN (fra *router*)
- Le LAN IEEE 802 prevedono un indirizzo di *multicast* che viene utilizzato in modo diretto nel dominio di collisione e, anche tramite VLAN, nel dominio di *broadcast*.
- La gestione della registrazione degli utenti può avvenire a livello di VLAN tramite GMRP (*Multicast Registration Protocol*)

6.110

Instradamento
Multicast



- L'indirizzo 224.0.0.0 non viene assegnato a nessun gruppo
- 224.0.0.1 è il gruppo permanente composto da tutti gli host (che rispondono al *multicast*) su una LAN
- 224.0.0.2 è il gruppo di tutti i *router* su una LAN

6.111

Instradamento
Multicast - IGMP

- ***Internet Group Management Protocol*** (IGMP) è usato dai *Multicast router (Mrouter)* per la gestione dei gruppi *multicast* di *host* sulle B-LAN.
- La versione attuale dell'IGMP è la numero 2
  - esistono installazioni della 1
  - la numero 0 è obsoleta
- I messaggi IGMP vengono spediti in *multicast* (indirizzo 224.0.0.1 e TTL = 1) per tutti gli *host* sulla LAN e sono incapsulati in un *datagram* IP con campo *protocol* uguale a 2

6.112

Instradamento

Multicast - IGMP

- Un M-router designato sulla LAN invia periodicamente (1 al minuto, al massimo) dei pacchetti di "*query*" a cui ogni *host* interessato risponde in *broadcast* con un elenco dei gruppi a cui ha aderito o vuole aderire.
- Gli *host* rispondono alle *query* generando dei *report*, con cui segnalano all' *Mrouter* tutti gli *host group* a cui appartengono
- In particolare, inviano un *report* per ciascun *host group* a cui sono iscritti

6.113

Instradamento

Multicast - IGMP

- Si osservi che all' *Mrouter* non interessa in numero totale di elementi che aderiscono ad un gruppo, ma solo se ce né almeno uno.
- Per evitare troppo traffico e collisioni, quando un *host* A riceve una *query*, ritarda la propria trasmissione di un tempo casuale. Se prima che abbia trasmesso A, un altro nodo B trasmette segnalando la propria adesione agli stessi gruppi di interesse di A, A non trasmette più.
- Per aderire ad un gruppo, un *host* deve
  - configurare la propria interfaccia di rete per riceve un dato indirizzo *multicast*.
  - Se una altro *host* ha già richiesto di aderire non deve fare altro, altrimenti deve attendere una *query*.

6.114

Instradamento

Multicast - IGMP

- ***Type***
  - 0x11 = *Host Membership Query*: inviate dall' *Mrouter* verso gli *host*, per tenere aggiornata la lista degli *host group* attivi sulla LAN
  - 0x16 = *Host Membership Report*: inviate dall'*host* in risposta alle query del *router*
  - 0x17 = *Leave Group*: inviato (opzionalmente) agli *Mrouter* da un *host* per annuciare l'abbandono di un gruppo quando sial'unico membro

6.115

Instradamento

Multicast - IGMP

- *Max Resp Time*
  - Usato per *Membership Query*: Massimo tempo entro cui deve essere inviata la risposta, se contiene un
    - » Valore piccolo: i *router* sono aggiornati più velocemente sullo stato dei gruppi
    - » Valore grande: i *report* sono più sparsi nel tempo, minore *burstiness*
- *Group Address*
  - Viene impostato a zero nelle *query* generali per scoprire quali gruppi operano sulla LAN
  - Nei *report* e nelle *query* specifiche contiene l'indirizzo dell'*host group* a cui appartiene un *host*
- La versione 2 del protocollo prevede un meccanismo di designazione per il *router multicast*, la versione 3 (*draft*) prevede anche la possibilità di selezionare la sorgente.

6.116

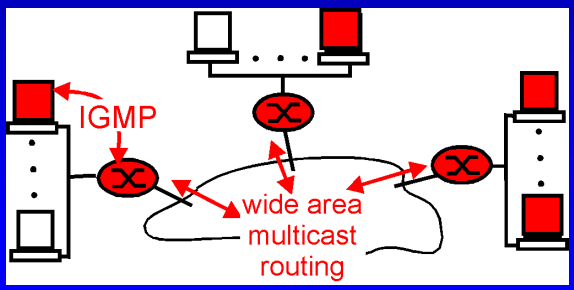
Instradamento
Multicast

- Alcune osservazioni
  - La creazione di un gruppo in Internet è una operazione "***receiver-driven***"
  - La sorgente non partecipa alla formazione del gruppo e quindi non ne può neppure controllare la composizione
  - Allo stesso modo non c'è controllo su chi invia al gruppo:
    - » Sovrapposizione di invii allo stesso gruppo
    - » Uso dello stesso indirizzo con sovrapposizioni di invii fra gruppi diversi
    - » Invii di trasmissioni di disturbo volute
  - Dal punto di vista della sicurezza il problema va affrontato a livello di applicazione

6.117

Instradamento
Multicast - WAN

- Una volta che i *router* sono conoscono la presenza di host ad essi connessi appartenenti a gruppi e sono in grado di inviargli e/o ricevere l'informazione *multicast*, il problema si sposta nel gestire il *routing muticast* fra i *router* (WAN)

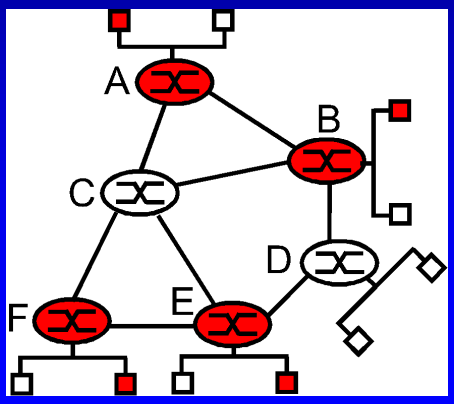


6.118

Instradamento

Multicast - WAN

- In linea di principio il *routing multicast* si realizza individuando uno *spanning tree* che comprenda tutti i nodi interessati ad un gruppo ossia un sottoinsieme di tutti quelli della rete.

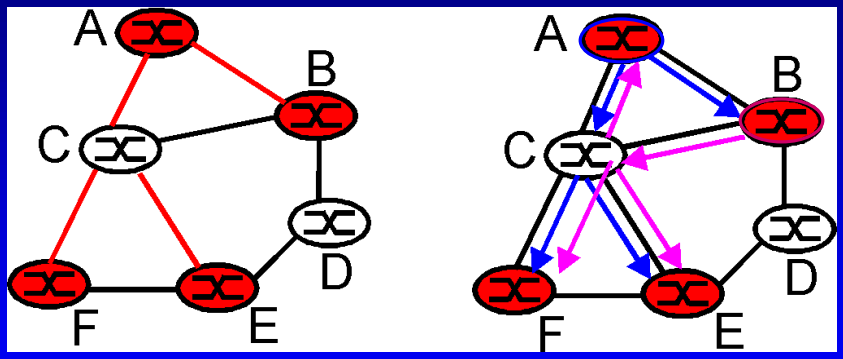


6.119

Multicast

WAN

- Si hanno due possibilità principali:



Group-shared tree
Ossia un unico *spanning tree* per tutti i nodi del gruppo.

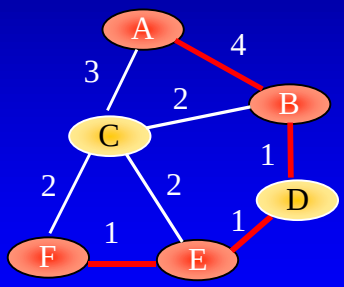
Source-based tree
Ossia uno *spanning tree* diverso per ciascuna sorgente.

6.120

Multicast

WAN - *Group-shared tree*

- ***Group-shared tree***
  - Si tratta di trovare lo *spanning tree* la cui somma dei costi su i *link* che lo compongono sia minima



Il problema di trovare questo albero è noto come "*Steiner Tree Problem*" ed un problema NP-completo.

- Esistono però euristiche che permettono di trovare buone approssimazioni

6.121

Multicast

WAN - *Group-shared tree*

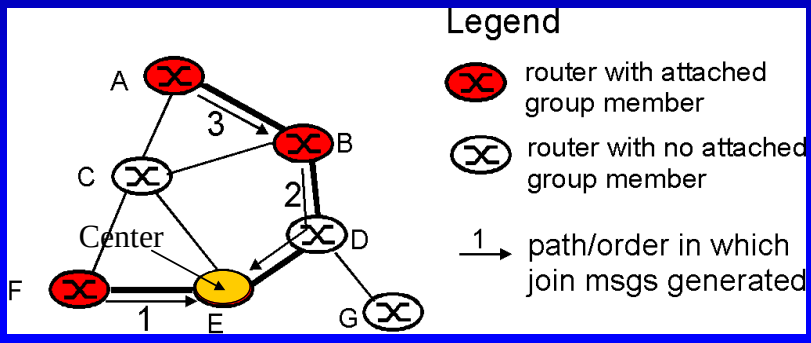
- Nonostante la presenza di calcoli approssimati efficaci, nessun algoritmo su Internet usa questo approccio.
- Questo perché:
  - Bisogna conoscere il costo di ogni *link* sulla rete
  - Si deve ripetere il calcolo ad ogni cambio di costo
  - Non riesce ad usare facilmente le tabelle di *routing* già calcolate per l'*unicast*
  - Ha dei limiti di prestazioni.

6.122

Multicast

WAN - *Group-shared tree*

- Un approccio alternativo in questa categoria è la "*Center Based Tree*", che fa uso di un nodo di riferimento (*center node* o *rendezvous point* o *core*)
- Tutti i *router* con un *host* che aderisce ad un gruppo, inviano un messaggio di *join* lungo il percorso *unicast* verso il nodo di centro
- Fino a che il messaggio o raggiunge il centro o incontra un *router* già parte del gruppo, crea un percorso dell'albero

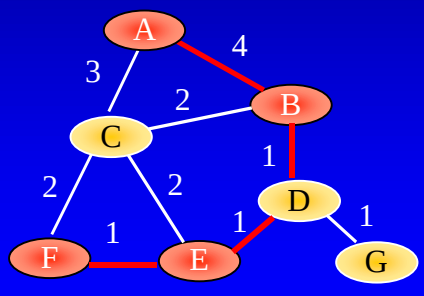


6.123

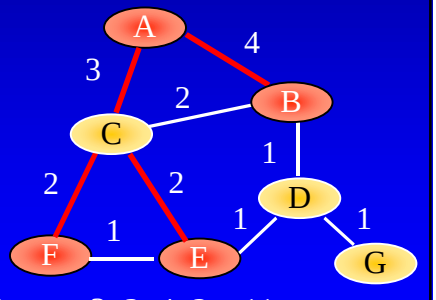
Multicast

WAN - *Source-based tree*

- Gli algoritmi LS (Dijkstra), ricavano, in sostanza, uno *short-path spanning tree* per ogni nodo.
- Depurando l'informazione della rete dei *router* non interessati al gruppo, ogni *router* può calcolare lo *short-path spanning tree* del gruppo per qualunque sorgente.



Costo = 1+1+1+4 = 7
Dist. Media = 5.67 (max= 7)



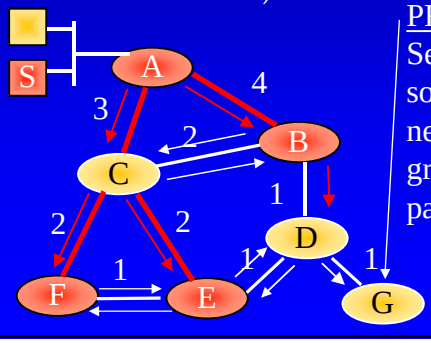
Costo = 3+2+4+2 = 11
Dist. Media = 4.67 (max = 5)

6.124

Multicast

WAN - *Source-based tree*

- Un modo elegante e più semplice per realizzare l'instradamento *multicast* è utilizzare il *Reverse Path Forwarding* (RPF)
- Un pacchetto proveniente dalla sorgente S viene inviato su tutte le uscite (tranne quella da cui è arrivato) solo se arriva dalla interfaccia che corrisponde al percorso più corto verso S (il *next hop* con destinazione S della RT).



PROBLEMA
Se dietro a questo nodo ci sono altri *router*, anche se nessuno di essi fa parte del gruppo, riceveranno tutti il pacchetto

6.125

Multicast

WAN - *Source-based tree*

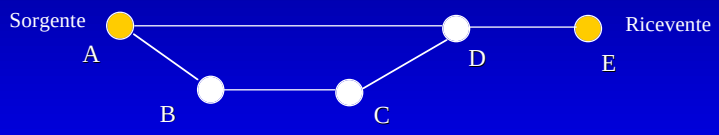
- Un modo efficace per migliorare l' RPF è dato dall'utilizzo della tecnica di *pruning* (potatura), ossia l'esclusione dalla disseminazione dell'informazione dei nodi "foglia" (*leaf*) o terminali non interessati ad un gruppo
- Per far ciò bisogna
  - Identificare le foglie;
    - » Usando un ERPF si può verificare se si è sul percorso a lunghezza minima la sorgente; i nodi che non sono su percorsi sono nodi foglie.
  - Comunicare l'assenza di partecipanti al gruppo;
    - » Inviando messaggio di *pruning* per quel gruppo (sorg.).
- L'eventuale ri-inserimento può avvenire tramite richiesta esplicita (*graft*) o automaticamente legando il *pruning* ad un *timeout*.

6.126

Instradamento

Multicast - WAN

- Si osservi, però, che l'RPF non evita la presenza di copie multiple dello stesso pacchetto e quindi non corrisponde a utilizzare uno ST.
- Un miglioramento (*Extended* RPF) lo si ottiene imponendo che un nodo C il pacchetto verso D solo se il percorso più corto da S (sorgente) a D include C stesso.



- Questo implica però che ogni nodo sappia di essere transito per quella sorgente:
  - *Split horizon* invia infinito ai nodi "*next hop*"
  - L'informazione può essere inviata esplicitamente con i DV (serve un solo bit associato ai vettori verso destinazioni per cui il nodo ricevente è transito).

6.127

IP Multicast

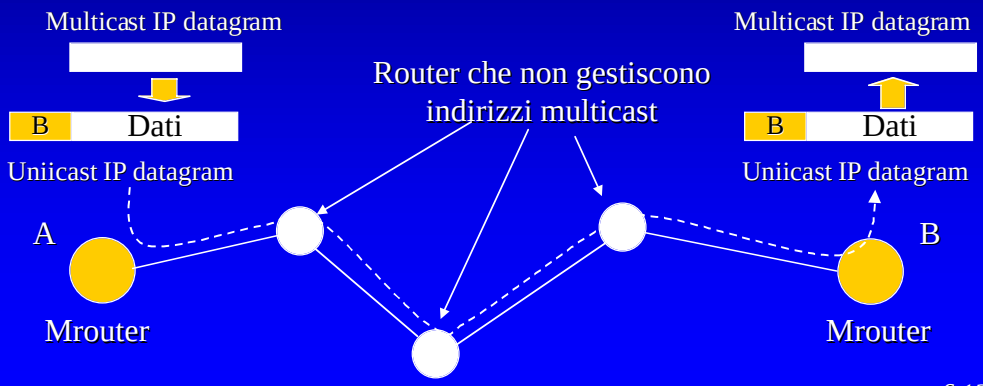
Mbone

- Le funzionalità relative al *multicast* non sono presenti in tutti i *router*
- Per fornire servizi multicast su Internet è stata realizzata una rete virtuale che interconnette tutti gli *Mrouter* che è stata chiamata:
  Multicast Backbone (Mbone)
- Mbone nasce nel 1992 in forma sperimentale e permette a chi ci si connette di realizzare servizi *multicast* su WAN.

6.128

IP Multicast
Mbone

- Se due *Mrouter* non sono direttamente connessi, viene creata fra loro una connessione attraverso un *tunneling*:



6.129

IP Multicast
Mbone

- Ogni *Tunnel* è definito da
  - *Local end-point* (fisso)
  - *Remote end-point* (fisso)
  - *Metric* (dinamico)
    - » Costo del tunneling
  - *Threshold* (dinamico)
    - » Valore minimo del TTL perché il pacchetto possa essere instradato nel tunnel (ogni *Mrouer* decrementa TTL di 1 nel pacchetto *multicast*).

6.130

IP Multicast

Protocolli

- All'interno della rete virtuale di Mbone vengono usati i protocolli di *routing multicast*
- Per quanto concerne gli IGP *multicast*:
  - Protocolli *flood and prune*
    - » ***Distance Vector Multicast Routing Protocol*** (DVMRP)
    - » ***Protocol Indipendent Multicast Dense Mode***(PIM-DM)
  - ***Multicast* OSPF** (MOSPF)
  - Protocolli *Center Based Tree* (CBT)
    - » ***Core Based Tree*** (CBT)
    - » **PIN - *Sparse Mode*** (PIN-SM)

6.131

IP Multicast

Protocolli *flood and prune* - DVMRP

- ***Distance Vector Multicast Routing Protocol*** (DVMRP)
- E' un protocollo DV definito nel RFC 1075
- Ignora le informazioni relative ad altri protocolli (*unicast*) e realizza un proprio DV classico con metrica in numero di *hop* effettuati sulla rete virtuale MBone.
- Usando la propria RT applica un *Reverse Path Forwarding* usando messaggi espliciti per le procedure di *prune,* ossia il traffico fluisce ovunque ed i nodi coinvolti si "potano" esplicitamente.

6.132

IP Multicast

Protocolli *flood and prune* - PIN-DM

- ***Protocol Indipendent Multicast Dense Mode***(PIM-DM)
- E' molto simile al DVMRP ma, a differenza di questi, utilizza la RT dell'instradamento *unicast*.
- Sia il DVMRP che il PIM-DM non sono adatti a operare in modo globale, infatti costringono tutti i nodi non interessati raggiunti a "potarsi".
- Quando l'utenza è distribuita in modo "denso" (per esempio all'interno di una organizzazione) sono molto efficaci.

6.133

IP Multicast

MOSPF

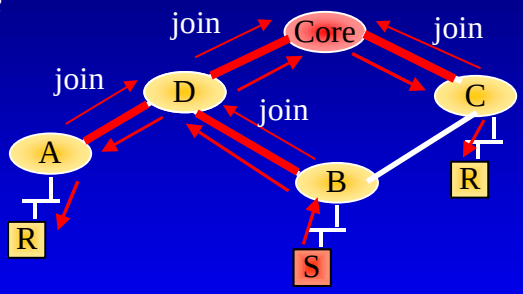
- E' un LS che estende le funzionalità dell'OSPF per la gestione del *multicast*.
- E' definito dall'RFC 1584
- Estende il DataBase dei LS per memorizzare anche i diversi gruppi attivi presso gli altri *Mrouter*.
- Usando il Database esteso ogni *Mrouter* calcola gli ST troncati in modo autonomo.
- E' adatto per gruppi *multicast* a bassa densità, ma è poco scalabile perché richiede, in ogni nodo, della informazione esplicita sui partecipanti ai gruppi.

6.134

IP Multicast

Protocolli *Center Based Tree* - CBT

- Il ***Core Base Tree*** è cronologicamente il primo dei protocolli *Center Based Tree*



- *IL CBT* costruisce un albero bidirezionale perché i pacchetti possono viaggiare sia in direzione core che nella direzione opposta, a seconda della posizione della sorgente

6.135

IP Multicast

Protocolli *Center Based Tree* - CBT

- La sorgente non deve necessariamente appartenere al *Tree*, in ogni caso il pacchetto viene inviato verso il *Core*, il primo nodo dell'albero che raggiunge, viene propagato sull'albero stesso.
- Può esserci più di un *Core*
- I limiti sono
  - posizionare il *Core* opportunamente è difficile e, se il *Core* non è ben posizionato, l'albero è inefficiente
  - Non si ha un metodo consolidato per legare l'indirizzo del *Core* e quello del gruppo
- Pregi
  - Efficiente per quanto concerne lo stato da mantenere nei *router*, solo informazione sulle porte di *forwarding* per il gruppo e nessuna informazione sulle sorgenti
  - Si scala meglio dei *flood and prune* su gruppi sparsi

6.136

IP Multicast

Protocolli *Center Based Tree* - PIM *Sparse Mode*

- Nel PIM *Sparse Mode* il nodo di riferimento si chiama *Rendezvous Point* invece che *Core*, ma ha le stesse funzioni.
- Un *receiver* che voglia aggregarsi ad un gruppo manda un messaggio di *join* al RP. I *router* che il messaggio incontra registrano la presenza del percorso *multicast* creando un albero ma unidirezionale, ossia dal RP verso i ricevitori
- Una sorgente invece invia il pacchetto al *router*, che lo incapsula in un altro pacchetto *unicast* e lo invia al RP. Il RP lo estrae e lo invia sull'albero del gruppo corrispondente

6.137

IP Multicast

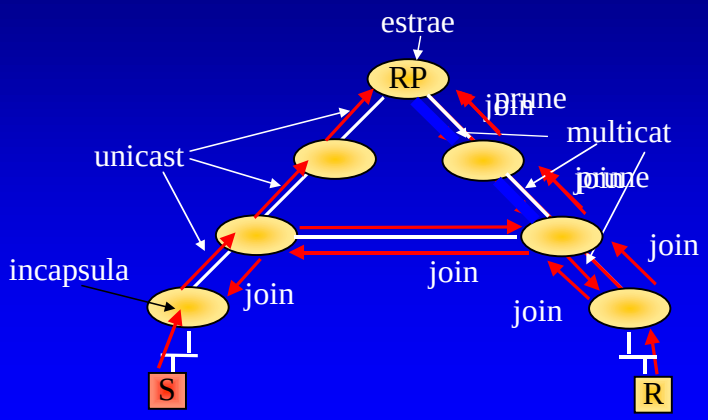
Protocolli *Center Based Tree* - PIM *Sparse Mode*

- Questo significa che la distribuzione dei pacchetti inizia sempre dal RP e che l'albero di distribuzione è unidirezionale.
- La destinazione ha però la possibilità di cambiare l'albero di distribuzione:
  - Quando riceve i primi pacchetti dalla sorgente, invia alla sorgente stessa un messaggio di *join* diretto, che ovviamente raggiunge la sorgente lungo lo *Shortest path*
  - ogni *router* attraversato dal *join* attiva il *forwarding* per quel gruppo-sorgente.
  - Quando la destinazione comincia a ricevere i pacchetti dal nuovo percorso, invia un *prune* sull'albero principale
  - Se tutte le destinazioni fanno la stessa operazione, per quella sorgente si crea uno *Shortest Path Spanning Tree* di distribuzione per il gruppo

6.138

IP Multicast

Protocolli *Center Based Tree* - PIM *Sparse Mode*



6.139

IP Multicast

Protocolli *Center Based Tree* - PIM *Sparse Mode*

- L'eventuale inefficienza dell'albero creato con il RP, viene corretta creando alberi esplicitamente.
- D'altro canto sorgenti che generano flussi ridotti non aumentano eccessivamente il carico di informazioni di stato perché possono usare l'albero del RT.
- Esiste un meccanismo per la scelta degli RT di riferimento che riduce la scalabilità dell'algoritmo che resta adatto anche per domini ampi ma non enormi.

6.140

IP Multicast

IP Multicast - EGP

- Tutte le tecniche viste hanno dei limiti di scalabilità, che non le rende adatte ad essere applicate in ambito multi dominio.
- Inoltre fra domini diversi spesso ci sono *router* che non supportano il *multicast*.
- Al momento, in ambito EGP, esiste una soluzione che vede l'uso di due protocolli:
  - **Multiprocol Extension for BGP4 (MBGP)**
  - **Multicast Source Discovery Protocol (MSDP)**

6.141

IP Multicast

IP Multicast - MBGP

- E' una estensione del BGP che permette di costruire ed aggiornare tabelle di *routing* multiple.
- Questa caratteristica permette di mantenere una tabella separata che costruisca una connettività per i *router* con capacità di *multicast*
- Tale tabella può essere sfruttata da algoritmi PIM per inviare messaggi di *join*.

6.142

IP Multicast

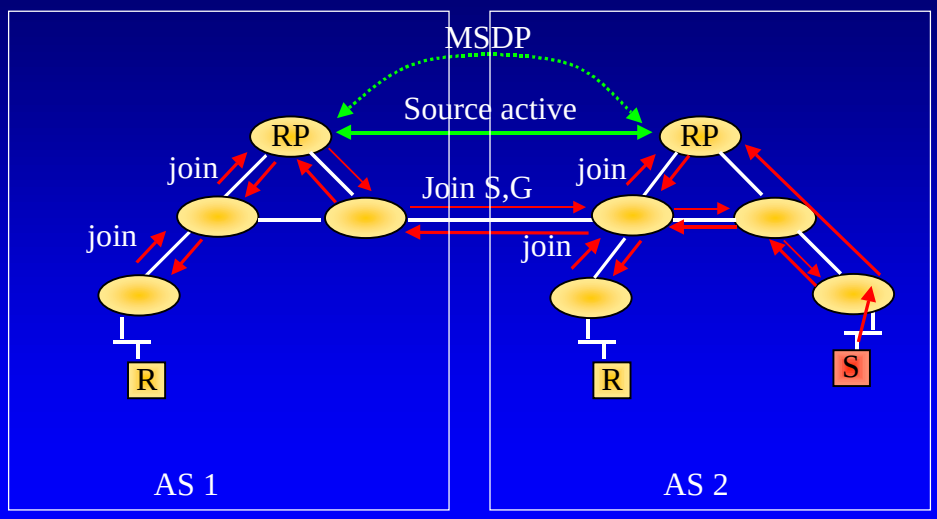
IP Multicast - MSDP

- Anche usando il PIM-SM, domini diversi in genere non vogliono dipendere da RP che non siano al loro interno.
- Questo protocollo permette, usando anche il MBGP, di far dialogare RP in domini diversi per creare alberi misti.

6.143

IP Multicast

IP Multicast - MSDP



6.144

IP Multicast

IP Multicast - *Scope*

- Il *multicast* consuma molte risorse, quindi è importante limitarne la dispersione ("*scope*") sia per ragioni di prestazioni sia per sicurezza (mandare molti multicast può essere un modo per bloccare una rete); inoltre permette il riutilizzo di indirizzi
- Due modalità
  - TTL, si fissa una soglia al di sotto della quale il pacchetto multicast non viene propagato dei Border *Router* in Europa la soglia è 64)
  - Amministrativa: difficile da gestire specialmente in presenza di aree sovrapposte.

6.145