Reti di Calcolatori

Livello di rete: protocolli di supporto

Università degli studi di Verona
Dipartimento di Informatica

Docente: Damiano Carra

Acknowledgement

❑ *Credits*

- *Part of the material is based on slides provided by the following authors*
  - *Douglas Comer, "Computer Networks and Internets," 5th edition, Prentice Hall*
  - *Behrouz A. Forouzan, Sophia Chung Fegan, "TCP/IP Protocol Suite," McGraw-Hill*, January 2005

Argomenti trattati

- ❑ Error reporting ➔ protocollo ICMP
- ❑ Bootstrapping ➔ protocollo DHCP

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol

- Al protocollo IP e' stato associato un protocollo complementare: ICMP
  - usato principlamente per inviare messaggi di errore alla sorgente in caso di problemi
- IP e ICMP dipendono l'uno dall'altro
  - IP dipende da ICMP per segnalare eventuali errori
  - e ICMP utilizza IP per trasportare i messaggi di errore
- Sono stati definiti molti messaggi ICMP

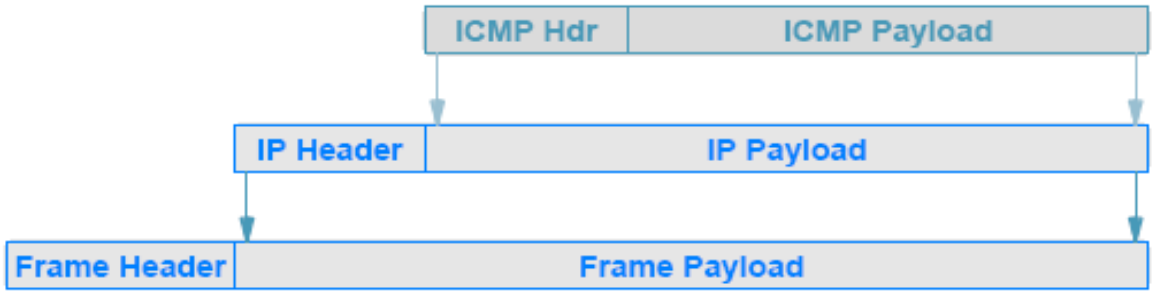
Internet Control Message Protocol

Number	Type	Purpose
0	Echo Reply	Used by the ping program
3	Dest. Unreachable	Datagram could not be delivered
5	Redirect	Host must change a route
8	Echo	Used by the ping program
11	Time Exceeded	TTL expired or fragments timed out
12	Parameter Problem	IP header is incorrect
30	Traceroute	Used by the traceroute program

Internet Control Message Protocol

- ICMP contiene due tipi di messaggi:
  - messaggi per la segnalazione di errori
    - ad es., Time Exceeded and Destination Unreachable
  - messaggi per ottenere informazioni
    - ad es., Echo Request and Echo Reply
- Echo Request/Reply sono usati dall' applicazione "ping" per testare la connessione
  - Quando un host riceve un messaggio di "echo request"
    - il software ICMP sull' host o sul router invia un messaggio di "echo reply" che trasporta gli stessi dati del messaggio di richiesta

ICMP: formato dei messaggi e trasporto



- ICMP usa IP per trasportare i propri messaggi:
  - quando un router ha un messaggio ICMP da inviare
    - crea un datagramma IP e mette nel payload il messaggio ICMP
  - il datagramma viene poi inviato come al solito

ICMP: formato dei messaggi e trasporto

- I messaggi ICMP non hanno una priorita' particolare
  - sono inviati come ogni altro datagramma con una sola eccezione
- Se un messaggio ICMP di errore causa un errore
  - non viene inviato nessun messaggio di errore
- Il motivo e' chiaro:
  - rischio di effetto valanga

Dynamic Host Configuration Protocol (DHCP)

Software di Protocollo: parametri e configurazione

- Quando un host o un router vengono accesi, il sistema operativo e il software che gestisce il protocollo di rete vengono inizializzati
  - in che modo il software di rete (di un host o di un router) inizia ad operare?
- Chi gestice un router deve specificare dei valori iniziali, tra cui
  - l'indirizzo per ciascuna connessione di rete (interfaccia)
  - quale software di protocollo utilizzare
  - i valori iniziali delle tabelle di forwarding

  ➔ Tale configurazione viene salvata e caricata dal router durante lo startup
- La configurazione degli host di solito segue un processo noto come bootstrapping
  - E' stato ideato un protocollo per permettere ad un host di ottenere una serie di parametri con una singola richiesta (Bootstrap Protocol, BOOTP)
  - Attualmente DHCP viene usato per fornire la maggior parte delle informazioni di configurazione necessarie

Dynamic Host Configuration Protocol (DHCP)

- Sono stati creati diversi meccanismi per permettere ad un host di ottenere i diversi parametri
  - uno di questi, noto come Reverse Address Resolution Protocol (RARP), permetteva di ottenere un indirizzo IP da un server
  - ICMP ha due messaggi: "Address Mask Request" e "Router Discovery"
    - usati per ottenere la maschera della rete e l'indirizzo di un router
- Ciascuno di questi meccanismi veniva usato indipendentemente
- DHCP racchiude molti di tali meccanismi e permette ad un host di connettersi ad una rete e ottenere l'indirizzo IP e altre informazioni automaticamente
  - Tale approccio viene di solito chiamato "plug-and-play networking"

Dynamic Host Configuration Protocol (DHCP)

- Quando un host si accende
  - invia in broadcast una richiesta DHCP
  - il server invia una risposta DHCP
    - DHCP usa il termine "offerta" per indicare il messaggio inviato dal server
    - Si dice che il server sta offrendo un indirizzo al client
- E' possibile configurare il server DHCP per fornire due tipi di indirizzi:
  - indirizzi assegnati permanentemente (come nel caso BOOTP)
  - oppure un indirizzo dinamico scelto da un insieme allocato appositamente
- Tipicamente, gli indirizzi permanenti sono assegnati a server, mentre gli indirizzi dinamici sono assegnati ad host generici
  - in ogni caso, gli indirizzi dinamici vengono assegnati solo per un predeterminato periodo di tempo

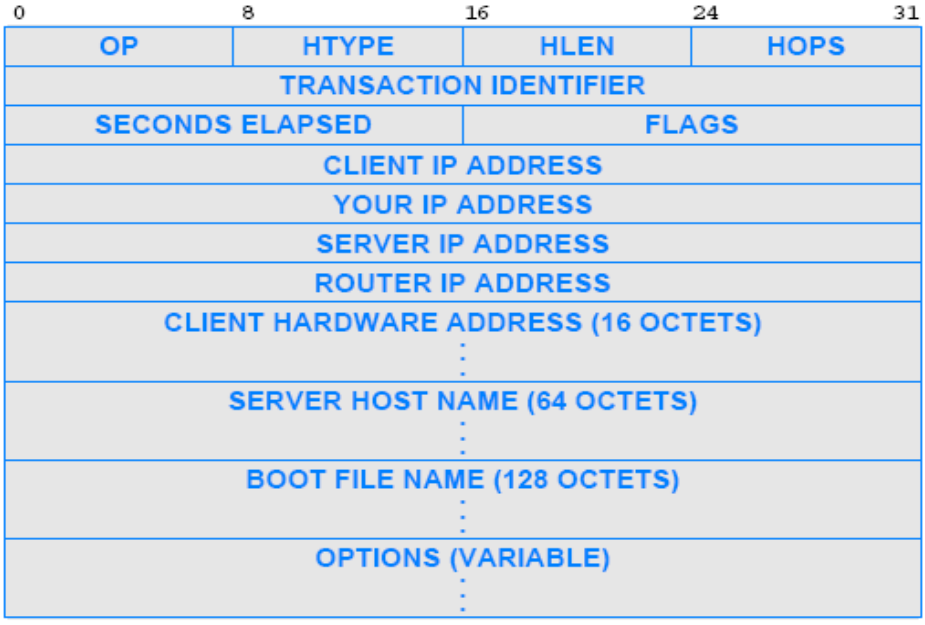
Dynamic Host Configuration Protocol (DHCP)

- DHCP rilascia un indirizzo per un periodo limitato (lease)
  - In questo modo il server DHCP puo' tornare in possesso degli indirizzi
- Quando il periodo di lease scade
  - il server considera l'indirizzo come disponibile per un'eventuale nuova assegnazione
  - un host puo' liberare l'indirizzo o rinegoziare con il server DHCP l'estensione del periodo
- Di solito, il server DHCP approva le richieste di estensione
  - L'host continua a lavorare senza interruzioni
  - In ogni caso, un server DHCP puo' essere configurato per negare l'estensione per ragioni tecniche o amministrative
  - Se il server nega l'estensione, l'host deve smettere di usare l'indirizzo
    - il server DHCP ha il controllo degli indirizzi

Operazioni del Protocollo DHCP e Ottimizzazioni

- Perdite o duplicazioni di pacchetti DHCP
  - DHCP e' stato progettato in modo che la perdita o la duplicazione di pacchetti DHCP non risulti in una configurazione errata
    - Se l'host non riceve risposta, ritrasmette la richiesta
    - Se arriva una risposta duplicata, l'host ignora la replica
- Caching dell'indirizzo del server
  - una volta che l'host ha trovato il server DHCP, lo memorizza per utilizzi futuri
- Limitazione della sincronizzazione delle richieste
  - Il server DCHP usa tecniche per prevenire la ricezione di richieste in contemporanea

DHCP: Formato dei messaggi



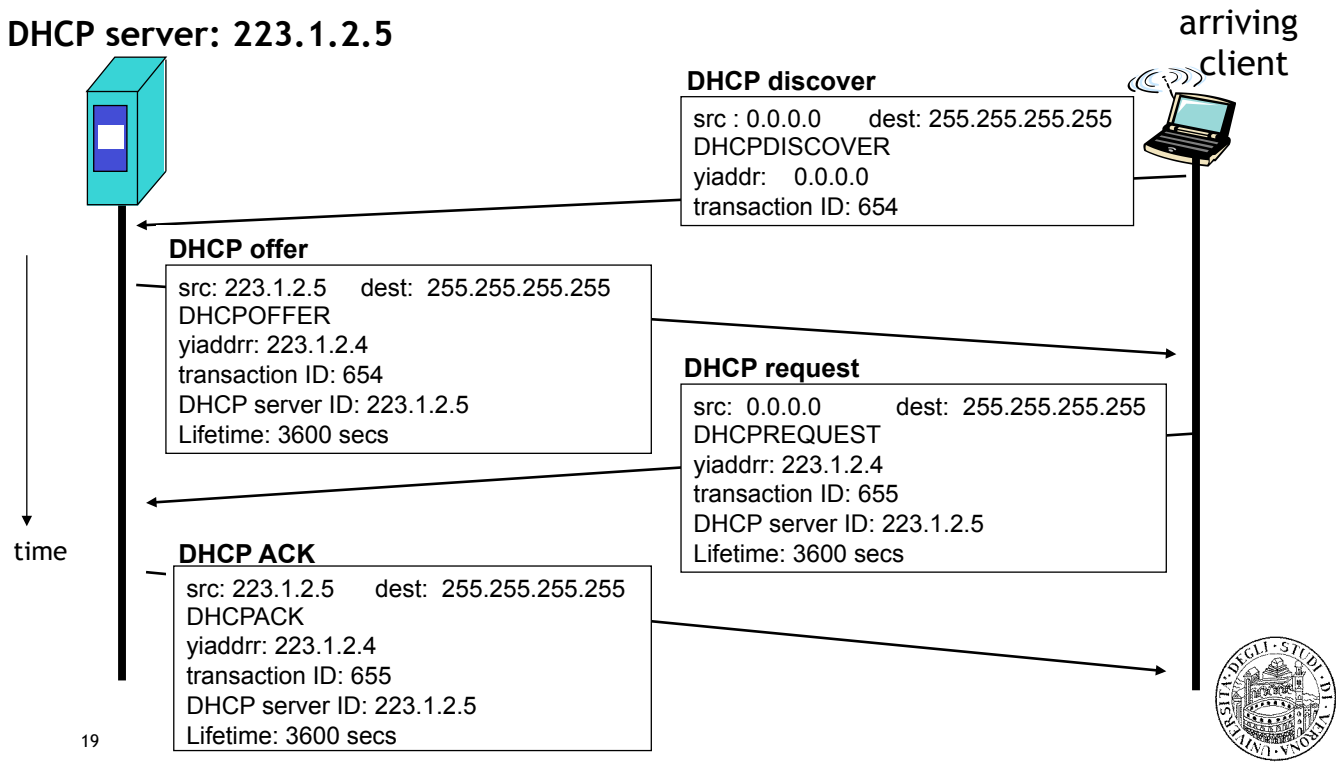
DHCP: Formato dei messaggi

- DHCP adotta un versione leggermente modificata del formato dei messaggi BOOTP
  - OP indica se si tratta di una "Request" o una "Response"
  - i campi HTYPE and HLEN il tipo di hardware della rete e la lunghezza dell' indirizzo hardware
  - FLAGS indica se l' host puo' ricevere messaggi broadcast o risposte dirette
  - HOPS indica a quanti server rigirare la richiesta
  - TRANSACTION IDENTIFIER contiene un valore usato da un host per capire se la risposta di riferisce ad una sua richiesta
  - SECONDS ELAPSED indica quanti secondi sono passati dall' avvio dell' host

DHCP: Formato dei messaggi

- I campi finali sono usati per trasportare nelle risposte informazioni verso la sorgente
  - se un host non conosce il proprio indirizzo IP, il server usa il campo "YOUR IP ADDRESS" per fornire il valore
  - il server usa i campi "SERVER IP ADDRESS" e "SERVER HOST NAME" per fornire all' host informazioni sulla posizione del server
  - il campo "ROUTER IP ADDRESS" contiene l' indirizzo IP del router di default

Scenario client-server DHCP



Riassunto

Cosa accade quando un host…

- … viene acceso
  - ottiene l'indirizzo IP dal server DHCP
    - in alternativa, l'indirizzo puo' essere impostato manualmente
- … vuole inviare un messaggio ad un host sulla stessa rete
  - ottiene l'indirizzo IP dal DNS
  - controlla se l'indirizzo appartiene alla stessa rete
    - ovvero controlla il prefisso (NetID) dell'indirizzo IP
  - invia i dati
- … vuole inviare un messaggio ad un host su una rete diversa dalla propria

  ➔ argomento delle prossime lezioni

Reti di Calcolatori

Algoritmi di routing (I parte)

Università degli studi di Verona
Dipartimento di Informatica

Docente: Damiano Carra

Consegna diretta / indiretta

- Quando un host vuole inviare un messaggio ad un altro host che appartiene alla stessa rete

  ➔ consegna diretta

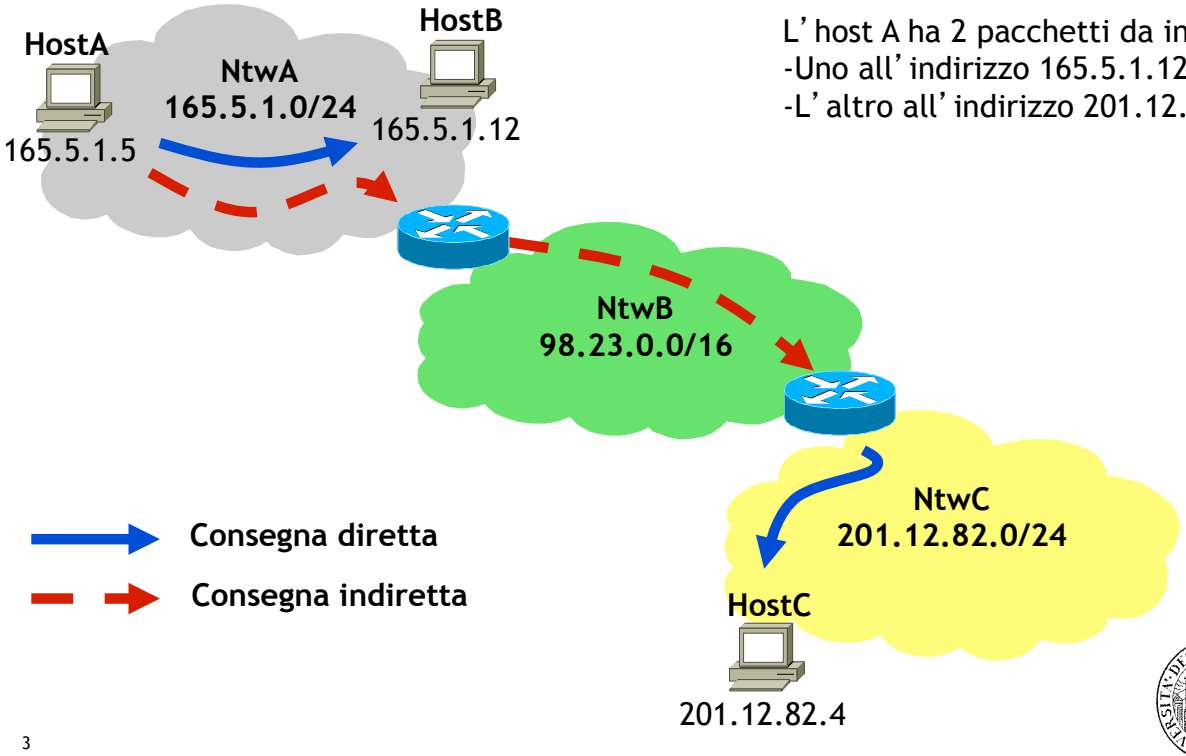
  - L'indirizzo IP appartiene alla stessa rete
  - L'indirizzo fisico viene ottenuto tramite ARP (argomento trattato quando vedremo il livello data link)

- Quando un host vuole inviare un messaggio ad un altro host che appartiene ad un'altra rete

  ➔ consegna indiretta

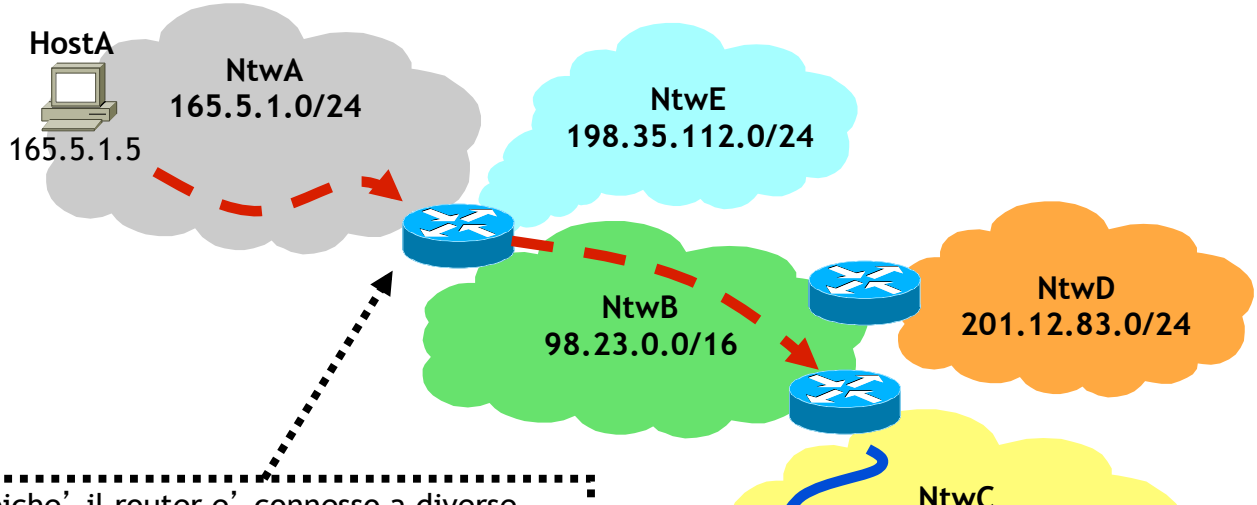
  - Passa il messaggio al router che si fara' carico della consegna
  - I passi intermedi per raggiungere la destinazione vengono fatti grazie agli algoritmi di routing

Consegna diretta / indiretta



L' host A ha 2 pacchetti da inviare:
-Uno all' indirizzo 165.5.1.12
-L' altro all' indirizzo 201.12.82.4

Consegna diretta / indiretta



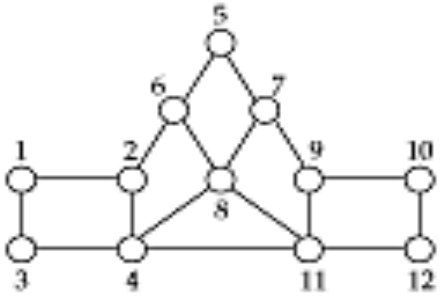
Poiche' il router e' connesso a diverse
- reti
- e router all' interno di ciascuna rete
come puo' il router sapere a chi
consegnare il pacchetto con destinazione
201.12.82.4?

Routing: che cos' e' ?

- Processo di scoperta del cammino da una sorgente ad ogni destinazione nella rete
- Si assuma che un utente si voglia collegare con l' Antartide dal proprio desktop
  - quale cammino dovrebbe prendere?
  - esiste un cammino piu' corto? e piu' veloce?
  - cosa succede se un link sul cammino individuato si guasta?
- Il routing gestisce questi tipi di problemi

Nozioni di base

- Un protocollo di routing gestisce una tabella di routing nei router
  - la tabella indica, per ogni destinazione, qual' e' l' output su cui inviare il pacchetto
- I nodi fanno scelte locali basandosi su topologia globale
  - questo rappresenta il problema principale



ROUTING TABLE AT 1

Destination	Next hop	Destination	Next hop
1	—	7	2
2	2	8	2
3	3	9	2
4	3	10	2
5	2	11	3
6	2	12	3

Problema chiave

- Come effettuare decisioni locali corrette?
  - ciascun router deve conoscere qualcosa sullo stato globale
- Stato globale della rete
  - intrinsecamente grande
  - dinamico
  - informazioni di difficile reperibilita'
- Un protocollo di routing deve saper riassumere le informazioni piu' importanti

Requisiti

- Minimizzare le tabelle di routing
  - per velocizzare il "look up" (data una destinazione, trovare il next hop)
  - per minimizzare i dati da scambiare
- Minimizzare il numero e la frequenza dei messaggi di controllo
- Robustezza: per evitare
  - buchi neri
  - routing loop
  - oscillazioni
- Utilizzo del cammino ottimo

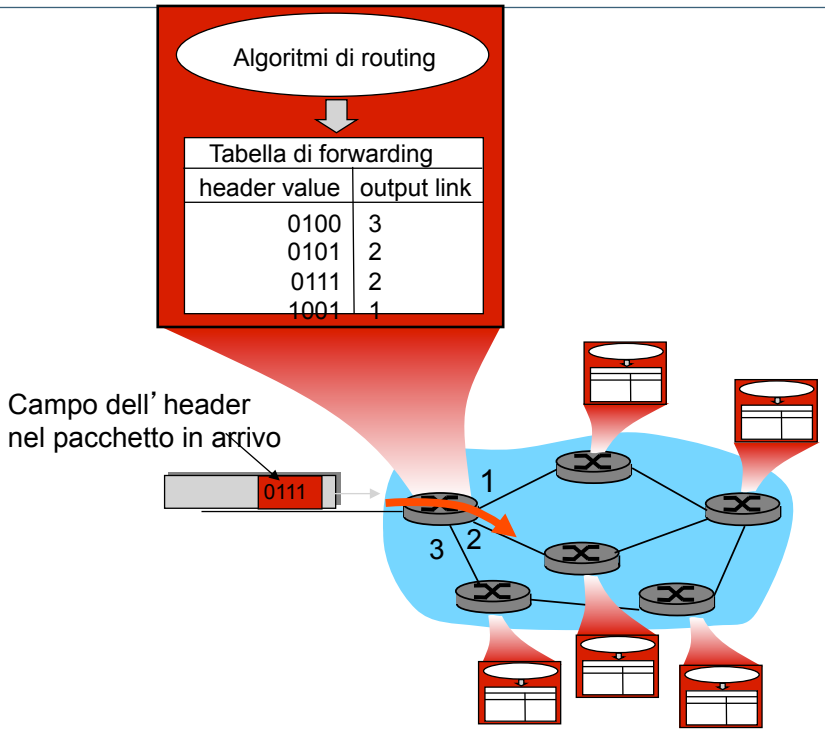
Molti gradi di liberta'

- Routing centralizzato vs distribuito
  - centralizzato e' semplice, ma propenso a guasti e congestioni
- Scambio di informazioni globale vs locale
  - trasmettere informazioni globali e' costoso
- Statico vs dinamico
  - statico puo' andare bene ai bordi della rete, non nel "core"
- Stocastico vs deterministico
  - stocastico favorisce il "load balancing", evita oscillazioni, ma aumenta la probabilita' di pacchetti fuori sequenza
- Cammini singoli vs multipli
  - cammini primari e alternativi
- State-dependent vs state-independent
  - rendere il calcolo dipendente dallo stato della rete (ad es., congestione)

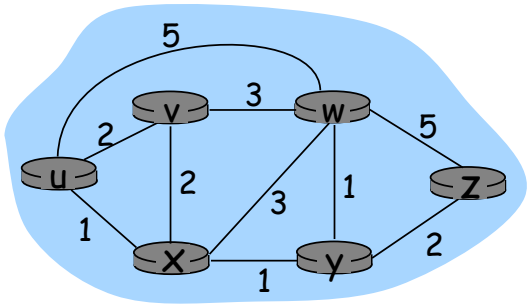
Routing dinamico e Router

- Per assicurare che tutti i router mantengano le informazioni su come raggiungere ogni possibile destinazione
  - ciascun router utilizza un protocollo di propagazione dei cammini
    - per scambiare le informazioni con altri router
  - quando viene a sapere di cambiamenti nei cammini
    - aggiorna la propria tabella di routing
- Poiche' i router scambiano informazioni periodicamente
  - la tabella di routing locale viene aggiornata continuamente

Interazione tra Routing e Forwarding



Astrazione con grafi

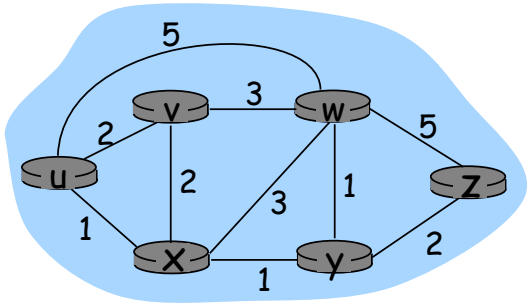


Grafo: G = (N,E)

N = insieme dei router (nodes) = { u, v, w, x, y, z }

E = insieme dei collegamenti (edges) ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Astrazione con grafi: costi



- c(x,x') = costo del collegamento (x,x'

  - ad es., c(w,z) = 5

- il costo puo' essere impostato a 1, o inversamente proporzionale alla banda, o inversamente proporzionale al livello di congestione

Costo del cammino $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Domanda: Qual' e' il cammino a costo minimo tra u e z ?

Algoritmo di routing: algoritmo che trova il cammino a costo minimo

Algoritmi basati su vettori distanza (Distance Vector Algorithms)

Criterio di consistenza

Dato

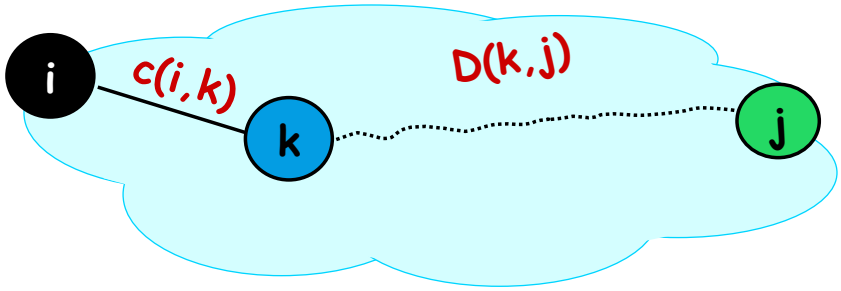
c(i,k) := costo da i a k (collegamento diretto)

D(i,j) := costo del cammino a costo minimo tra i e j

➔ Una porzione di cammino minimo e' anche il cammino minimo tra i nodi che deliminato tale porzione

❑ Quindi, se il cammino minimo dal nodo i al nodo j, con distanza D(i,j), passa attraverso il nodo k, con costo del collegamento pari a c(i,k), allora:

D(i,j) = c(i,k) + D(k,j)



Algoritmi Distance Vector (DV)

❑ Inizializzazione delle distanze:

- D(i,i) = 0 ;
- D(i,k) = c(i,k) se k e' un vicino diretto
- D(i,j) = INFINITO per tutti gli altri nodi

❑ L'insieme dei valori D(i,*) e' il vettore delle distanze del nodo i

❑ L'algoritmo mantiene anche il valore del next-hop (tabella di forwarding) per ogni destinazione j, inizializzato con:

- next-hop(i) = i;
- next-hop(k) = k se k e' un vicino diretto
- next-hop(j) = UNKNOWN altrimenti

Algoritmi Distance Vector (DV)

- Ad ogni iterazione, ciascun nodo invia il proprio vettore delle distanze D(i,*) ai vicini diretti
  - e riceve i vettori delle distanze dai propri vicini diretti
- Per ogni vicino diretto k, se c(i,k) + D(k,j) < D(i,j), allora:
  - D(i,j) = c(i,k) + D(k,j)
  - next-hop(j) = k

Riassunto

Idea di base:

- Periodicamente ogni nodo invia ai propri vicini il vettore delle distanze (distance vector, DV)

Asincrono

- Quando un nodo x riceve il DV da un vicino, aggiorna il priprio DV usando l'equazione di Bellman-Ford:

  $D(x,y) \leftarrow \min_v\{c(x,v) + D(v,y)\}$ per ciacun nodo $y \in N$

- Sotto poche, naturali ipotesi, la stima di D(x,y) converge al valore minimo reale

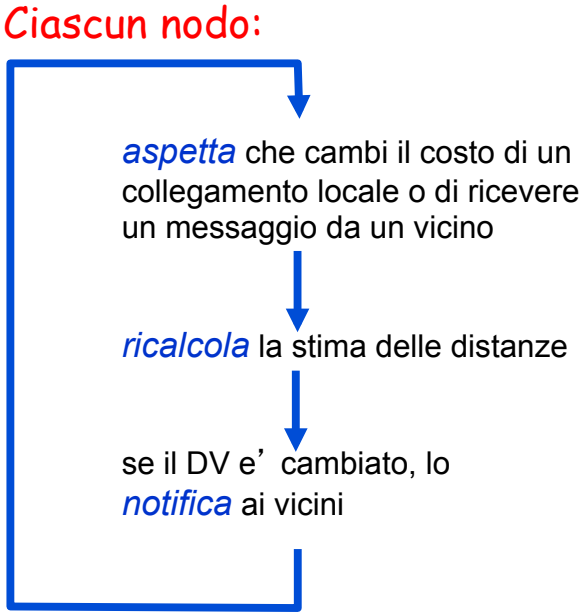
Riassunto

- Iterativo, asincrono:

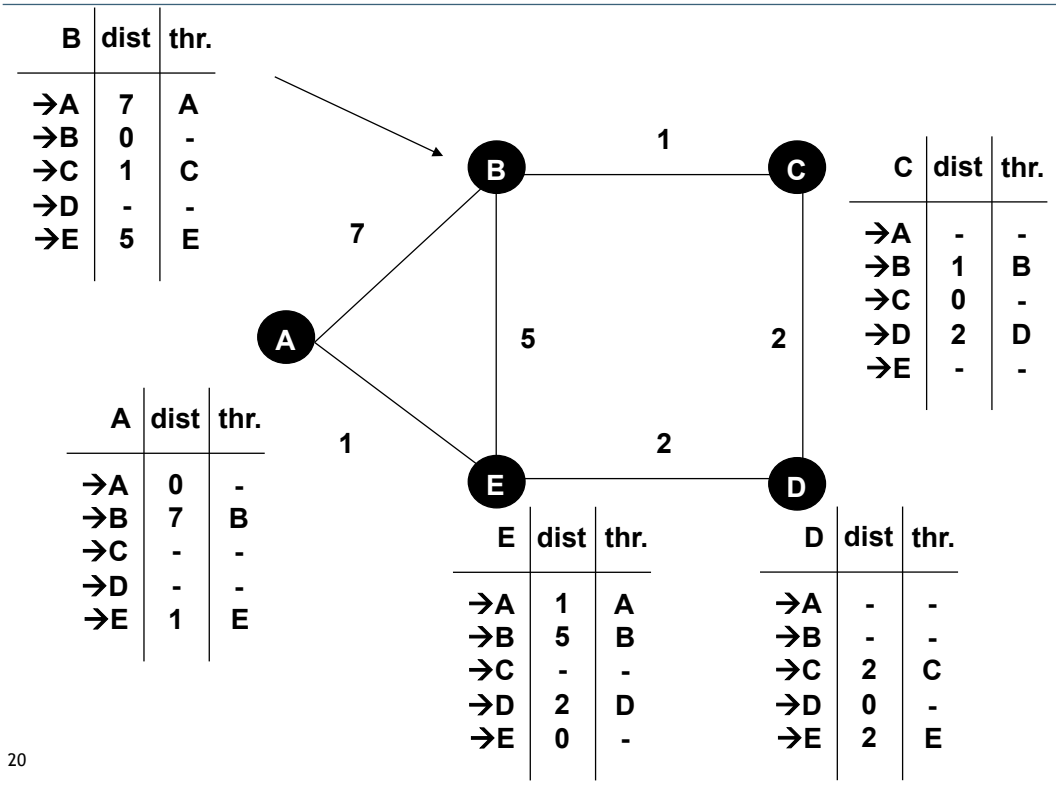
  ciascuna iterazione locale causata da:
  - cambio del costo del collegamento locale
  - ricezione del DV da un vicino

- Distribuito:

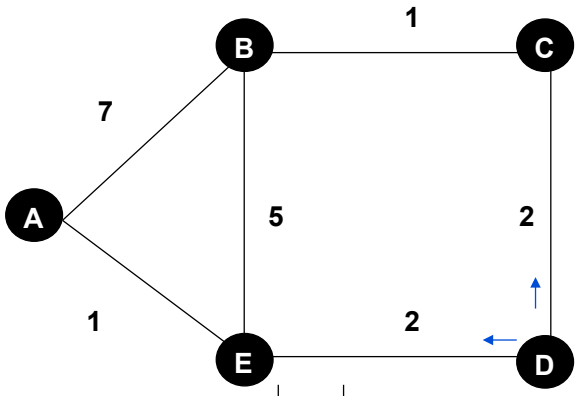
  ciascun nodo manda il proprio DV solo se cambia
  - quindi solo se necessario



Distance Vector: esempio (inizializzazione)



Distance Vector: esempio (scambio di messaggi)

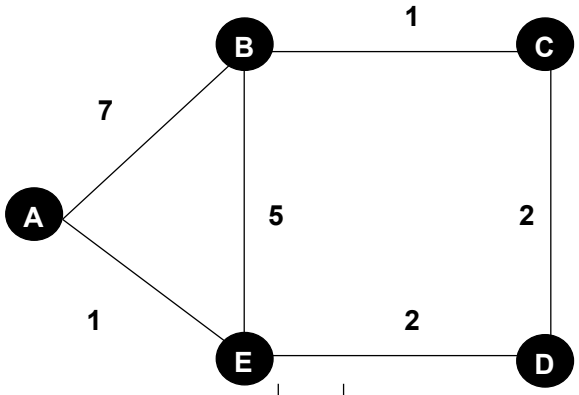


C	dist	thr.
→A	-	-
→B	1	B
→C	0	-
→D	2	D
→E	-	-

E	dist	thr.
→A	1	A
→B	5	B
→C	-	-
→D	2	D
→E	0	-

D	dist	thr.
→A	-	-
→B	-	-
→C	2	C
→D	0	-
→E	2	E

Distance Vector: esempio (scambio di messaggi)

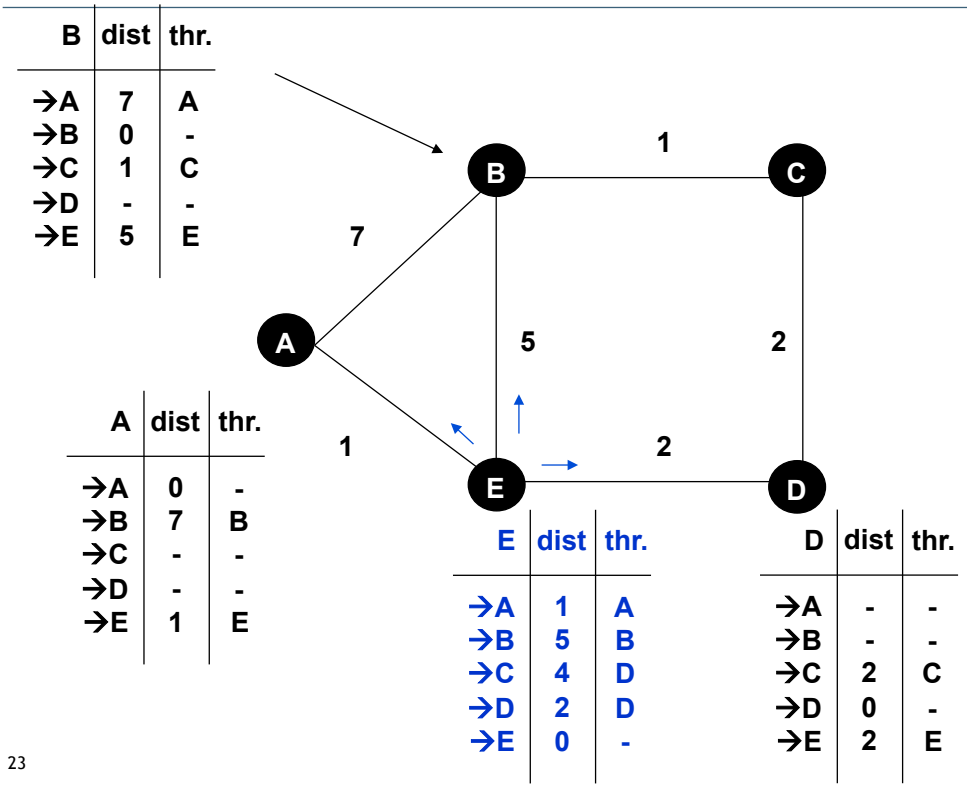


C	dist	thr.
→A	-	-
→B	1	B
→C	0	-
→D	2	D
→E	4	D

E	dist	thr.
→A	1	A
→B	5	B
→C	4	D
→D	2	D
→E	0	-

D	dist	thr.
→A	-	-
→B	-	-
→C	2	C
→D	0	-
→E	2	E

Distance Vector: esempio (scambio di messaggi)



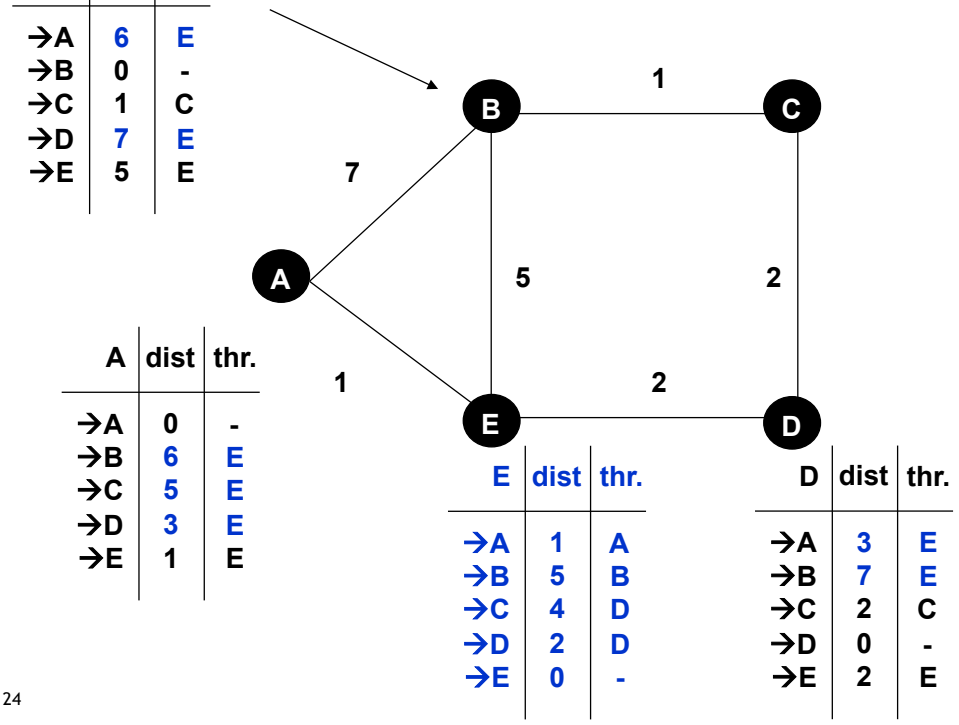
B	dist	thr.
→A	7	A
→B	0	-
→C	1	C
→D	-	-
→E	5	E

A	dist	thr.
→A	0	-
→B	7	B
→C	-	-
→D	-	-
→E	1	E

E	dist	thr.
→A	1	A
→B	5	B
→C	4	D
→D	2	D
→E	0	-

D	dist	thr.
→A	-	-
→B	-	-
→C	2	C
→D	0	-
→E	2	E

Distance Vector: esempio (scambio di messaggi)



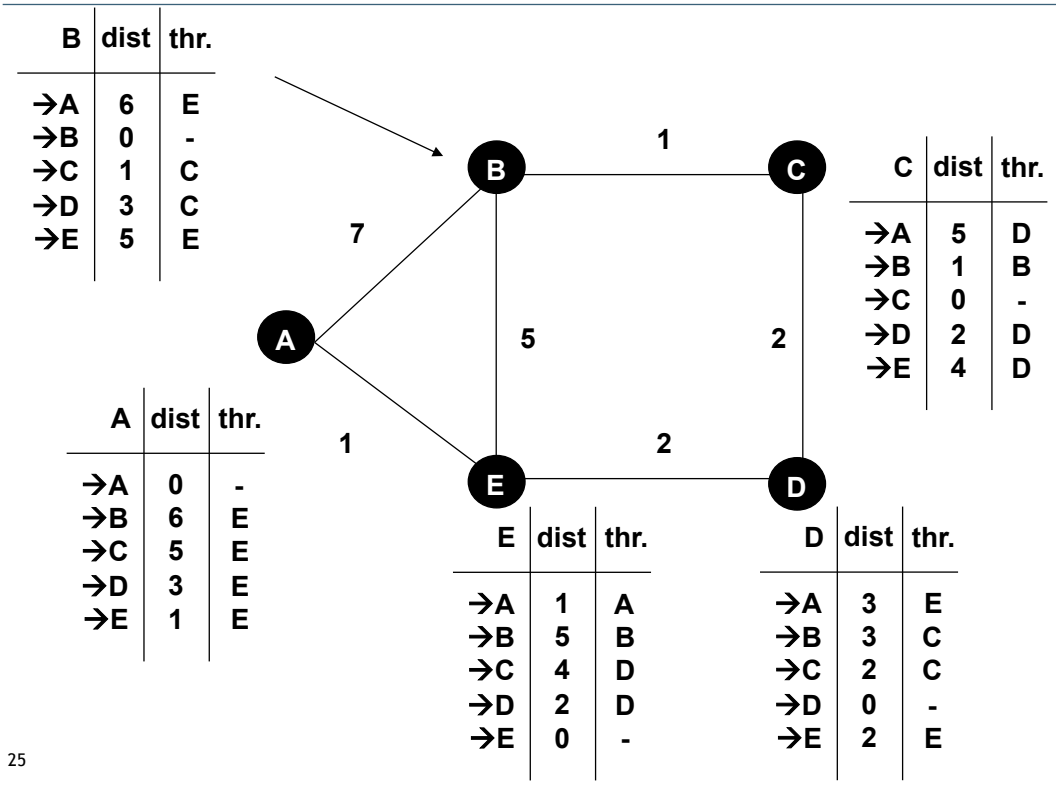
B	dist	thr.
→A	6	E
→B	0	-
→C	1	C
→D	7	E
→E	5	E

A	dist	thr.
→A	0	-
→B	6	E
→C	5	E
→D	3	E
→E	1	E

E	dist	thr.
→A	1	A
→B	5	B
→C	4	D
→D	2	D
→E	0	-

D	dist	thr.
→A	3	E
→B	7	E
→C	2	C
→D	0	-
→E	2	E

Distance Vector: esempio (a regime)



B	dist	thr.
→A	6	E
→B	0	-
→C	1	C
→D	3	C
→E	5	E

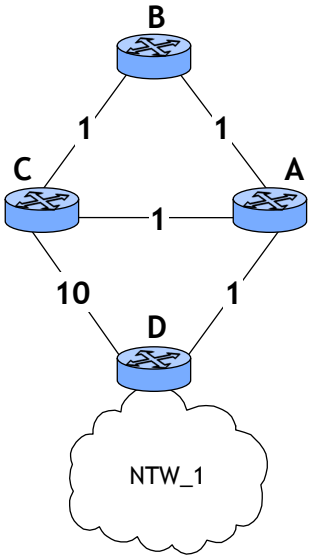
C	dist	thr.
→A	5	D
→B	1	B
→C	0	-
→D	2	D
→E	4	D

A	dist	thr.
→A	0	-
→B	6	E
→C	5	E
→D	3	E
→E	1	E

E	dist	thr.
→A	1	A
→B	5	B
→C	4	D
→D	2	D
→E	0	-

D	dist	thr.
→A	3	E
→B	3	C
→C	2	C
→D	0	-
→E	2	E

Problema: "counting to infinity"



Router A		
Dest	Next	Metric
NTW_1	D	2

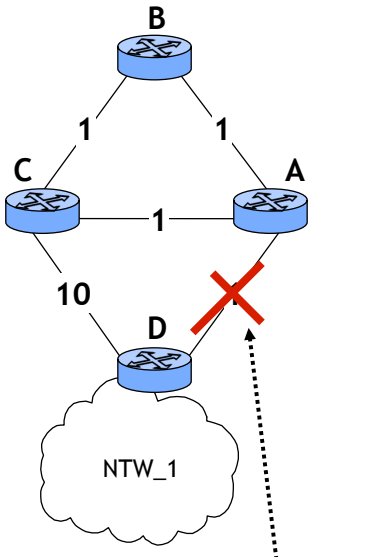
Router B		
Dest	Next	Metric
NTW_1	A	3

Router C		
Dest	Next	Metric
NTW_1	A	3

Router D		
Dest	Next	Metric
NTW_1	dir	1

- Si considerino le informazioni in ciascuna tabella di routing relative alla rete NTW_1
- Il Router D e' connesso direttamente alla rete NTW_1

Problema: “counting to infinity”



time

Router A		
Dest	Next	Metric
NTW_1	Unr.	-

Router B		
Dest	Next	Metric
NTW_1	A	3

Router C		
Dest	Next	Metric
NTW_1	A	3

Router D		
Dest	Next	Metric
NTW_1	dir	1

Router A		
Dest	Next	Metric
NTW_1	C	4

Router B		
Dest	Next	Metric
NTW_1	C	4

Router C		
Dest	Next	Metric
NTW_1	B	4

Router D		
Dest	Next	Metric
NTW_1	dir	1

Router A		
Dest	Next	Metric
NTW_1	C	5

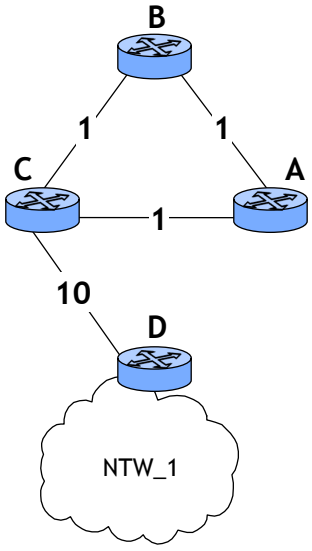
Router B		
Dest	Next	Metric
NTW_1	C	5

Router C		
Dest	Next	Metric
NTW_1	B	5

Router D		
Dest	Next	Metric
NTW_1	dir	1

Il collegamento tra B e D si guasta

Problema: “counting to infinity”



time

…

Router A		
Dest	Next	Metric
NTW_1	C	11

Router B		
Dest	Next	Metric
NTW_1	C	11

Router C		
Dest	Next	Metric
NTW_1	B	11

Router D		
Dest	Next	Metric
NTW_1	dir	1

Router A		
Dest	Next	Metric
NTW_1	C	12

Router B		
Dest	Next	Metric
NTW_1	C	12

Router C		
Dest	Next	Metric
NTW_1	D	11

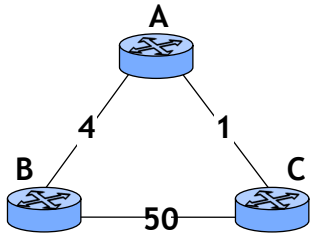
Router D		
Dest	Next	Metric
NTW_1	dir	1

Soluzione al "counting to infinity"

- Il costo massimo di un collegamento e' limitato a 15
  - questo limita automaticamente il tempo di convergenza
- Split Horizon
  - semplice
    - ciascun nodo, quando invia il DV ad un vicino k, omette le destinazioni che hanno k come next hop
  - con "poisoned reverse"
    - ciascun nodo, quando invia il DV ad un vicino k, imposta la distanza a infinito per le destinazioni che hanno k come next hop

Distance Vector: cambio del costo del collegam.

- Se il costo del collegamento cambia:
  - "le buone notizia viaggiano velocemente"
    - buone = il costo diminuisce
  - "le cattive notizia viaggiano lentamente"
    - cattive = il costo aumenta



- Esercizio
  - si provi ad applicare l'algoritmo alla rete mostrata in figura quando
    - il costo del collegamento A → B cambia da 4 a 1
    - il costo del collegamento A → B cambia da 4 a 60

Routing Information Protocol (RIP)

RIP – breve storia

❑fine anni 60:	Protocolli Distance Vector utilizzati per ARPANET
❑meta' anni 70:	protocollo di routing XNS (Xerox Network system), precursore del RIP in IP
	Rilascio di "routed" per BSD Unix
❑1982	RIPv1 (RFC 1058)
❑1988	- routing classful
❑1993	RIPv2 (RFC 1388) - aggiunge le maschere di sottorete a ciascuna entry di routing - permette il routing classless
❑1998	Versione attuale di RIPv2 (RFC 2453)

RIP: Introduzione

- Protocollo intra-dominio semplice
- Implementazione diretta del routing basato su Distance Vector…
  - Versione distribuita delll' algoritmo di Bellman-Ford (DBF)

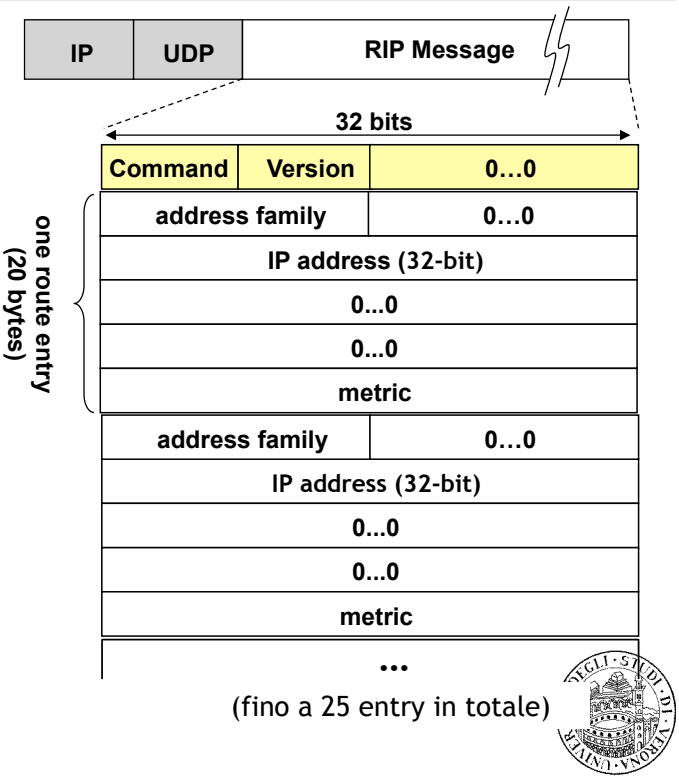
  …con i problemi noti di tali algoritmi
  - convergenza lenta (in caso di guasto)
  - funziona con reti di dimensione limitata
- Punti di forza
  - semplice da implementare
  - semplice da gestire
  - uso diffuso

RIP: Introduzione

- Metrica basata su conteggio degli hop
  - valore massimo e' 15, considerato come "∞"
    - imposto per limitare il tempo di convergenza
  - l' amministratore di rete puo' assegnare valori maggiori di "1" al singolo hop
- Ciascun router invia i vettori delle distanze ogni 30 secondi (o qualora le tabelle di routing cambino per motivi esterni) a tutti i vicini
  - RIP usa UDP, porta 520, per l' invio dei messaggi
- I cambiamenti si propagano sulla rete
- Le entry hanno un timeout di 3 minuti
  - se scade, la distanza viene posta a 15

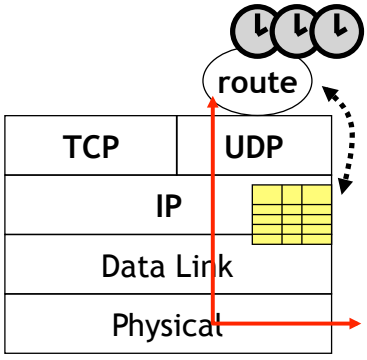
RIP: Formato dei messaggi

- Command: 1=request 2=response
  - Gli aggiornamenti sono considerati "response" sia che ci sia stata una richiesta esplicita che non
  - Un nodo appena connesso invia in broadcast le richieste
  - Alle richieste si risponde immediatamente
- Version: 1
- Address family: 2 per IP
- IP address: la parte di HostID e' sempre posta a zero
- Metric
  - Distanza dal router alla rete specificata nell' indirizzo IP
  - Tipicamente = 1, ovvero la metrica rappresenta il numero di hop



RIP: procedure

- Le tabelle di routing di RIP sono gestite da processi di livello applicativo
  - ad es., *routed* sulle macchine UNIX
- I messaggi vengono inviati su UDP (porta 520)
- RIP mantiene 3 timer per le proprie operazioni
  - Aggiornamento periodico (25-30 sec)
    - usato per inviare i messaggi di aggiornamento
  - Timer di invalidazione (180 sec)
    - Se un' entry non e' stata aggiornata per 180 secondi, essa non viene ritenuta piu' valida
  - Timer per il garbage collection (120 sec)
    - Un' entry non valida viene marcata, ma non rimossa
    - Per 120 sec il router include la destinazione ma con distanza infinita

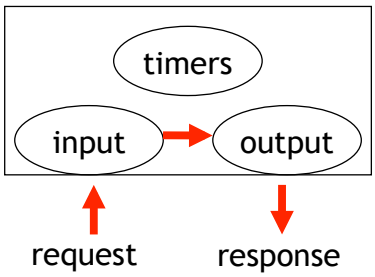


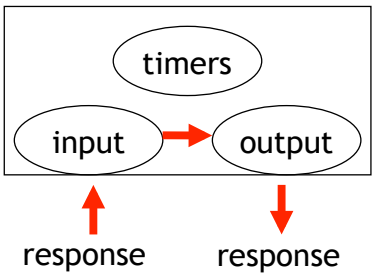
RIP: input processing

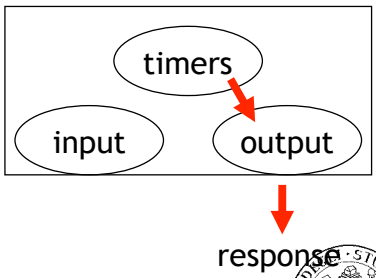
- Messaggi di "Request"
  - generati da router appena avviati
  - azione: il router risponde direttamente a chi ha fatto la richiesta
- Messaggi di "Response"
  - possono arrivare da router che inviano messaggi di aggiornamento, o in risposta ad una query specifica
  - azione: il router aggiorna la sua tabella di routing

RIP: output processing

- Un output viene generato
  - quando un router viene avviato
  - se richiesto dalla procedura di processing degli input
  - dall' aggiornamento regolare
- Azione: il router genera il messaggio a seconda del comando ricevuto

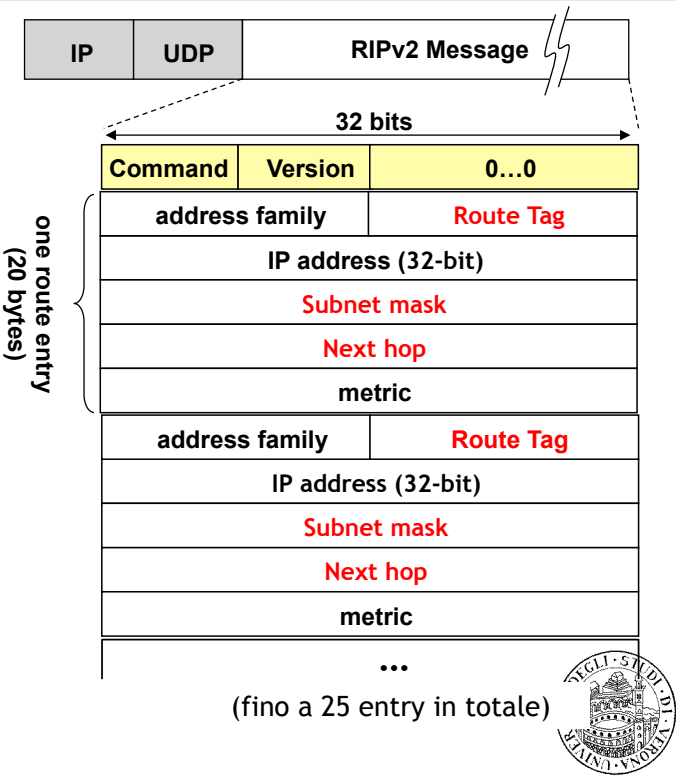






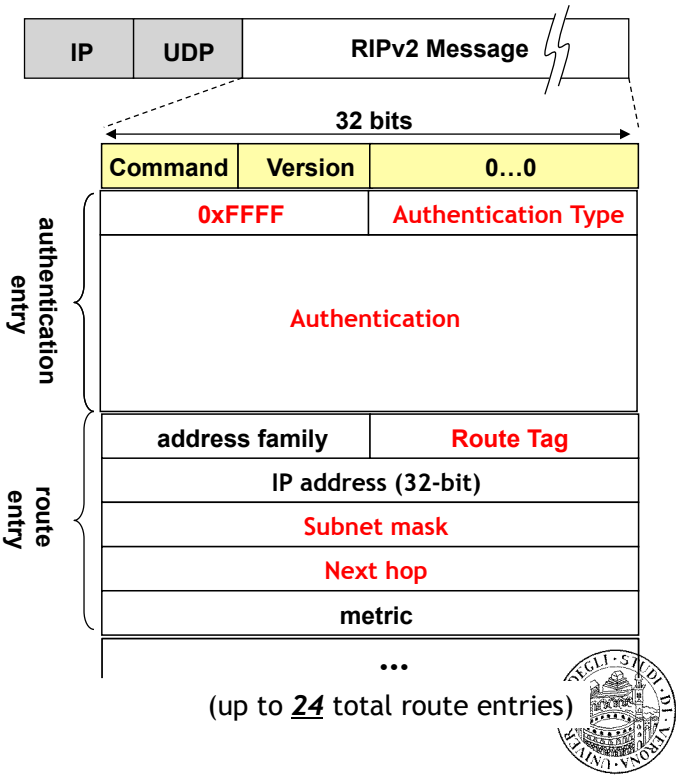
RIPv2: Formato dei messaggi

- Version: 2
- Route Tag: usato per trasportare informazioni di altri protocolli di routing
  - ad es., numero dell'autonomous system
- Maschera di subnet della rete identificata dall'indirizzo IP
- Next hop
  - identifica un indirizzo di next-hop migliore rispetto a quello pubblicizzato dal router (se esiste, altrimenti impostato a zero)



RIPv2: autenticazione

- Qualsiasi host che invia pacchetti UDP sulla porta 520 potrebbe essere considerato un router
  - Possibilita' di iniettare informazioni false
- Con l'autenticazione, solo i router autorizzati possono inviare paccheti RIP
  - Authentication type
    - password
    - MD5
  - Authentication
    - plain text password
    - MD5 hash



RIPv2: altri aspetti

- Uso esplicito delle subnet
- Interoperabilita'
  - RIPv1 e RIPv2 possono essere usati sulla stessa rete perche' RIPv1 ignora i campi sconosciuti
    - RIPv2 risponde alle richieste di RIPv1 con risposte RIPv1
- Multicast
  - invece di inviare i messaggi di RIP in broadcast, RIPv2 usa l'indirizzo di multicast 224.0.0.9

RIP: limitazioni (il costo della semplicita')

- Destinazioni con metriche superiori a 15 non sono raggiungibili
- Una metrica semplice comporta tabelle di routing sub-ottime
- Se non vi e' autenticazione, i router accettano aggiornamenti RIP da chiunque
  - Router mal configurati possono causare danni

Reti di Calcolatori

Algoritmi di routing (II parte)

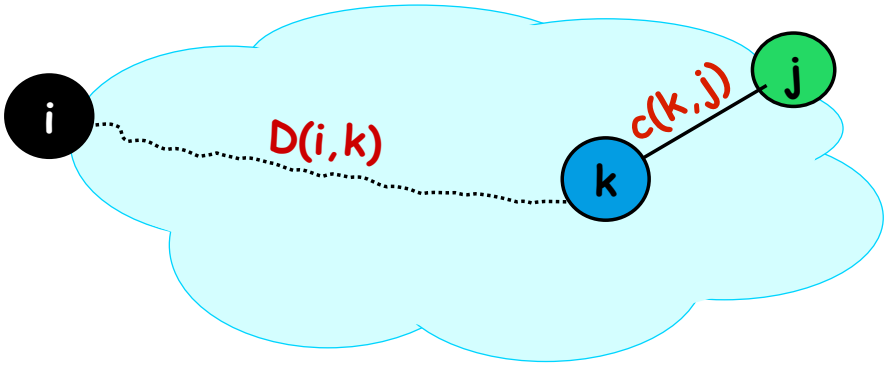
Algoritmi Link State: Introduzione

- Algoritmi Distance Vector
  - Ciascun nodo ha visibilita' locale
    - vicini diretti
  - Le informazioni sul routing sono desunte dalle informazioni ottenute dai vicini
    - ma la struttura (del grafo che rappresenta la rete) non viene specificata

- Gli algoritmi Link State, viceversa, cercano di ottenere una visione globale

Algoritmi Link State: Approccio

- Approccio iterativo, ma prendendo come riferimento la destinazione j e i predecessori di j, ovvero k = p(j)
  - altrimenti noto come algoritmo di Dijkstra
  - si osservi che vale una versione alternativa della condizione di consistenza: D(i,j) = D(i,k) + c(k,j)



- Principale differenza con algoritmi Distance Vector:
  - Ciascun nodo colleziona prima TUTTI i link state c(*,*) e successivamente applica l'algoritmo di Dijkstra al grafo ottenuto

Algoritmi Link State: Approccio

- Dopo ciascuna iterazione, l'algoritmo trova una nuova destinazione *j* e il cammino minimo verso tale destinazione
  - Dopo m iterazioni, l'algoritmo ha esplorato i cammini fino a m hop dal nodo *i*
  - Approccio simile al distance vector
- L'algoritmo di Dijkstra al nodo *i* mantiene due insiemi:
  - l'insieme N dei nodi per cui e' stato trovato il cammino minimo fino a questo momento
  - l'insieme M che contiene tutti gli altri nodi
  - Per tutti i nodi *k*, vengono mantenuti due valori:
    - D(i,k): valore aggiornato della distanza da i a k
    - p(k): il nodo predecessore al nodo k lungo il cammino minimo da i

Dijkstra: inizializzazione

- Inizializzazione:
  - D(i,i) = 0 e p(i) = i;
  - D(i,k) = c(i,k) e p(k) = i se k e' un vicino di i
  - D(i,k) = INFINITO e p(k) = UNKNOWN se k non e' vicino di *i*
  - Insieme N = { i }, e next-hop (i) = i
  - Insieme M = { j | j diverso da i}

- All' inzio l' insieme N ha solo il nodo *i* e l' insieme M ha tutti gli altri nodi
- Al termine dell' esecuzione dell' algoritmo, l' insieme N contiene tutti i nodi, mentre l' insieme M e' vuoto

Dijkstra: iterazione

- In ciascuna iterazione
  - Un nodo j viene spostato dall' insieme M all' insieme N; il nodo j viene scelto secondo il seguente criterio
    - j ha distanza minima da *i* tra tutti i nodi in M, cioe' D(i,j) = min {l ε M} D(i,l)
    - in caso di distanze uguali, la scelta e' casuale tra i nodi a distanza minima
  - Next-hop(j) = il vicino di *i* sul cammino minimo tra i e j, oppure
    - Next-hop(j) = next-hop(p(j)) se p(j) e' diverso da i
    - Next-hop(j) = j se p(j) = i
  - La distanza di tutti i vicini k del nodo j nell' insieme M, se

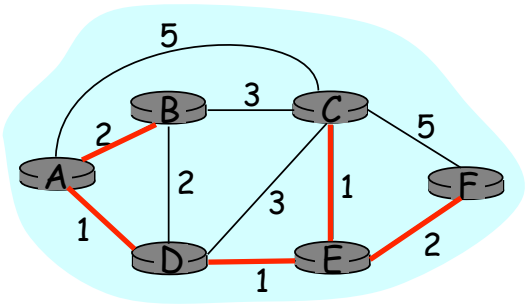
    D(i,k) < D(i,j) + c(j,k)

    viene impostata a:

    D(i,k) = D(i,j) + c(j,k) e p(k) = j.

Algoritmo di Dijkstra: esempio

Step	insieme N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinity	infinity
1	AD	2,A	4,D		2,D	infinity
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



L'albero dei cammini minimi viene chiamato "shortest-paths spanning tree"

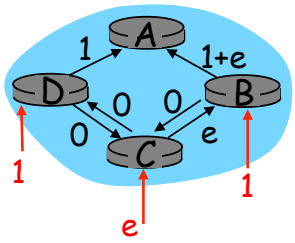
Algoritmo di Dijkstra: discussione

Complessita' dell'algoritmo con n nodi

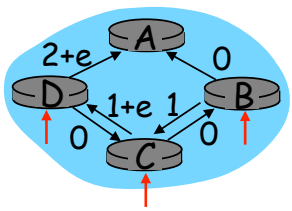
- in ciascuna iterazione serve controllare tutti i nodi nell'insieme M
- n(n+1)/2 confronti: $O(n^2)$
- Esistono implementazioni piu' efficienti: O(nlogn)

Possibili oscillazioni:

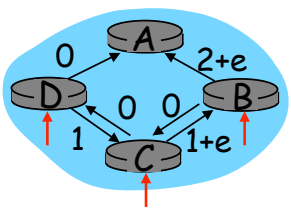
- ad es. se il costo del link = quantita' di traffico



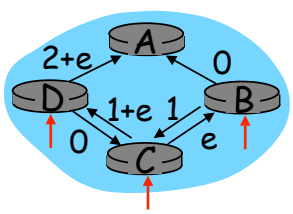
inizialmente



... ricalcolo del routing



... ricalcolo



... ricalcolo

Come assegnare il costo?

- La scelta del costo dei collegamenti ha un impatto sul traffico
  - Costo basso = probabilita' alta di appartenere all'albero dei cammini minimi e quindi di gestire piu' traffico
- Tradeoff: convergenza vs distribuzione del traffico
  - Si dovrebbero evitare oscillazioni…
  - … ottenendo contemporaneamente un'equa utilizzazione della rete
- Metriche statiche (ad es., numero degli hop)
  - non considerano il traffico
- Metriche dinamiche (ad es., costo basato sull'occupazione delle code o sul ritardo)
  - Molte oscillazioni, difficili da gestire
- Metriche quasi-statiche:
  - si ricalcolano periodicamente le metriche statiche valutando il traffico generale (su una finestra ampia di misura)

Riassunto: approcci al routing

Link State

- Le informazioni sulla topologia sono inviate su tutta la rete (flooding)
- Il miglior cammino viene calcolato da ciascun router localmente
- Il miglior cammino determina il next-hop
- Funziona solo se la metrica e' condivisa e uniforme
- Esempio: OSPF

Distance Vector

- Ciascun router ha una visione limitata della topologia della rete
- Data una destinazione e' possibile individuare il miglior next-hop
- Il cammino end-to-end e' il risultato della composizione di tutte le scelte di next-hop
- Non richiede metriche uniformi tra tutti i router
- Esempio: RIP

Confronto tra algoritmi LS e DV

Complessita' relativa ai messaggi

❑LS:

- con n nodi e E link, O(nE) messaggi inviati

❑DV:

- scambio solo tra vicini diretti

Velocita' di convergenza

❑LS:

- O(nlogn)
- potrebbe avere oscillazioni

❑DV:

- variabile
- possibili routing loops
- problema del count-to-infinity

Robustezza: cosa succede in caso di guasti?

❑LS:

- i nodi possono inviare informazioni non corrette sul costo dei link
- ciascun nodo calcola solamente la propria tabella di routing

❑DV:

- i nodi possono inviare informazioni non corrette sul costo dei cammini
- tabelle dei nodi usate da altri nodi
  - gli errori si propagano sulla rete

OSPF
Open Shortest Path First

Open Shortest Path First

- Nel 1988 IETF ha avviato la standardizzazione di un nuovo protocollo di routing
- IETF ha elencato in fase di avvio della standardizzazione un insieme di requisiti che il nuovo protocollo avrebbe dovuto rispettare:
  - soluzione NON proprietaria – aperta
  - parametri di distanza multipli
  - algoritmo dinamico
  - routing basato su *Type of Service*
  - *load balancing*
  - supporto di sistemi gerarchici
  - funzionalità di sicurezza
- Open Shortest Path First (1990, RFC 1247)

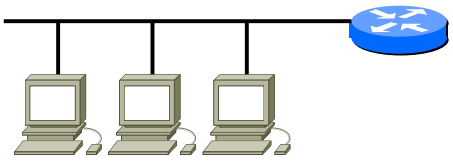
Criteri di progettazione

- I tre principali criteri di progettazione del protocollo OSPF sono:
  - distinzione tra host e router
  - reti broadcast
  - suddivisione delle reti di grandi dimensioni

Distinzione host/router (1)

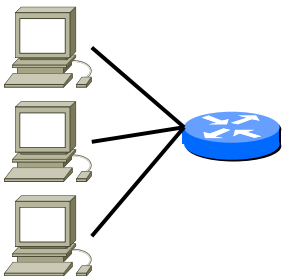
- Nelle reti IP generalmente gli host sono collocati nelle aree periferiche della rete a sottoreti locali connesse alla Big Internet attraverso router
- Il modello link state prevede che il database *link state* includa una entry per ogni link tra host e router
- OSPF introduce il concetto di link ad una *stub network*
  - il link viene identificato dall' indirizzo della sottorete

Distinzione host/router (2)

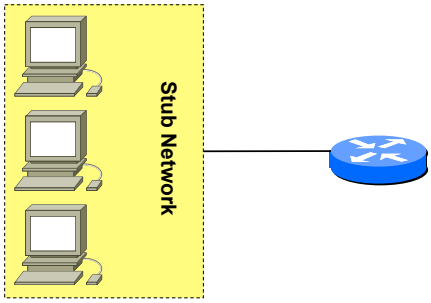


Configurazione fisica

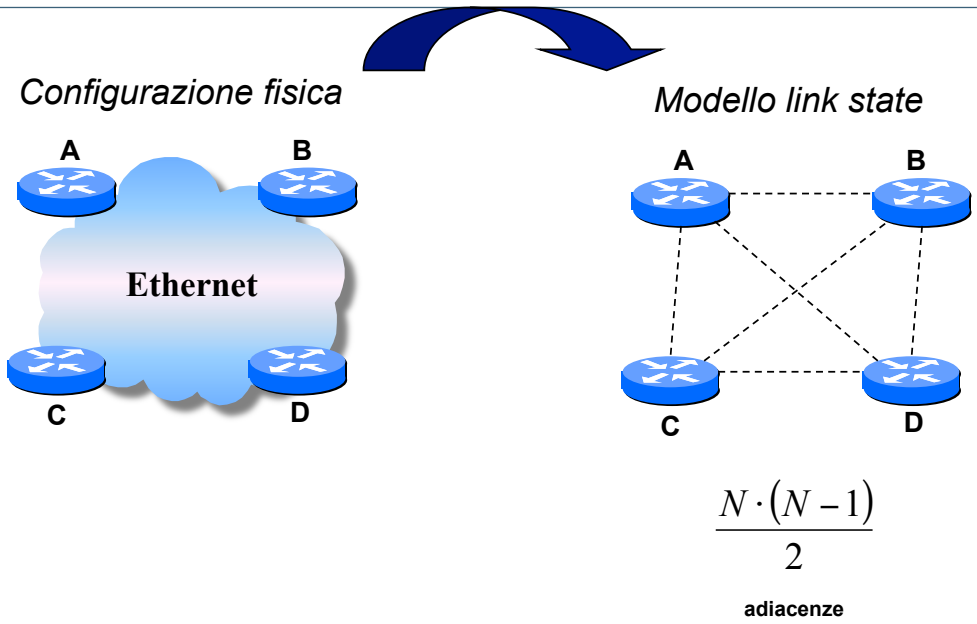
Modello link state classico



Modello OSPF



Reti broadcast (1)

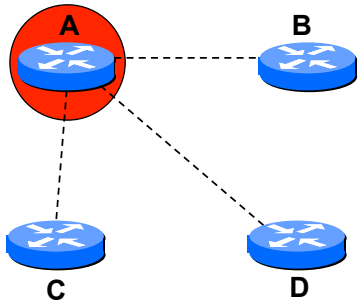


$$\frac{N \cdot (N-1)}{2}$$

adiacenze

Reti broadcast (2)

- Elezione di un nodo della rete broadcast a *designated router*
- L' aggiornamento delle adiacenze viene fatto da tutti gli altri router solamente verso il designated router

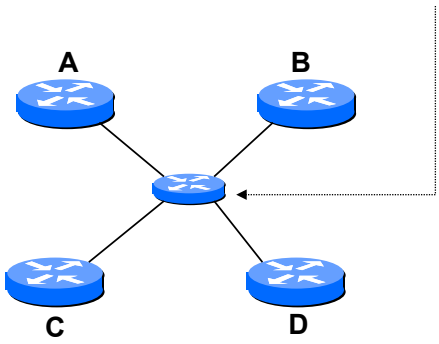


Reti broadcast (3)

- Ogni aggiornamento di link viene notificato solamente al designated router
  - indirizzo multicast “all designated router”: 224.0.0.6
- Se l’aggiornamento modifica il database link state del designated router, quest’ultimo lo propaga in flooding a tutti gli altri nodi
  - indirizzo multcast “all OSPF router”: 224.0.0.5
- Affidabilità ottenuta attraverso un *backup designated router* operante in modo “silenzioso”

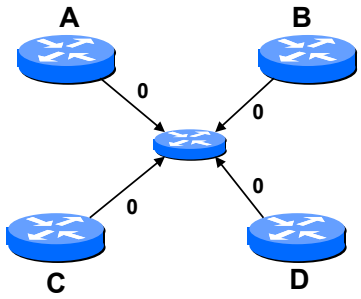
Reti broadcast (4)

- La rete viene rappresentata come un **nodo virtuale**
- Due link per ogni router
  - dal nodo vitruale al router
  - dal router al nodo virtuale (**network link**)



Reti broadcast (5)

- La distanza tra due nodi “raddoppia”
- Soluzione: i network link hanno peso nullo

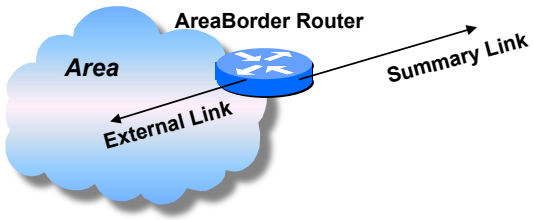


Aree multiple (1)

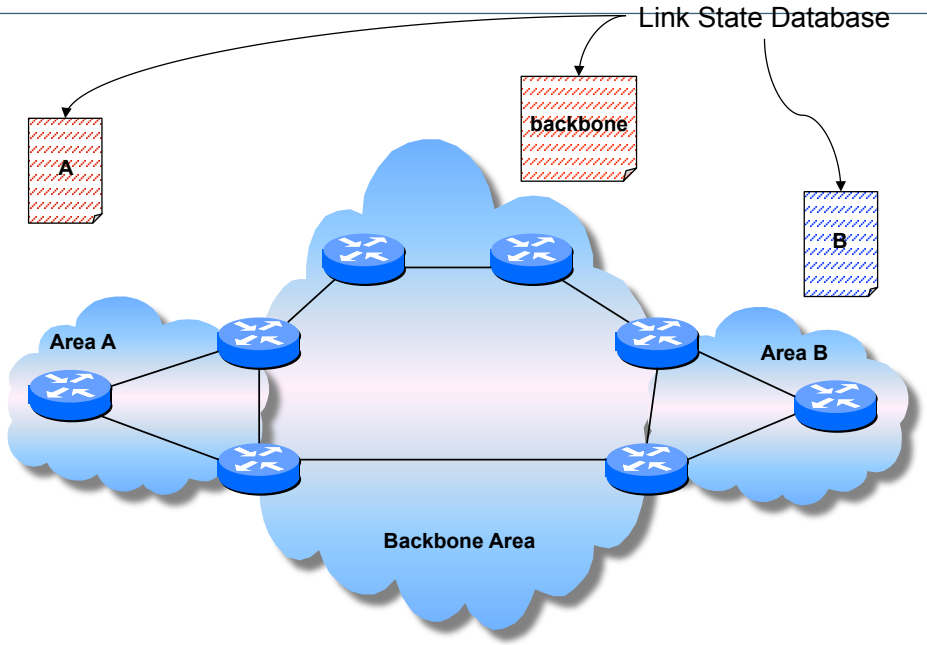
- Il numero di nodi della rete influenza direttamente:
  - dimensione del database di link state di ogni nodo
  - tempo di calcolo dei percorsi ottimi nella rete
  - quantità dei messaggi di routing distribuiti
- OSPF prevede di “spezzare” l’ intera rete in un insieme di sezioni independenti chiamate **aree**
- Sono locali ad ongi area
  - i record del database di link state
  - il flooding dei messaggi di routing
  - il calcolo dei percorsi ottimi di instradamento
- **Backbone Area**: area di livello gerarchico superiore

Aree multiple (2)

- **Area-Border Router**
  - router sono configurati come appartenenti a più aree in modo da garantire l' instradamento inter-area
- Gli Area-Border Router distribuiscono
  - external link state record
    - informano i nodi di un area relativamente ai percorsi uscenti
  - summary link state record
    - informano i nodi della backbone area dei percorsi entranti

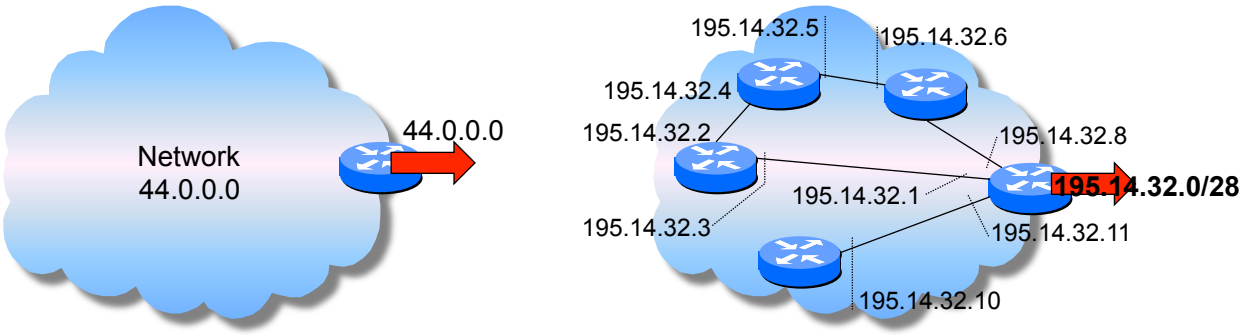


Aree multiple (3)



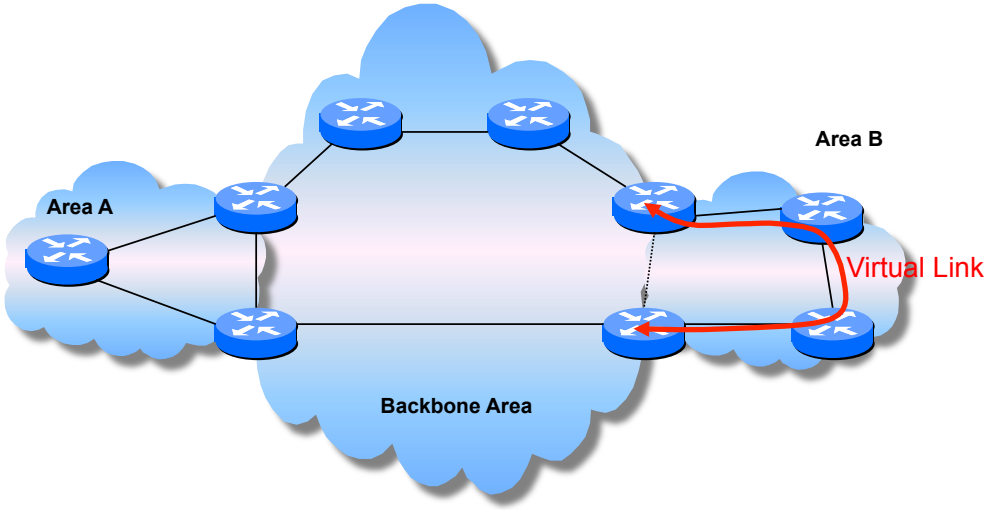
Aree multiple (4)

- Ogni border router "sommarizza" le informazioni di instradamento relative alla propria area



Aree multiple (5)

- Guasti in una Backbone Area possono essere gestiti utilizzando i *virtual link*



Type of Service

- Per ogni link nel database link state possono essere memorizzate più metriche
  - Type of Service Metrics
- Al momento di aggiornamento delle tabelle di routing vengono distribuite tutte le metriche presenti per ogni link
- Il calcolo del percorso ottimo viene fatto
  - sempre per quanto riguarda la metrica di default (ToS 0)
  - opzionalmente per le altre metriche
- I pacchetti IP vengono quindi instradati sulla base del valore contenuto nel campo ToS del loro header

Il protocollo OSPF

- Il protocollo OSPF utilizza a sua volta 3 protocolli per svolgere le proprie funzionalità
  - Hello Protocol
  - Exchange Protocol
  - Flooding Protocol

Messaggi OSPF (1)

- I messaggi OSPF sono trasportati direttamente all' interno dei pacchetti IP
  - non viene utilizzato il livello di trasporto
- Tutti i messaggi OSPF condividono lo stesso header

Version #	*Type*	*Packet length*
Router ID		
Area ID		
Checksum		*Auth Type*
Authentication		
Authentication		

Messaggi OSPF (2)

- Version # = 2
- Type: indica il tipo di messaggio
- Packet Length: numero di byte del messaggio
- Router ID: indirizzo IP del router di riferimento

Version #	*Type*	*Packet length*
Router ID		
Area ID		
Checksum		*Auth Type*
Authentication		
Authentication		

Messaggi OSPF (3)

- Area ID: identificativo dell' area
  - 0 per la Bacvbone area
- Auth Type: tipo di autenticazione
  - 0 no autenticazione, 1 autenticazione con passwd
- Authentication: password

Version #	*Type*	*Packet length*
Router ID		
Area ID		
Checksum		*Auth Type*
Authentication		
Authentication		

Il protocollo Hello

- Funzioni:
  - verificare l' operatività dei link
  - elezione del *designated router* (e relativo elemento di backup)
- Messaggi:
  - Hello

Common header (type = 1, hello)		
Network mask		
Hello interval	*Options*	*Priority*
Dead interval		
Designated router		
Backup Designated router		
Neighbor		

Hello Protocol: formato pacchetto (1)

- Network mask: maschera della sottorete cui appartiene l'interfaccia
- Hello interval: intervallo temporale di separazione tra due messaggi di Hello

Common header (type = 1, hello)		
Network mask		
Hello interval	*Options*	*Priority*
Dead interval		
Designated router		
Backup Designated router		
Neighbor		

Hello Protocol: formato pacchetto (2)

- Designated router: indirizzo IP del designated router
  - 0 se non è stato ancora eletto
- Backup desigated router: indirizzo IP del backup designated router

Common header (type = 1, hello)		
Network mask		
Hello interval	*Options*	*Priority*
Dead interval		
Designated router		
Backup Designated router		
Neighbor		

Hello Protocol: formato pacchetto (3)

- Neighbor: lista di nodi adiacenti da cui ha ricevuto un messaggio di Hello negli ultimi **dead interval** secondi

Common header (type = 1, hello)		
Network mask		
Hello interval	*Options*	*Priority*
Dead interval		
Designated router		
Backup Designated router		
Neighbor		

Hello protocol: procedure

- Regole di elezione del designated router
  - viene utilizzato il campo **priority** del pacchetto Hello
  - ogni router viene configurato staticamente con un valore di priority
    - [0,255]
  - viene selezionato il router con il più alto valore
  - i router con priorità 0 non possono essere eletti

Il protocollo Exchange

- Funzioni:
  - sincronizzazione dei database link state (bring up adjacencies) tra due router che hanno appena verificato l'operatività bidirezionale del link che li connette
  - protocollo client-server
  - messaggi:
    - Database Description Packets
    - Link State Request
    - Link State Update
  - N.B. il messaggio Link State Update viene distribuito secondo le politiche del protocollo di Flooding

Exchange Protocol: messaggi (1)

- Database Description

Common header (type = 2, db description)			
0	*0*	*Options*	*0*
DD sequence number			
Link State Type			
Link State ID			
Advertising router			
Link State Sequence Number			
Link State Checksum		*Link State Age*	

Exchange Protocol: messaggi (2)

- Link State Request

Common header (type = 3, link state request)
Link State Type
Link State ID
Advertising router

- Link state Update

Common header (type = 4, link state update)
Number of link state advertisement
Link state advertisement #1
Link state advertisement #2

Il protocollo di Flooding

- Funzioni:
  - aggiornare il database link state dell' autonomous system a seguito del cambiamento di stato di un link
- Messaggi:
  - Link State Update

Common header (type = 4, link state update)
Number of link state advertisement
Link state advertisement #1
Link state advertisement #2

Routing: considerazioni finali

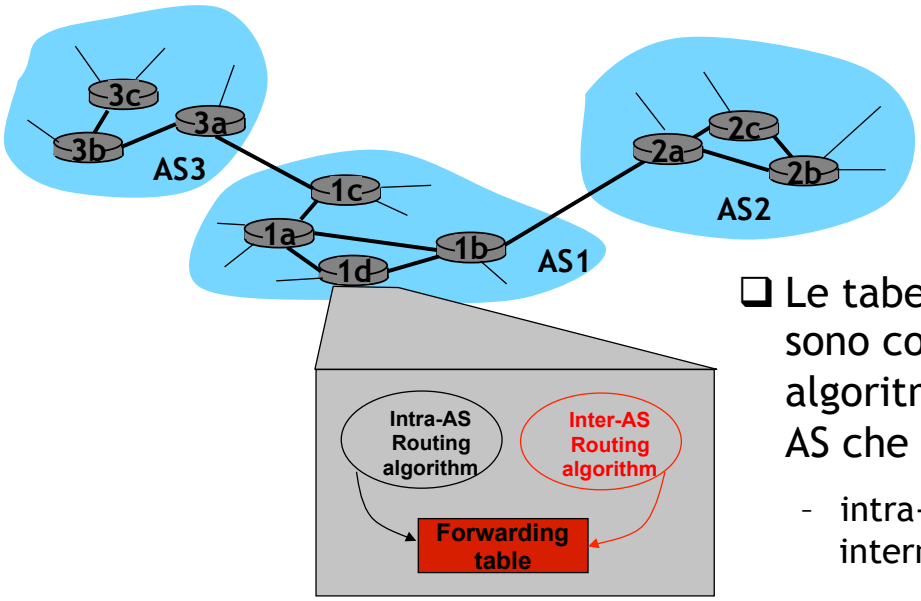
Routing gerarchico

- Il routing che abbiamo visto finora fornisce una visione idealizzata
  - tutti i router sono identici
  - la rete e' "piatta"
- In pratica, la situazione reale e' ben diversa
  - Scalabilita' ➔ con 200 milioni possibili destinazioni:
    - impossibile memorizzare tutte le destinazioni nelle tabelle di routing
    - gli scambi di tabelle cosi' grandi saturerebbero i collegamenti
  - Autonomia amministrativa
    - Internet = rete delle reti
    - ciascun amministratore di rete ha il controllo del routing nella propria rete

Routing gerarchico

- Aggregazione dei router in regioni
  - "autonomous systems" (AS)
- I router nello stesso AS usano lo stesso protocollo di routing
  - protocollo "intra-AS"
  - I router in AS differenti possono usare differenti protocolli intra-AS
- Router di bordo (gateway)
  - collegamento tra un router di un AS con un router di un altro AS

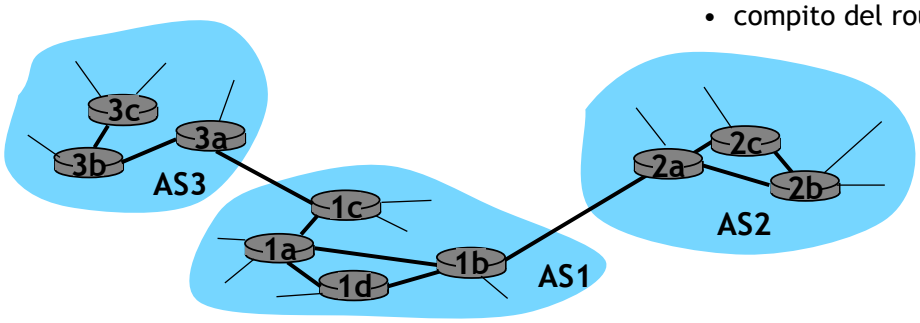
AS interconnessi



- Le tabelle di forwarding sono configurate sia dagli algoritmi di routing intra-AS che da quelli inter-AS
  - intra-AS per le destinazioni interne
  - inter-AS & intra-AS per le destinazioni esterne

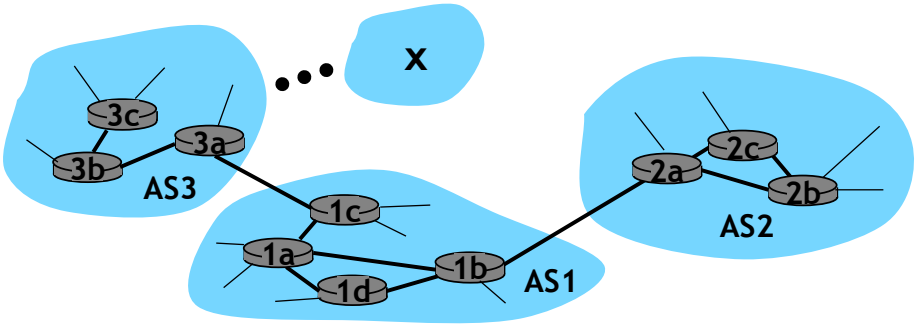
Compiti dei protocolli inter-AS

- Si supponga che un router in AS1 riceva un datagramma con destinazione esterna ad AS1:
  - il router dovrebbe rigirare il pacchetto ad un gateway router, ma quale?
- AS1 deve:
  - conoscere quali destinazioni sono raggiungibili attraverso AS2 e quali attraverso AS3
  - propogare questa informazione di raggiungibilita' a tutti I router in AS1
    - compito del routing inter-AS!



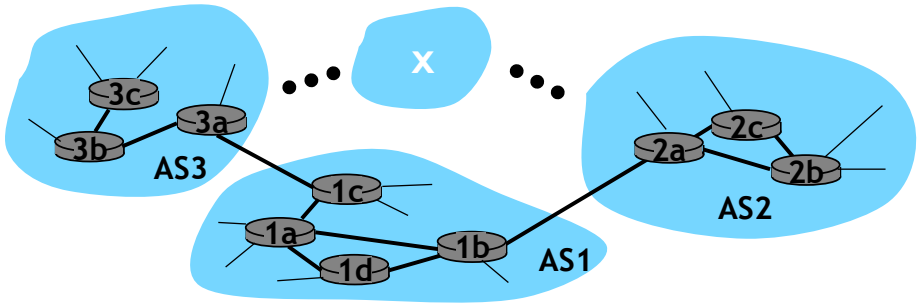
Esempio: impostazione tabelle nel router 1d

- Si supponga che AS1 sappia (attraverso un protocollo inter-AS) che la subnet x e' raggiungibile attraverso AS3 (gateway 1c), ma non attraverso AS2
  - I protocolli inter-AS propagano l'informazione di raggiungibilita' a tutti i router interni
- Il router 1d determina attraverso i protocolli di routing intra-AS che il next hop per raggiungere 1c e' il router 1a
  - verra' impostata una nuova entry con destinazione "x" e next hop "1a"



Esempio: scelta tra piu' AS

- Si supponga che AS1 sappia (attraverso un protocollo inter-AS) che la subnet x e' raggiungibile attraverso AS3 (gateway 1c) e attraverso AS2 (gateway 1b)
- Per configurare la tabella di routing, il router 1d dovrebbe determinare verso quale gateway inoltrare il pacchetto destinato alla rete "x"
- Hot potato routing: inviare il pacchetto al gateway piu' vicino
  - verra' impostata una nuova entry con destinazione "x" e next hop il router verso il gateway piu' vicino



Routing intra-AS

- Noto anche come "Interior Gateway Protocols" (IGP)
- Esempi comuni di protocolli di routing intra-AS:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco, proprietario)

Routing inter-AS: BGP

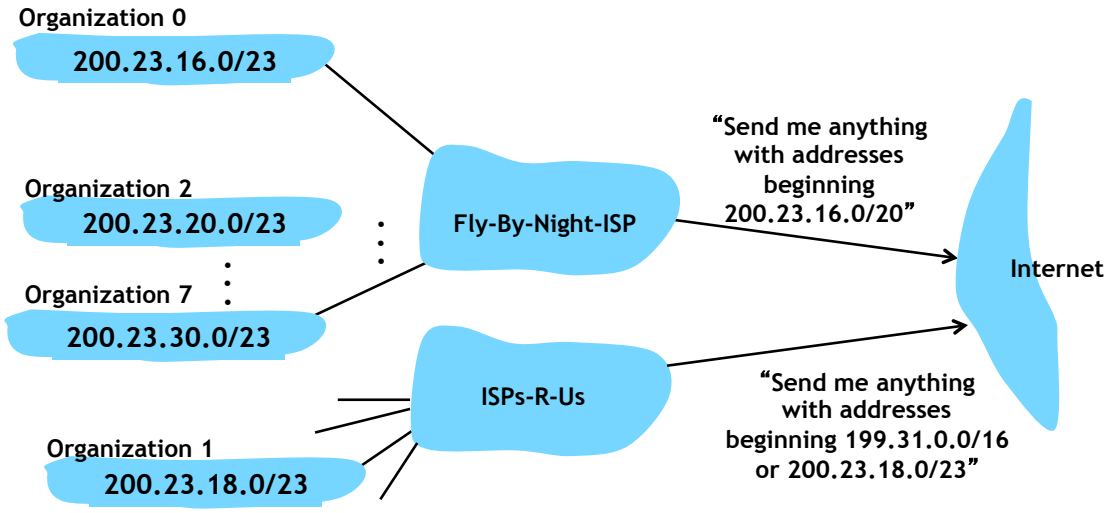
- BGP (Border Gateway Protocol): lo standard de facto
- BGP fornisce agli AS i mezzi per:
  - ottenere le informazioni di raggiungibilita' delle diverse reti dagli AS vicini
  - propagare le informazioni di raggiungibilita' a tutti i router interni degli AS
  - determinare i migliori cammini verso le reti basandosi sulle informazioni di raggiungibilita' e *policy*
- Permette alle reti di far conoscere la propria esistenza al resto di Internet

Perche' differenziare tra routing intra- e inter-AS?

- Scalabilita':
  - routing gerarchico riduce la dimensione delle tabelle e il traffico di update
- Policy:
  - Intra-AS: singolo dominio ammistrativo, policy uniforme
  - Inter-AS: l'amministratore puo' controllare come il traffico viene instradato attraverso la propria rete
- Prestazioni:
  - Intra-AS: puo' essere focalizzato sulle prestazioni (cammini minimi)
  - Inter-AS: le policy possono essere piu' importanti delle prestazioni

Indirizzamento gerarchico: aggregazione

L'indirizzamento gerarchico permette una gestione piu' efficiente degli update



Reti di Calcolatori

Esercizi su routing

Convenzioni utilizzate

- Ogni nodo invia gli update periodicamente ogni T secondi
- Tutti i nodi sono sincronizzati e iniziano a scambiarsi i distance vector (DV) a partire dal tempo t=0;
  - i successivi update vengono inviati dai diversi nodi esattamente nello stesso istante;
- Se il costo di un link cambia (ad es. se un link si guasta), il nodo aspetta il successivo invio degli update
  - non notifica immediatamente il cambiamento ai vicini
  - esempio: se il link si guasta al tempo t = 3T + T/2, l'update viene inviato al tempo t = 4T;
    - semplificazione rispetto al caso generale, dove invece si invia subito un update;
- Se un update ricevuto dai vicini fa cambiare la tabella di routing di un nodo, il nodo aspetta il successivo invio degli update per notificare tale cambiamento
  - non notifica immediatamente il cambiamento ai vicini)
  - semplificazione rispetto al caso generale, dove invece si invia subito un update;

Convenzioni utilizzate

- I nodi utilizzano gli update dei vicini per aggiornare la tabella di routing, e poi scartano l'update ricevuto
  - non tengono memoria del precedente DV ricevuto;
- Se un link si guasta, tutte le destinazioni che hanno come next-hop il nodo coinvolto vengono poste come irraggiungibili
- In definitiva:
  1. ogni nodo invia il proprio DV all'istante T, 2T, 3T, …
  2. ogni nodo riceve il DV dei vicini una frazione di tempo successiva all'instante T, 2T, 3T, …
  3. con i DV ricevuti ogni nodo aggiorna la propria tabella di routing e torna al punto 1;

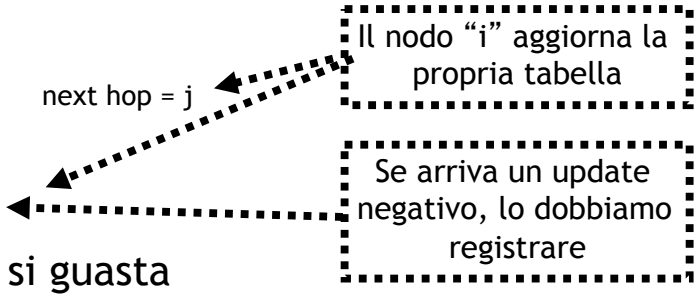
Algoritmo di aggiornamento delle tabelle di routing

- Convenzione
  - c(i,j) e' il costo del link diretto tra il nodo "i" e il suo vicino "j"
  - D(i,k) e' il costo del cammino tra il nodo "i" e il nodo "k"

- Al generico nodo "i"
  - Inizializzazione
    - D(i,i) = 0 e Next-hop(i) = "i";
    - D(i,j) = c(i,j) e Next-hop(i) = "j" se "j" e' un vicino
    - D(i,k) = inf. e Next-hop(i) = - per tutti gli altri

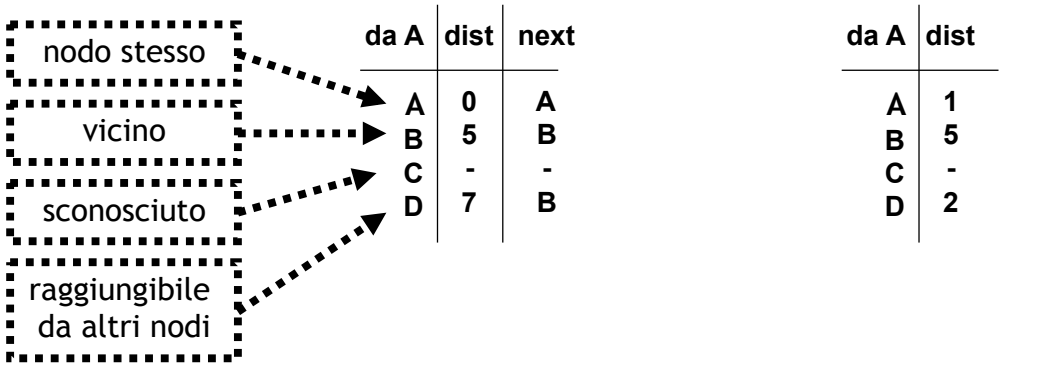
Aggiornamento delle tabelle di routing

- Per ogni distance vector (DV) ricevuto dal nodo "j"
  - per ogni destinazione "k" contenuta nel DV
    - il nodo calcola c(i,j)+D(j,k) e lo confronta con D(i,k) della propria tabella di routing;
    - se c(i,j)+D(j,k) < D(i,k)
      - D(i,k) = c(i,j)+D(j,k) e next hop = j
    - altrimenti, se next hop == j
      - D(i,k) = c(i,j)+D(j,k)



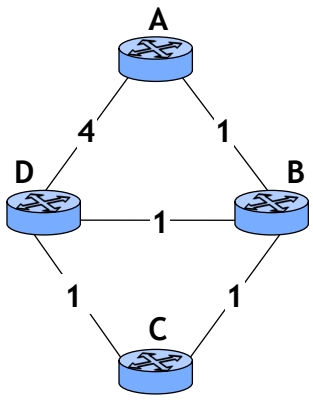
- Se il link verso il nodo "q" si guasta
  - per ogni destinazione "k" contenuta nella tabella di routing
    - altrimenti, se next hop == q
      - D(i,q) = inf.

Notazione utilizzata



Esercizio 1

- Con riferimento alla rete in figura, ove e' utilizzato l'algoritmo Distributed Bellman-Ford (DBF) classico senza alcun meccanismo aggiuntivo
  - Si indichi quale sarà la tabella di routing dei diversi nodi a regime
  - Si mostrino i messaggi scambiati nel caso in cui il link tra A e D si guasti
  - Si mostrino i messaggi scambiati nel caso in cui il link tra A e B si guasti
  - Nel caso in cui l'algoritmo implementi split-horizon con poison-reverse, si mostrino i messaggi scambiati nel caso in cui il link tra A e B si guasti



Esercizio 1 - Soluzione

Tabelle a regime

da A	dist	next
A	0	A
B	1	B
C	2	B
D	2	B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	1	D

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

da D	dist	next
A	2	B
B	1	B
C	1	C
D	0	D

dopo il guasto del link A-D ➔ nessun cambiamento

Tabelle subito dopo il guasto del link A-B

da A	dist	next
A	0	A
B	inf	-
C	inf	-
D	inf	-

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	inf	-
C	inf	-
D	inf	-

B

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

D

da D	dist	next
A	2	B
B	1	B
C	1	C
D	0	D

da D	dist
A	2
B	1
C	1
D	0

da C	dist
A	2
B	1
C	0
D	1

da D	dist
A	2
B	1
C	1
D	0

da B	dist
A	inf
B	0
C	1
D	1

da D	dist
A	2
B	1
C	1
D	0

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	inf
B	0
C	1
D	1

da C	dist
A	2
B	1
C	0
D	1

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	3	C
B	0	B
C	1	C
D	1	D

poteva scegliere anche D

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da D	dist
A	3
B	1
C	1
D	0

B

da B	dist	next
A	3	C
B	0	B
C	1	C
D	1	D

da C	dist
A	3
B	1
C	0
D	1

da D	dist
A	3
B	1
C	1
D	0

C

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

da B	dist
A	3
B	0
C	1
D	1

da D	dist
A	3
B	1
C	1
D	0

D

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	5
C	5
D	4

da B	dist
A	3
B	0
C	1
D	1

da C	dist
A	3
B	1
C	0
D	1

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

← poteva scegliere anche D

da C	dist	next
A	4	D
B	1	B
C	0	C
D	1	D

← poteva scegliere anche B

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

← predilige il colleg. diretto

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da D	dist
A	4
B	1
C	1
D	0

B

da B	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

da C	dist
A	4
B	1
C	0
D	1

da D	dist
A	4
B	1
C	1
D	0

C

da C	dist	next
A	4	D
B	1	B
C	0	C
D	1	D

da B	dist
A	4
B	0
C	1
D	1

da D	dist
A	4
B	1
C	1
D	0

D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	5
C	5
D	4

da B	dist
A	4
B	0
C	1
D	1

da C	dist
A	4
B	1
C	0
D	1

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	5	C
B	0	B
C	1	C
D	1	D

← poteva scegliere anche D

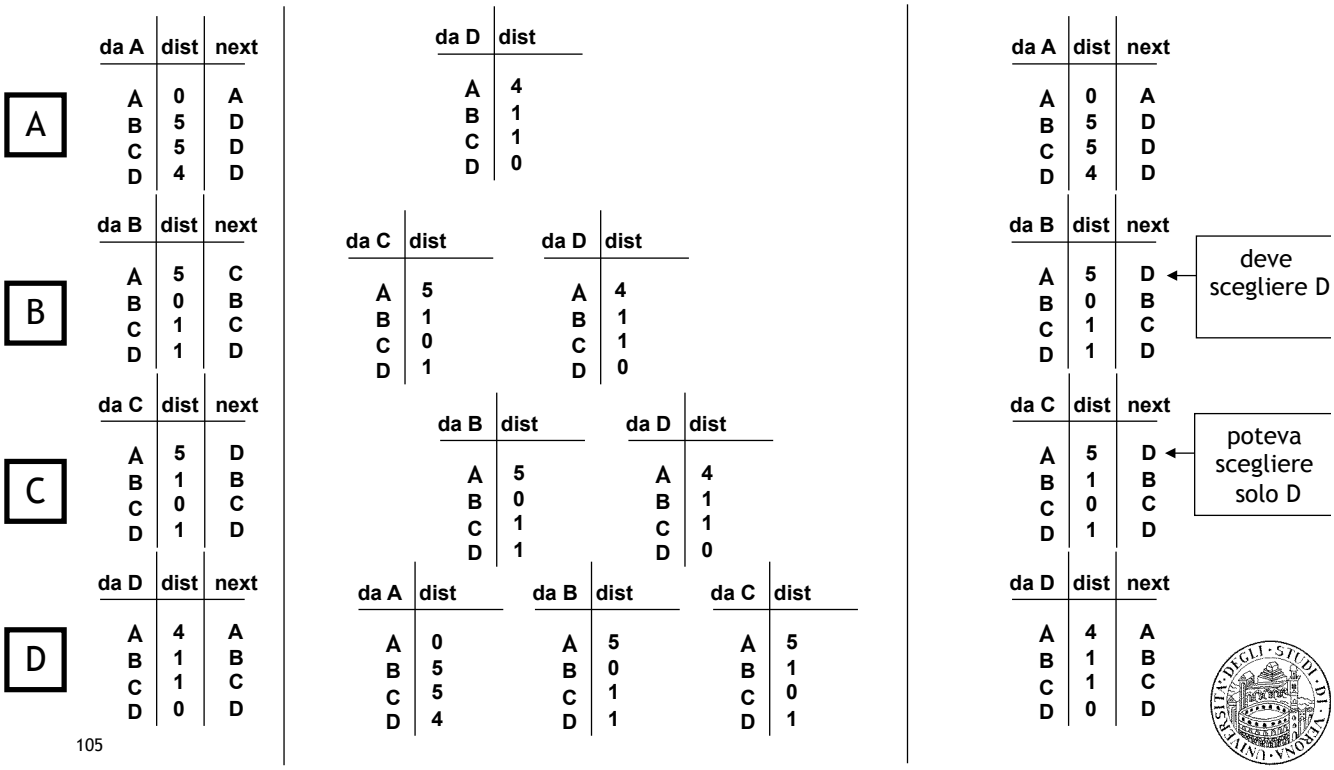
da C	dist	next
A	5	D
B	1	B
C	0	C
D	1	D

← poteva scegliere anche B

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione



Split horizon con poison reverse

- In questo caso, i Distance Vector inviati da "i" a "j" contengono esplicitamente un valore pari ad infinito nelle righe in cui "i" ha come next hop "j"
- Esempio

Tabella del nodo C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

DV inviato da C a B

da C	dist
A	inf
B	inf
C	0
D	1

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	inf	-
C	inf	-
D	inf	-

da D	dist
A	2
B	1
C	1
D	0

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

B

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

da C	dist
A	inf
B	inf
C	0
D	1

da D	dist
A	inf
B	inf
C	1
D	0

da B	dist
A	inf
B	0
C	1
D	1

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

da B	dist
A	inf
B	0
C	inf
D	1

da D	dist
A	2
B	1
C	inf
D	0

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

D

da D	dist	next
A	2	B
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	inf
B	0
C	1
D	inf

da C	dist
A	2
B	1
C	0
D	inf

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	-
C	5	-
D	4	-

da D	dist
A	2
B	1
C	1
D	0

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

B

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

da C	dist
A	3
B	inf
C	0
D	1

da D	dist
A	3
B	inf
C	1
D	0

da B	dist
A	inf
B	0
C	inf
D	1

da C	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

C

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

da B	dist
A	inf
B	0
C	inf
D	1

da D	dist
A	inf
B	1
C	inf
D	0

da C	dist	next
A	inf	-
B	1	B
C	0	C
D	1	D

D

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	inf
B	0
C	1
D	inf

da C	dist
A	inf
B	1
C	0
D	inf

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da D	dist
A	2
B	1
C	1
D	0

B

da B	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

da C	dist
A	inf
B	inf
C	0
D	1

da D	dist
A	4
B	inf
C	1
D	0

C

da C	dist	next
A	inf	-
B	1	B
C	0	C
D	1	D

da B	dist
A	inf
B	0
C	inf
D	1

da D	dist
A	4
B	1
C	inf
D	0

D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	4
B	0
C	1
D	inf

da C	dist
A	inf
B	1
C	0
D	inf

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

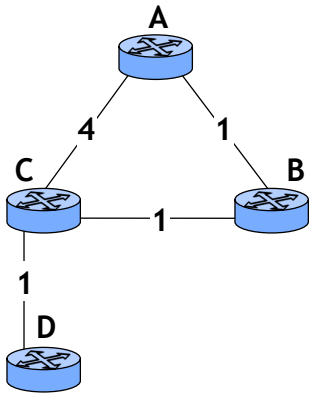
da B	dist	next
A	5	D
B	0	B
C	1	C
D	1	D

da C	dist	next
A	5	D
B	1	B
C	0	C
D	1	D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

Esercizio 2

- Con riferimento alla rete in figura, ove e' utilizzato l'algoritmo Distributed Bellman-Ford (DBF) classico senza alcun meccanismo aggiuntivo
  - Si indichi quale sarà la tabella di routing dei diversi nodi a regime
  - Si mostrino i messaggi scambiati nel caso in cui il link tra A e C cambi costo, da 4 a 1
  - Si mostrino i messaggi scambiati nel caso in cui il link tra C e D si guasti (evento successivo al cambio del costo del link A-C da 4 a 1)
  - Nel caso in cui l'algoritmo implementi split-horizon con poison-reverse, si mostrino i messaggi scambiati nel caso in cui il link tra C e D si guasti (evento successivo al cambio del costo del link A-C da 4 a 1)



Esercizio 2 - Soluzione

Tabelle a regime

da A	dist	next
A	0	A
B	1	B
C	2	B
D	3	B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

da D	dist	next
A	3	C
B	2	C
C	1	C
D	0	D

Tabelle subito dopo il cambio di costo del link A-C

da A	dist	next
A	0	A
B	1	B
C	1	C
D	3	B

da C	dist	next
A	1	A
B	1	B
C	0	C
D	1	D

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l'iterazione

A

da A	dist	next
A	0	A
B	1	B
C	1	B
D	3	B

da B	dist
A	1
B	0
C	1
D	2

da C	dist
A	1
B	1
C	0
D	1

da A	dist	next
A	0	A
B	1	B
C	1	C
D	2	C

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

da A	dist
A	0
B	1
C	1
D	3

da C	dist
A	1
B	1
C	0
D	1

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	1	D

da A	dist
A	0
B	1
C	1
D	3

da B	dist
A	1
B	0
C	1
D	2

da C	dist	next
A	1	A
B	1	B
C	0	C
D	1	D

D

da D	dist	next
A	3	C
B	2	C
C	1	C
D	0	D

da C	dist
A	1
B	1
C	0
D	1

da D	dist	next
A	2	C
B	2	C
C	1	C
D	0	D

Dopo un' iterazione siamo gia' a regime! (le tabelle sono stabili)

Tabelle subito dopo il cambio il guasto del link C-D

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

da D	dist	next
A	inf	-
B	inf	-
C	inf	-
D	0	D

Tabelle dopo il guasto	Distance Vector ricevuti dai vicini	Tabelle dopo l' iterazione	Distance Vector ricevuti dai vicini	Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	2	C

da B	dist
A	1
B	0
C	1
D	2

da C	dist
A	1
B	1
C	0
D	inf

da A	dist	next
A	0	A
B	1	B
C	1	C
D	3	B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	3	A

da C	dist	next
A	1	A
B	1	B
C	0	C
D	3	A

poteva scegliere anche B

da A	dist	next
A	0	A
B	1	B
C	1	C
D	4	B

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

da A	dist
A	0
B	1
C	1
D	2

da C	dist
A	1
B	1
C	0
D	inf

da B	dist	next
A	1	A
B	0	B
C	1	C
D	3	A

da A	dist
A	0
B	1
C	1
D	3

da C	dist
A	1
B	1
C	0
D	3

da B	dist	next
A	1	A
B	0	B
C	1	C
D	4	A

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

da A	dist
A	0
B	1
C	1
D	2

da B	dist
A	1
B	0
C	1
D	2

da C	dist	next
A	1	A
B	1	B
C	0	C
D	3	A

da A	dist
A	0
B	1
C	1
D	3

da B	dist
A	1
B	0
C	1
D	3

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

Distance Vector ricevuti dai vicini	Tabelle dopo l' iterazione	Distance Vector ricevuti dai vicini	Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	4	B

da B	dist
A	1
B	0
C	1
D	4

da C	dist
A	1
B	1
C	0
D	4

da A	dist	next
A	0	A
B	1	B
C	1	C
D	5	B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	5	A

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	6	B

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	4	A

da A	dist
A	0
B	1
C	1
D	4

da C	dist
A	1
B	1
C	0
D	4

da B	dist	next
A	1	A
B	0	B
C	1	C
D	5	A

da A	dist
A	0
B	1
C	1
D	5

da C	dist
A	1
B	1
C	0
D	4

da B	dist	next
A	1	A
B	0	B
C	1	C
D	6	A

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

da A	dist
A	0
B	1
C	1
D	4

da B	dist
A	1
B	0
C	1
D	4

da C	dist	next
A	1	A
B	1	B
C	0	C
D	5	A

da A	dist
A	0
B	1
C	1
D	5

da B	dist
A	1
B	0
C	1
D	5

da C	dist	next
A	1	A
B	1	B
C	0	C
D	6	A

...all' infinito

Soluzione con Split Horizon (+ poison reverse)

Tabelle dopo il guasto

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	2	C

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

Distance Vector ricevuti dai vicini

da B	dist
A	inf
B	0
C	1
D	2

da C	dist
A	inf
B	1
C	0
D	inf

da A	dist
A	0
B	inf
C	1
D	2

da C	dist
A	1
B	inf
C	0
D	inf

da A	dist
A	0
B	1
C	inf
D	inf

da B	dist
A	1
B	0
C	inf
D	inf

Tabelle dopo l' iterazione

da A	dist	next
A	0	A
B	1	B
C	1	B
D	3	C

da B	dist	next
A	1	A
B	0	B
C	1	C
D	3	A

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

Distance Vector ricevuti dai vicini

da B	dist
A	inf
B	0
C	1
D	inf

da C	dist
A	1
B	1
C	0
D	inf

da A	dist
A	0
B	inf
C	1
D	inf

da C	dist
A	1
B	inf
C	0
D	inf

da A	dist
A	0
B	1
C	1
D	3

da B	dist
A	1
B	0
C	1
D	3

Tabelle dopo l' iterazione

da A	dist	next
A	0	A
B	1	B
C	1	C
D	inf	-

da B	dist	next
A	1	A
B	0	B
C	1	C
D	inf	-

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

Soluzione con Split Horizon (+ poison reverse)

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	inf	-

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	inf	-

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

Distance Vector ricevuti dai vicini

da B	dist
A	inf
B	0
C	1
D	inf

da C	dist
A	inf
B	1
C	0
D	4

da A	dist
A	0
B	inf
C	1
D	inf

da C	dist
A	1
B	inf
C	0
D	4

da A	dist
A	0
B	1
C	inf
D	inf

da B	dist
A	1
B	0
C	inf
D	inf

Tabelle dopo l' iterazione

da A	dist	next
A	0	A
B	1	B
C	1	C
D	5	C

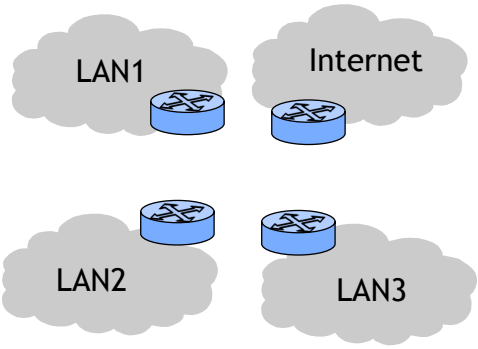
da B	dist	next
A	1	A
B	0	B
C	1	C
D	5	C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

Siamo tornati al punto di partenza, con la distanza verso D uguale a 5 invece che uguale a 2! Anche qui l' iterazione procede all' infinito

Esercizio 3

- Si consideri la rete rappresentata in figura a lato,
  - i quattro router (RA, RB, RC e RD) sono connessi tra loro da canali punto-punto;
  - sulla rete è in funzione un protocollo di routing di tipo Distance Vector che implementa split-horizon con poison-reverse,
  - La metrica utilizzata e' il numero di hop;
  - i distance-vector inviati dai router RA/B/C/D su ciascuna delle loro interfacce sono



	Router A		Router B			Router C		Router D		
	Interf-1	Interf-2	Interf-1	Interf-2	Interf-3	Interf-1	Interf-2	Interf-1	Interf-2	Interf-3
→ LAN 1	inf	1	2	inf	2	4	inf	3	3	inf
→ LAN 2	2	inf	inf	1	1	3	inf	2	2	inf
→ LAN 3	4	inf	3	3	inf	inf	1	2	inf	2
→Internet	3	inf	2	2	inf	2	inf	inf	1	1

Esercizio 3 (cnt'd)

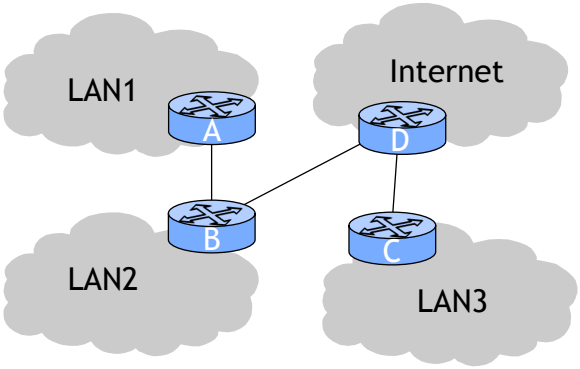
- Domande:
  - Si disegni la topologia del backbone.
  - Si scrivano le tabelle di routing dei router RA/B/C/D.
  - Si dica se in caso di guasti ad uno qualsiasi dei canali punto-punto si possano verificare dei routing loop. Se si, se ne specifichi la natura (permanenti, transitori, ...). Si motivi la risposta.

Esercizio 3: soluzione

- Router A
  - Interf.1: da A a LAN1
  - Interf.2: da A a B
- Router B
  - Interf.1: da B a LAN2
  - Interf.2: da B a A
  - Interf.3: da B a D
- Router C
  - Interf.1: da C a LAN3
  - Interf.2: da C a D
- Router D
  - Interf.1: da D a Internet
  - Interf.2: da D a B
  - Interf.3: da D a C
- In caso di guasti ai link, la rete viene partizionata, per cui ci saranno sicuramente dei routing loop permanenti (vedi esercizio precedente)

da A	dist	next
LAN1	1	dir
LAN2	2	B
LAN3	4	B
Internet	3	B

da D	dist	next
LAN1	3	B
LAN2	2	B
LAN3	2	C
Internet	1	dir



da B	dist	next
LAN1	2	A
LAN2	1	dir
LAN3	3	D
Internet	2	D

da C	dist	next
LAN1	4	D
LAN2	3	D
LAN3	1	dir
Internet	2	D

Reti di Calcolatori

Soluzioni per la carenza di indirizzi IP

Università degli studi di Verona
Dipartimento di Informatica

Docente: Damiano Carra

Acknowledgement

❑ *Credits*

- *Part of the material is based on slides provided by the following authors*
  - *Douglas Comer, "Computer Networks and Internets," 5th edition, Prentice Hall*
  - *Behrouz A. Forouzan, Sophia Chung Fegan, "TCP/IP Protocol Suite," McGraw-Hill*, January 2005

Argmenti trattati

- NAT
- IPv6

Indirizzamento privato: NAT

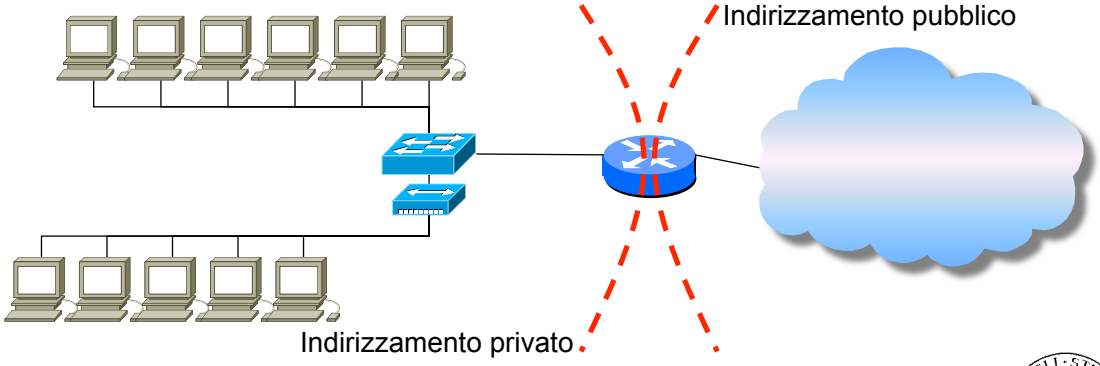
Indirizzi privati

- IETF ha definito alcuni range di indirizzi all' interno dello spazio di indirizzamento IP da utilizzare solamente in ambito privato
  - *private addresses* o *non-routable addresses*
  - ogni volta che un router pubblico riceve un pacchetto destinato ad un indirizzo IP privato, viene segnalato un errore

Prefisso	Indirizzo iniziale	Indirizzo finale
10.0.0.0/8	10.0.0.0	10.255.255.255
172.16.0.0/12	172.16.0.0	172.31.255.255
192.168.0.0/16	192.168.0.0	192.168.255.255
169.254.0.0/16	169.254.0.0	169.254.255.255

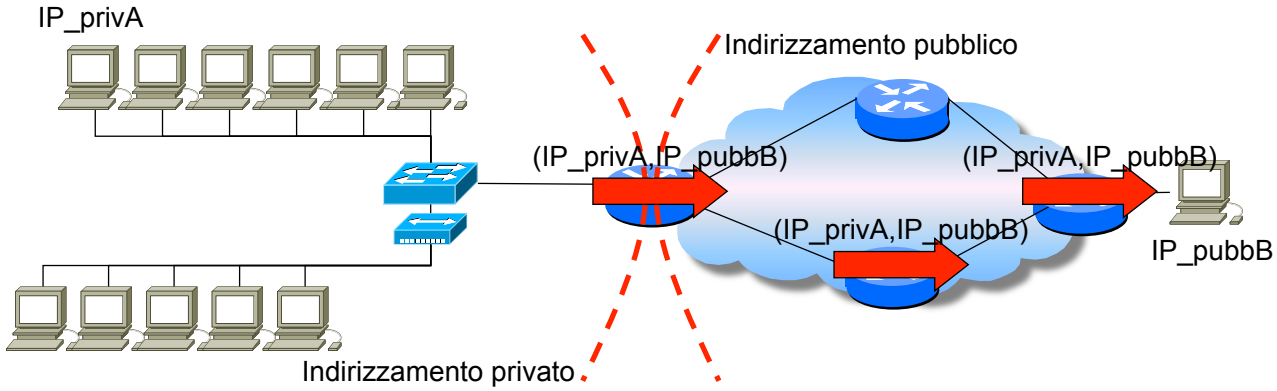
Indirizzi privati: ambito di impiego

- La carenza di indirizzi IP ed il costo degli archi di indirizzamento sono alla base dell' utilizzo degli indirizzi privati
  - le reti con una solo punto di connessione alla Big Internet possono utilizzare l' indirizzamento privato



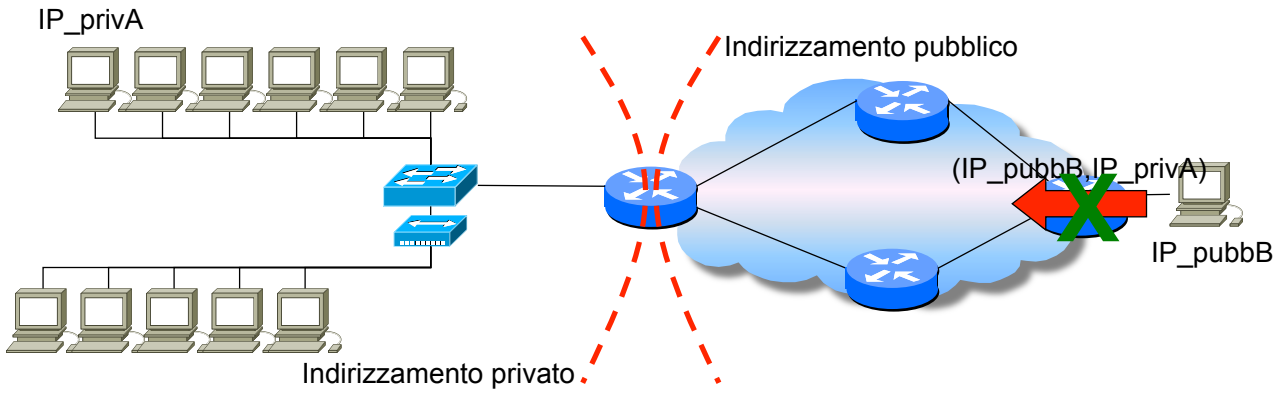
Indirizzi privati: instradamento (1)

- E' necessario introdurre un'ulteriore funzionalità sul bordo tra privato/pubblico per permettere di ricevere i pacchetti all'interno della rete privata



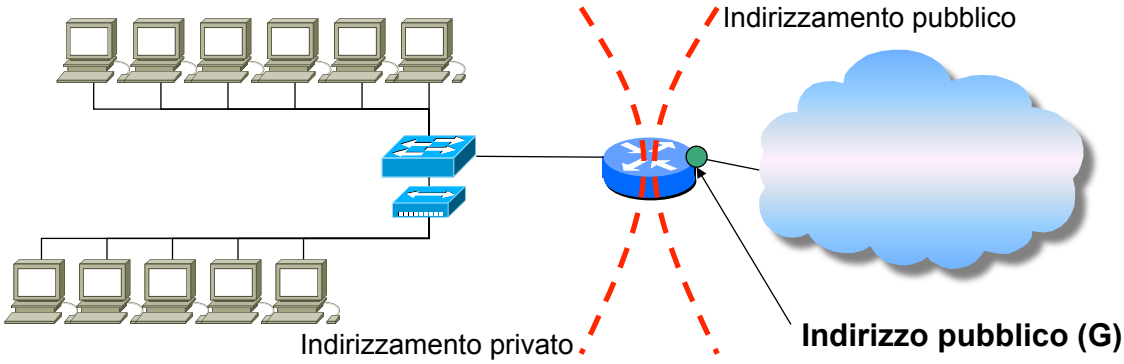
Indirizzi privati: instradamento (2)

- E' necessario introdurre un'ulteriore funzionalità sul bordo tra privato/pubblico per permettere di ricevere i pacchetti all'interno della rete privata



Network Address Translation (1)

- Network Address Translation: funzionalità introdotta per risolvere i problemi di instradamento tra una rete ad indirizzamento privato ed una rete ad indirizzamento pubblico
- Al router di confine tra privato e pubblico viene assegnato un indirizzo pubblico sull' interfaccia verso la rete esterna

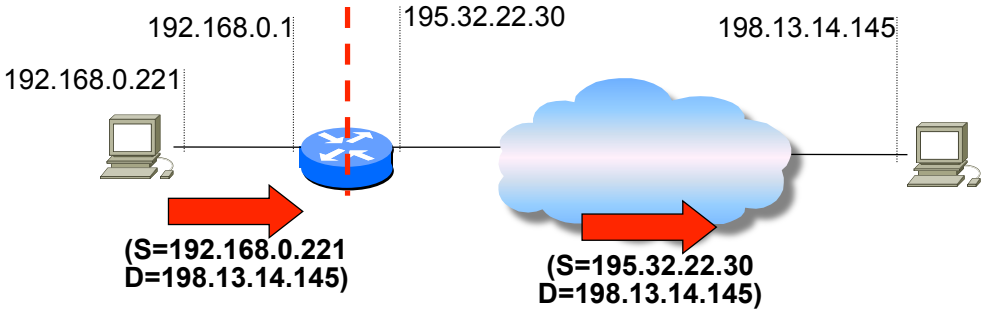


Network Address Translation (2)

- Al router di bordo (privato/pubblico) viene asegnata la funzionalità di **Network Address Translation**
  - NAT traduce l' indirizzo IP dei datagrammi uscenti ed entranti sostituendo
    - l' indirizzo sorgente di ogni pacchetto uscente con il proprio indirizzo pubblico
    - l' indirzzo destinazione di ogni pacchetto entrante con l' indirizzo privato dell' host corretto

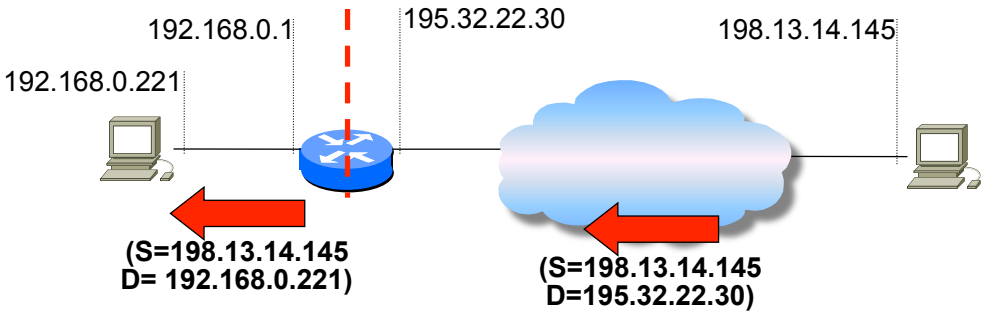
Network Address Translation (4)

- NAT traduce l'indirizzo IP dei datagrammi uscenti ed entranti sostituendo
  - l'indirizzo sorgente di ogni pacchetto uscente con il proprio indirizzo pubblico



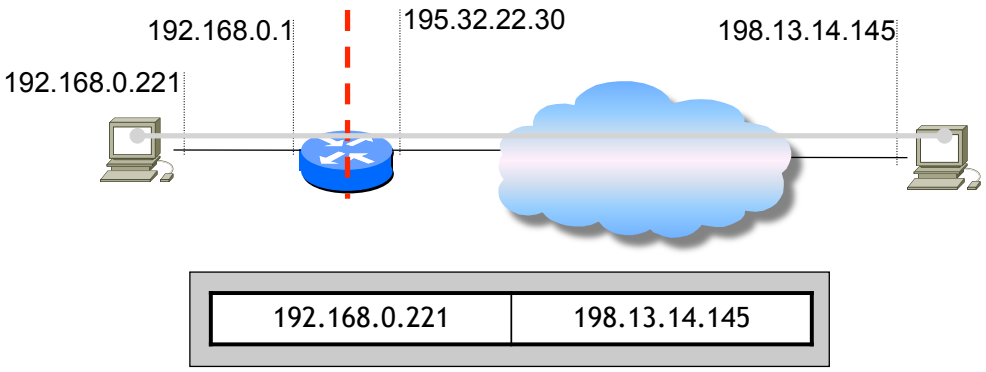
Network Address Translation (5)

- NAT traduce l'indirizzo IP dei datagrammi uscenti ed entranti sostituendo
  - l'indirzzo destinazione di ogni pacchetto entrante con l'indirizzo privato dell'host corretto



Network Address Translation Table (1)

- Il router NAT mantiene al suo interno una tabella di record con il mapping tra indirizzo privato sorgente della comunicazione ed indirizzo pubblico destinazione della comunicazione



Network Address Translation Table (2)

- Metodi di aggiornamento della NAT Table:
  - Configurazione manuale
    - il gestore della rete configura in modo statico i record della NAT Table
  - Datagrammi uscenti
    - i record vengono creati in modo dinamico ogni volta che un pacchetto verso una data destinazione attraversa il NAT
    - cancellati con meccanismo di timeout

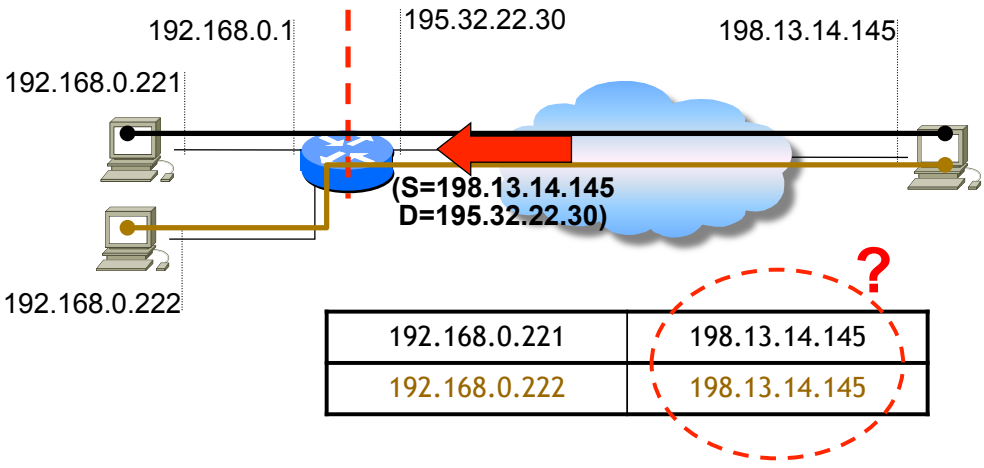
Network Address Translation Table (3)

Configurazione manuale	
Vantaggi	*Svantaggi*
Possibilità permanente di pacchetti in ingresso ed in uscita	Record statici

Datagrammi uscenti	
Vantaggi	*Svantaggi*
Record dinamici	Non permettono l'attivazione di una comunicazione dall'esterno

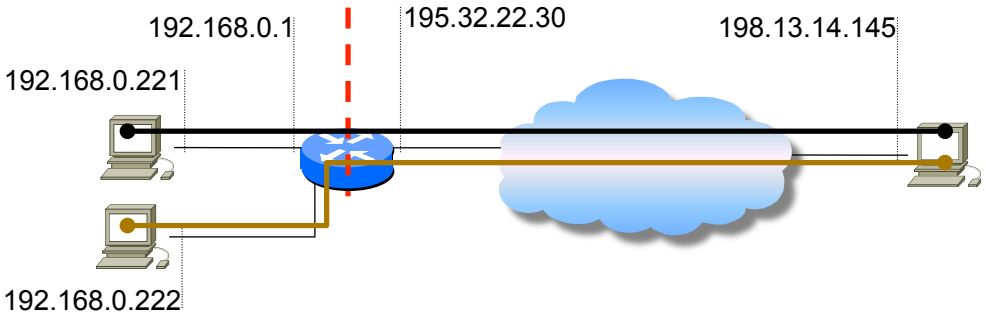
Limitazioni

- Il NAT basato unicamente sull'indirizzo non permette a differenti host privati di connettersi contemporaneamente allo stesso host pubblico



Port mapped NAT (1)

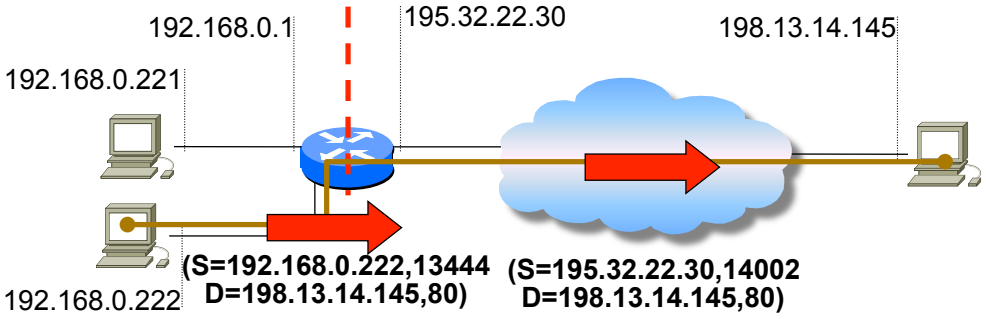
- Il router NAT agisce da gateway di livello 4
  - traduzione sia dell' indirizzo IP che della porta (TCP/UDP)



Priv. Addr	Priv. Port	Ext. Addr	Ext. Port	NATport	Prot. 4
192.168.0.221	21023	198.13.14.145	80	14001	TCP
192.168.0.222	13444	198.13.14.145	80	14002	TCP

Port mapped NAT (3)

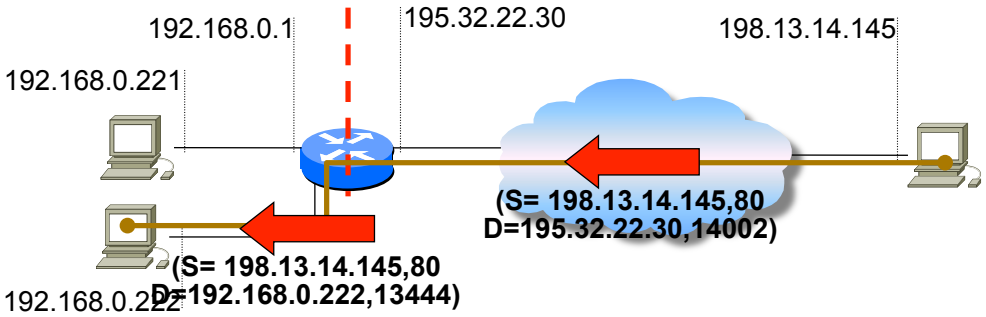
- Datagrammi uscenti



Priv. Addr	Priv. Port	Ext. Addr	Ext. Port	NATport	Prot. 4
192.168.0.222	13444	198.13.14.145	80	14002	TCP

Port mapped NAT (4)

- Datagrammi entranti



Priv. Addr	Priv. Port	Ext. Addr	Ext. Port	NATport	Prot. 4
192.168.0.222	13444	198.13.14.145	80	14002	TCP

IPv6

Le ragioni del cambiamento

- Il protocollo IP usa 32 bit per l'indirizzo
  - quando e' stato definito, lo spazio di indirizzamento sembrava sufficiente
- La crescita di Internet e' stata tuttavia esponenziale
- Se l'attuale tasso di crescita viene mantenuto
  - tutti i possibili prefissi di rete verranno prima o poi assegnati
  - bloccando di fatto un'ulteriore crescita

Le ragioni del cambiamento

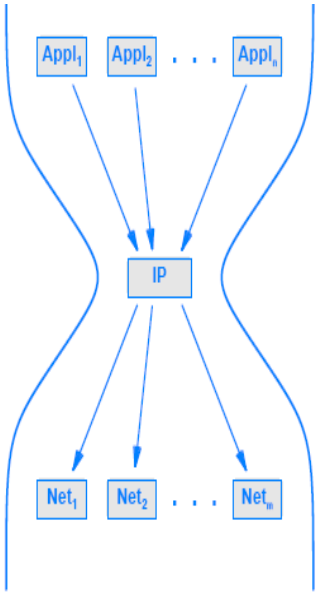
- Quali sono le ragioni per un cambio di protocollo?
  - spazio dell'indirizzamento limitato
    - fornire le stesse opportunita' di crescita anche ai paesi emergenti
  - introdurre funzionalita' per applicazioni che non sono state previste durante la progettazione dell'attuale versione
- Quando IP verra' sostituito
  - la nuova versione dovrebbe avere piu' funzionalita'…
    - ad esempio, supporto del traffico real-time
  - … e dovra' essere sufficientemente flessibile per adattarsi agli usi futuri

Le ragioni del cambiamento

- Una nuova versione di IP dovrebbe includere un indirizzamento piu' complesso e flessibile, e delle nuove funzionalita' legate al routing
- Ad es., Google opera usando molti data center
  - Quando un utente digita "google.com" in un browser, i datagrammi potrebbero essere inviati al data center di Google piu' vicino
    - attualmente, questa funzionalita' viene svolta da DNS e CDN
- Molte applicazioni prevedono che gli utenti collaborino tra loro: per rendere la collaborazione efficiente
  - Internet ha bisogno di meccanismi che permettano la creazione di gruppi e la loro gestione
  - Serve un modo per inviare la copia di un pacchetto ad ogni partecipante del gruppo

Il modello a clessidra e la difficolta' a cambiare

- La scarsita' degli indirizzi e' stata considerata fin dal 1993, quando si e' iniziato a lavorare su una nuova versione di IP
  - tuttavia non vi erano emergenze
  - e IP non e' stato cambiato
- Si pensi all'importanza di IP e al costo del cambiamento
  - modello a clessidra
  - necessario cambiare non solo end host, ma anche router



24

Nomi e Versioni

- Inizialmente noto come “IP The Next Generation”
  - abbreviato in IPng
- Per mantenere la continuita’, e’ stato scelto di utilizzare il nome della versione
  - L’attuale version e’ la 4 (IPv4)
  - La version 5 e’ stata assegnata ad una versione sperimentale successivamente abbandonata
  - La nuova versione e’ dunque ufficialmente la 6 (IPv6)

IPv6: Caratteristiche

- IPv6 mantiene molte delle caratteristiche di IPv4, tra cui
  - Come IPv4, IPv6 e’ connectionless
  - Come IPv4, l’header del datagramma contiene un numero massimo di hop che il datagramma puo’ fare prima di essere scartato
- Molti dettagli, tuttavia, sono cambiati
- Le funzionalita’ possono essere raggruppate in categorie generali (si vedano le prossime slides)

IPv6: Caratteristiche

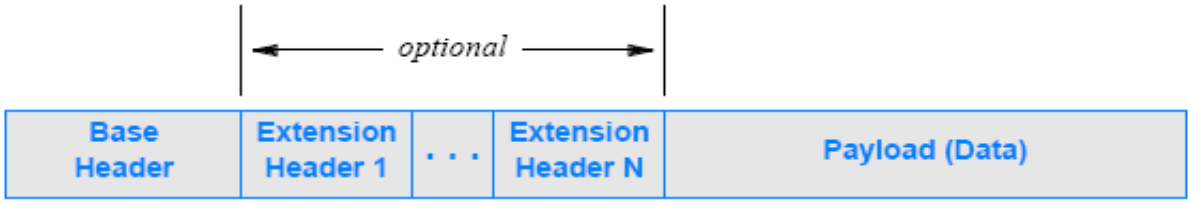
- Dimensione degli indirizzi
  - invece di 32 bit, gli indirizzi di IPv6 sono formati da 128 bit
  - lo spazio di indirizzamento dovrebbe essere sufficiente per contenere eventuali crescite future
    - ci sono circa N_A (= numero di Avogadro = $6*10^{23}$) indirizzi per metro quadro (oceani inclusi)
- Formato dell'header
  - Completamente differente rispetto a IPv4
- Introduzione del concetto di "Extension Header"
  - IPv6 raggruppa le informazioni in header separati
  - Un datagramma consiste in un header IPv6 di base, seguito da nessuno o piu' extension header, seguiti dai dati

IPv6: Caratteristiche

- Supporto del traffico Real-Time
  - introdotto un meccanismo che permetter di creare un cammino tra una sorgente e una destinazione, e di associare i datagrammi a tale cammino
  - utilizzato da applicazioni audio e video
  - o per applicazioni che richiedono una maggiore qualita' del servizio
- Protocollo estensibile
  - IPv6 permette alla sorgente di aggiungere informazioni addizionali al datagramma
  - Lo schema di estensione rende IPv6 piu' flessibile di IPv4
    - e permette di aggiungere funzionalita' se necessario

IPv6: Formato dei datagrammi

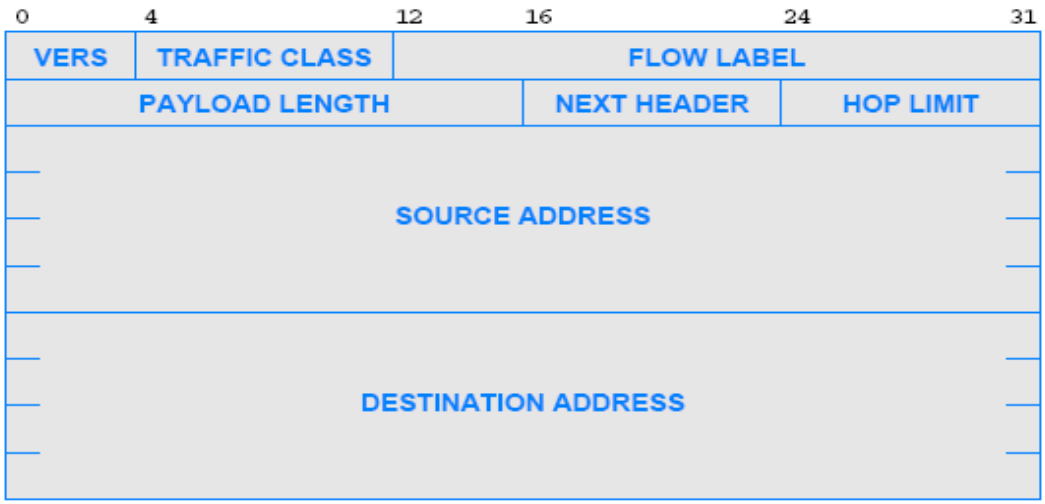
- Un datagramma IPv6 contiene una serie di header
  - ciascun datagramma inizia con un header di base
  - seguito da nessuno o piu' extension header
  - seguito dal payload



- Attenzione: la figura non e' in scala
  - alcuni extension header possono essere piu' grandi dell' header di base
  - la dimensione del payload piu' essere molto piu' grande della dimensione dell' header

IPv6: Formato dell' header di base

- Sebbene l' header IPv6 sia grande il doppio dell' header IPv4, contiene meno campi
- L' header base ha una lunghezza fissa (40 byte)



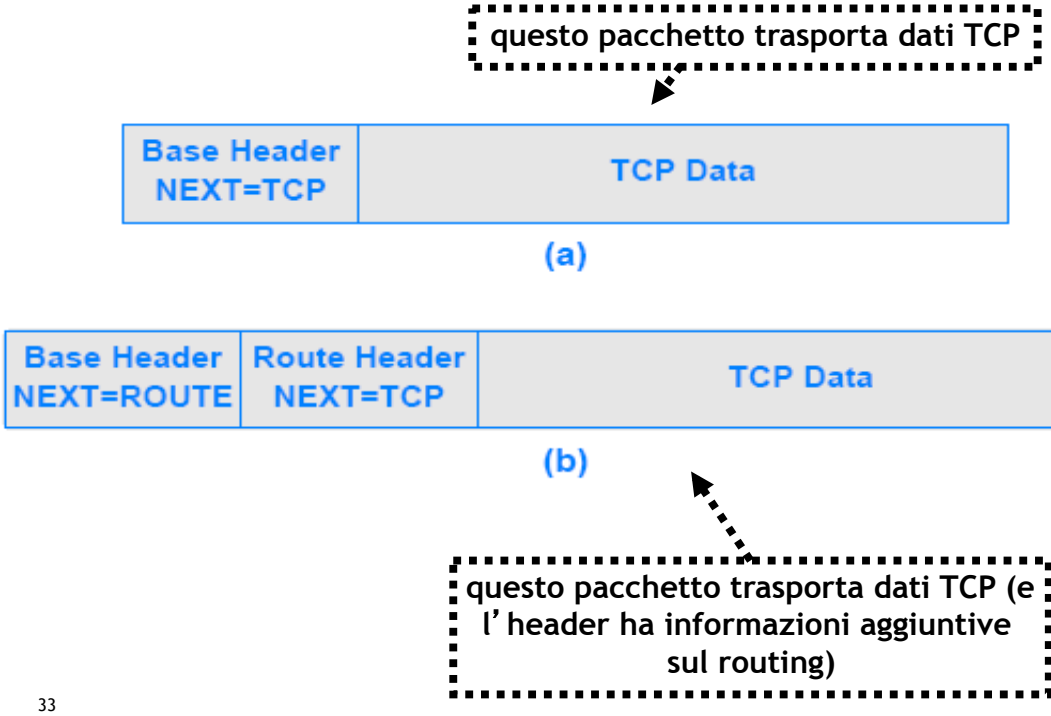
IPv6: Formato dell'header di base

- VERS (Versione: 6)
- TRAFFIC CLASS
  - specifica la classe di traffico in base al tipo di traffico
  - rientra nel framework "differentiated services" per specificare i requisiti che la rete dovrebbe soddisfare
  - Esempi
    - in caso di traffico interattivo (movimenti del mouse) ➔ la sorgente potrebbe specificare una classe con basso ritardo
    - in caso di audio real-time ➔ la sorgente potrebbe specificare un cammino con un jitter basso
- PAYLOAD LENGTH
  - specifica la dimensione del payload (dati trasportati dopo l'header)
  - in IPv4 c'era un campo "total length" che invece includeva l'header

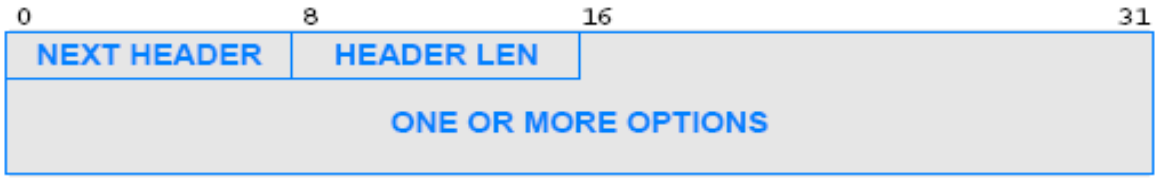
IPv6: Formato dell'header di base

- HOP LIMIT
  - corrisponde al campo TIME-TO-LIVE di IPv4
- FLOW LABEL
  - usato per associare un datagramma con un cammino specifico
- NEXT HEADER
  - campo con un doppio significato: specifica il tipo di informazione che segue l'header corrente
  - Se il datagramma include un extension header
    - il campo NEXT HEADER indica il tipo di extension header
  - Se non ci sono extension header
    - il campo NEXT HEADER specifica il tipo di dati trasportato nel payload

Next Header: esempio



Extension header

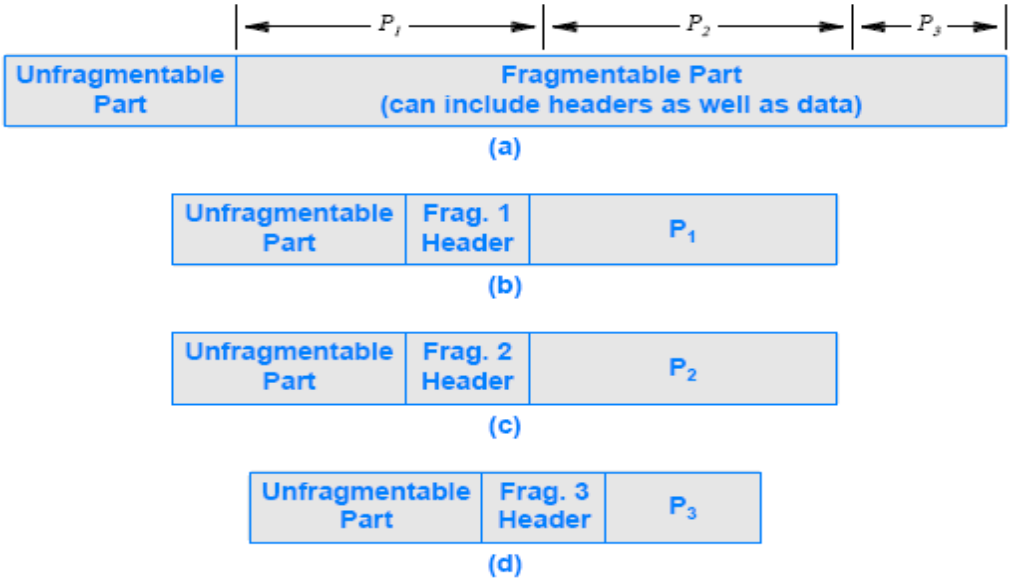


- Non c'e' ambiguita' nell'interpretazione del campo NEXT HEADER
  - lo standard specifica un valore unico per ciascun header possibile
- Un nodo elabora gli header sequenzialmente
  - usando il campo NEXT HEADER per capire cosa segue
- Al di la' dell'header di base (che ha dimensione fissa) in generale gli extension header possono avere dimensione variabile
  - l'header deve contenere le informazioni riguardo la dimensione dell'header stesso

Frammentazione, Riassemblaggio e Path MTU

- Frammentazione in IPv6 e' simile alla frammentazione in IPv4
  - ma ci sono comunque differenze
- Come in IPv4
  - il prefisso in ciascun datagramma viene copiato in ciascun frammento
  - la dimensione del payload viene modificato in base alla dalla Maximum Transmission Unit (MTU) della rete da attraversare
- Diversamente da IPv4
  - non esistono campi predeterminati nell' header di base per la frammentazione
  - bisogna aggiungere un extension header con le informazioni sulla frammentazione
    - la presenza stessa di un extension header di tipo "frammentazione" indica che si tratta di un frammento

Frammentazione, Riassemblaggio e Path MTU



La parte non frammentabile (Unfragmentable Part) e' formata dall' header di base e dagli extension header che controllano il routing

Motivazioni per la definizione di header multipli

- Motivazioni Economiche:
  - si risparmia spazio
  - un datagramma di solito usa solo un sottoinsieme limitato delle funzionalita' e quindi degli extension header
- Motivazioni di estensibilita'
  - e' possibile definire e aggiungere un insieme ampio di funzionalita'
    - senza imporre che tutti gli header abbiano un campo predeterminato
  - IPv4 richiede il cambio dello standard per introdurre nuove funzionalita'
  - In IPv6, invece, basta aggiungere un nuovo extension header

IPv6: indirizzamento

- Come con CIDR, la divisione tra prefisso e suffisso e' arbitraria
- IPv6 introduce il concetto di gerarchia multi-livello
  - Sebbene l'assegnazione degli indirizzi non e' fissa, si puo' assumere che
    - il livello piu' alto corrisponde agli ISP
    - il livello successvio corrsponde ad un' organizzazione (ad es., azienda)
    - il successivo ad un sito specifico, e cosi' via

IPv6: indirizzi speciali

Tipologia	Scopo
unicast	L'indirizzo corrisponde ad un singolo host. Un datagramma inviato a tale indirizzo viene instradato sul cammino minimo
multicast	L'indirizzo corrisponde ad un insieme di host e i membri dell'insieme possono cambiare in qualsiasi momento. Viene consegnata una copia del datagramma a ciascun membro dell'insieme
anycast	L'indirizzo corrisponde ad un insieme di host che condividono un prefisso. Il datagramma viene consegnato ad uno qualsiasi dei memberi dell'insieme (ad es., all'host piu' vicino)

IPv6: notazione esadecimale

- L'indirizzo IPv6 e' composto da 128 bit
  - difficile da gestire "su carta"
- Se venisse usata la notazione decimale puntata, si otterrebbero indirizzi del tipo:
  - 105.220.136.100.255.255.255.255.0.0.18.128.140.10.255.255
- Per ridurre il numero di caratteri da scrivere per indicare un indirizzo
  - e' stata introdotta una notazione esadecimale
    - in inglese: "colon hexadecimal notation" o "colon hex"
  - ciascun gruppo di 16 bit e' scritto in esadecimale separato da ":"
- Nell'esempio precedente:
  - 69DC : 8864 : FFFF : FFFF : 0 : 1280 : 8C0A : FFFF

IPv6: notazione esadecimale

- Le sequenze di zeri si possono “comprimere”
  - Esempio:

    FF0C:0:0:0:0:0:0:B1 → FF0C : : B1
- Tale approccio facilita anche la transizione
  - Mapping degli indirizzi IPv4 esistenti in indirizzi fatto ponendo a zero i primi 96 bit