

Reti di telecomunicazioni

saverio

giugno 2025

Contents

1 Livello 3 rete	2
1.1 Internet Protocol (IP) version 4	2
1.1.1 Introduzione	2
1.1.2 Datagramma IP	3
1.1.3 Indirizzo IP - classful	5
1.1.4 Indirizzo IP - classless CIDR	5
1.1.5 Subnetting	6
1.1.6 Classificazione degli indirizzi IP	6
1.1.7 Variable Length Subnet Mask (VLSM)	6
1.1.8 Indirizzi IP speciali	6
1.1.9 Address Resolution Protocol (ARP) - mappatura IP a MAC	7
1.1.10 Dynamic Host Configuration Protocol (DHCP) - configurazione automatica degli indirizzi IP tramite server DHCP	10
1.1.11 Zero-configuration networking	11
1.1.12 Network Address Translation (NAT) - da IP privato a IP pubblico	11
1.1.13 Internet control message protocol (ICMP) - diagnostica della rete	12
1.2 Algoritmi di routing	13
1.2.1 Introduzione agli algoritmi di routing - tipologie	13
1.2.2 Metriche per il costo del percorso	14
1.2.3 Instradamento gerarchico - protocolli intra e inter AS	14
1.2.4 Distance Vector	15
1.2.5 Algoritmo Bellman-Ford	16
1.2.6 Link State	20
1.2.7 Problemi di routing - routing loop	20
1.2.8 Protocolli di routing intra AS (RIP - OSPF)	21
1.2.9 Protocolli di routing inter-AS - (BGP)	23

Chapter 1

Livello 3 rete

1.1 Internet Protocol (IP) version 4

1.1.1 Introduzione

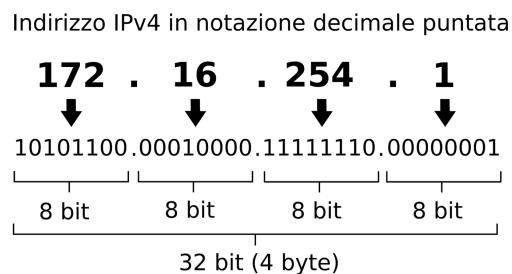


Figure 1.1: Struttura di un indirizzo IP

Al livello 3 ci preoccupiamo di trovare la strada migliore per raggiungere il destinatario.

IPv4 è il protocollo di rete più diffuso al mondo, IP instrada i pacchetti immessi nella rete, ogni nodo in rete ha indirizzo ip.

Ogni pdu(pacchetto) contiene gli indirizzi IP del host mittente e destinatario.

IP è connectionless, ad inoltrare i pacchetti(datagram) sono i router, lo fanno attraverso algoritmi di routing(leggendo l'header del pacchetto).

Ip è inoltre “best-effort”, ossia fa del suo meglio, non garantisce affidabilità massima o ottimale.

Gli indirizzi ip sono da 32 bit(4 numeri decimali compresi tra 0 e 255):

Gli host che appartengono alla stessa rete hanno in comune il livello fisico, un dominio di broadcast è un insieme di computer di una stessa rete(stesso indirizzo ip di rete), esempio rete di casa(collegandoci al router di casa è possibile inviare un messaggio broadcast che arriverà a tutti i dispositivi che fanno parte della rete IP). Le porte del router vengono chiamate interfacce, ogni interfaccia corrisponde ad un host; ogni rete ha un indirizzo IP di rete.

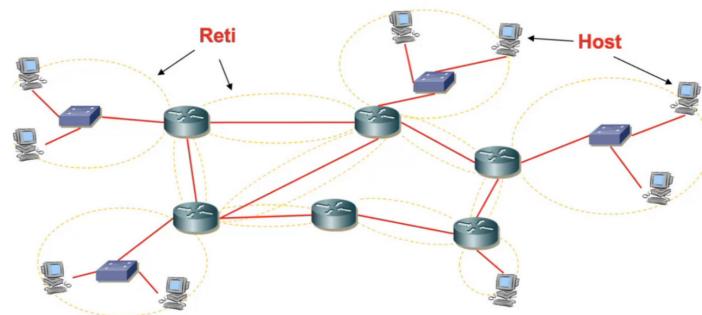


Figure 1.2: Esempio di rete IP di base

1.1.2 Datagramma IP

I datagrams IP sono i pacchetti di dati che vengono inviati attraverso la rete utilizzando il protocollo IP. Ogni datagramma contiene un header e un payload.

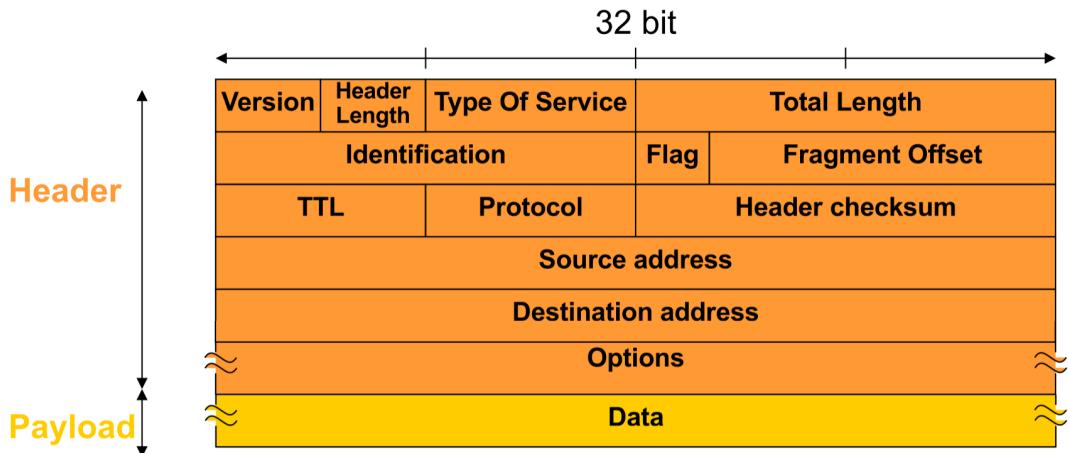


Figure 1.3: Struttura di un datagramma IP

- version: mi dice la versione del protocollo(IPv4 o IPv6) (è di 4 bit, mezzo byte)
- header length: lunghezza del header (4 bit, mezzo byte), se voglio dire che l'header è lungo 5 posso scrivere in questo campo 0101, che è $1 + 0 + 4 + 0$
- type of service(TOS): (1 byte, 8 bit) distingue i tipi diversi di pacchetti, questo è utile nel momento in cui un router deve dare o no la precedenza a certi pacchetti, in base al tipo di servizio. esempio: precedenza al servizio x piuttosto che a y il TOS viene utilizzato per definire il livello di quality of service con cui trattare il pacchetto(primi 6 bit)(gli ultimi 2 servono ad evitare la congestione)



Figure 1.4: Campo Type of Service (ToS) nell'header IP

- total length:(16 bit) lunghezza del datagramma in byte(max 65535 byte)
- identification(16 bit) identifica in modo univoco un datagramma generato dal mittente, quindi anche se il pacchetto viene frammentato, i suoi “frammenti”, ossia altri pacchetti che contengono parte del payload iniziale, avranno lo stesso id. serve quindi a gestire la frammentazione dei pacchetti ip
- options: opzioni per il routing del pacchetto, è un campo poco usato per via del carico di elaborazione molto variabile
- data: è l'informazione da inviare

- flag(3 bit)

il primo bit non usato, il secondo bit è il flag DF (don't fragment: specifica che il router può frammentare il pacchetto o no), il terzo bit è il flag MF (more fragment: indica se il seguente datagramma è l'ultimo dei frammenti). Per utilizzare questi flag metto il valore del bit corrispondente al flag ad 1. Esempio: 010, 0 iniziale è indifferente, 1 indica il flag DF attivo, mentre lo 0 alla fine mi dice che non è MF, quindi che questo datagram è l'ultimo frammento.

- fragment offset(13 bit) mi dice la posizione del frammento all'interno del pacchetto originario, lo spiazzamento. è espresso in multipli di 8 byte.

Frammentazione: quando un datagramma IP deve attraversare una rete con una MTU (Maximum Transmission Unit) inferiore alla sua dimensione, viene suddiviso in frammenti più piccoli. Ogni frammento contiene parte dei dati originali e un header IP con informazioni per il riasssemblaggio.

Nella figura il pacchetto è grande 4000 byte(total length), in questo caso viene frammentato in 3 poichè la sua MTU vale 1500 byte , i frammenti condividono lo stesso id del pacchetto non frammentato.

I frammenti devono avere un offset multiplo di 8.

Solamente il primo frammento ha offset pari a 0, rappresenta la testa. Solamente l'ultimo frammento ha MF pari a 0 poichè sta ad indicare che quello è l'ultimo pezzo del pacchetto, la coda. CHIARIRE



Figure 1.5: Campo Flags nell'header IP

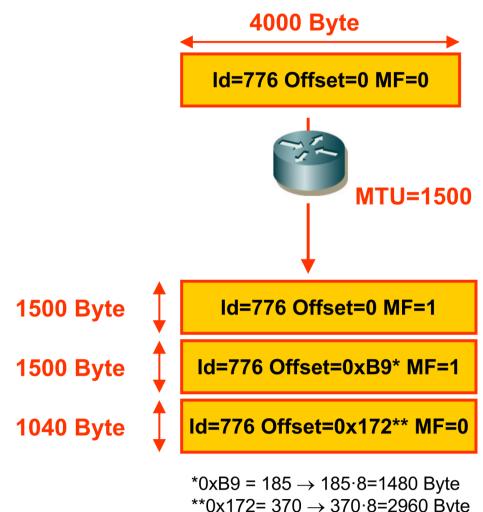


Figure 1.6: Esempio di frammentazione di un datagramma IP

- time to live(8 bit) evita il problema dei pacchetti che entrano in loop poichè non trovano il destinatario. è il tempo di vita del pacchetto, ogni router che riceve il pacchetto(ad ogni hop) decrementa il valore time to live, facendo così arriverà un punto in cui il time to live del pacchetto è stato decrementato fino a zero. quando arriva a 0 il datagram viene scartato. è un valore stabilito dal mittente(esempio 128)
 - protocol(8 bit): contiene un numero che specifica il protocollo usato dalla PDU encapsulata dal datagram. esempio TCP = 6 . UDP = 17 → livello 4
 - header checksum(16 bit): controlla i byte del header, è un controllo veloce su header. se c'è un errore sull'header allora il pacchetto viene scartato, come funziona? : somma 16 bit a 16 bit gli altri byte dell'header, fa la somma dell'eventuale resto e fa il complemento ad 1 del risultato.
- Se non ci sono errori, la somma a 16 bit di tutti questi valori deve dare come risultato 0xFFFF (cioè tutti i bit a 1).
- source/destination address: (32 bit) è l'indirizzo ip del mittente/destinatario

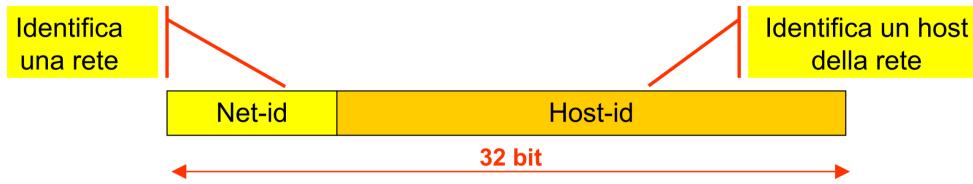


Figure 1.7: Struttura di un indirizzo IP: suddivisione in parte di rete e parte host

Struttura indirizzo IP

Attualmente si utilizza l'indirizzamento classless CIDR (Classless Inter-Domain Routing), che permette di specificare la lunghezza della maschera di rete in bit.

L'assegnazione degli indirizzi è compito dell'Internet Assigned Number Authority (IANA) gestita dall'ICANN (Internet Corporation for Assigned Names and Numbers).

Ogni interfaccia (host) della rete IP ha un suo indirizzo.

1.1.3 Indirizzo IP - classful

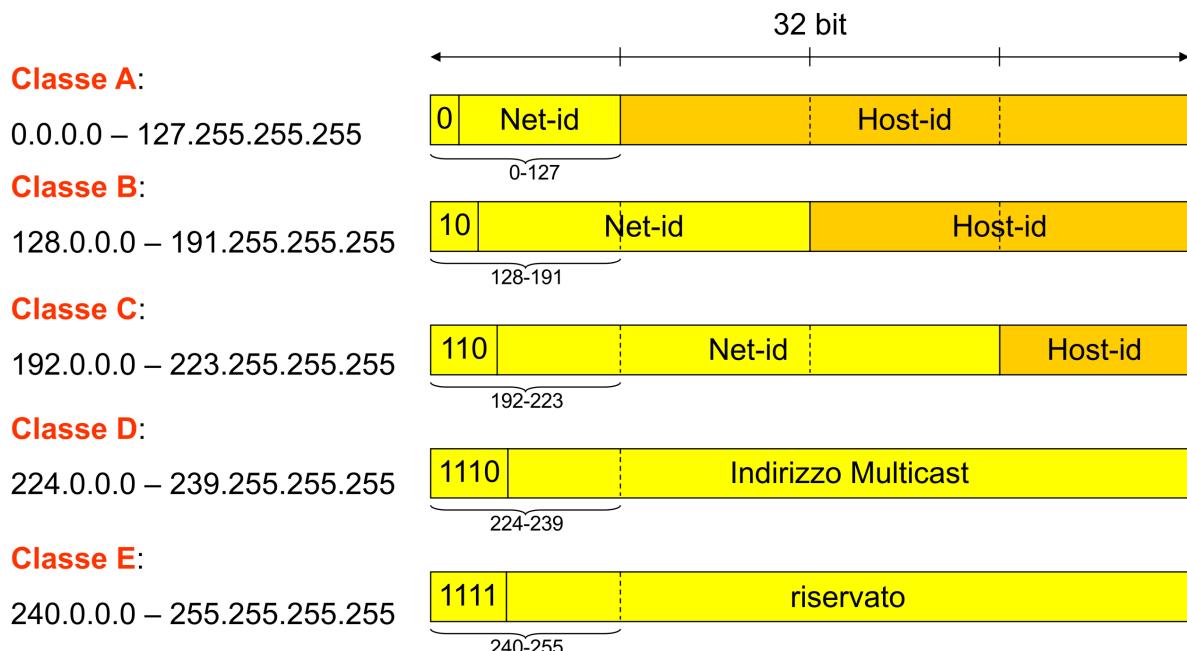
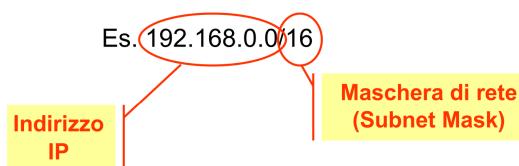


Figure 1.8: Classful Addressing: suddivisione degli indirizzi IP in classi A, B, C, D, E

1.1.4 Indirizzo IP - classless CIDR



Il formato dell'indirizzo è del tipo **a.b.c.d/x** dove **x** rappresenta la lunghezza della maschera di rete in bit. La maschera di rete permette di distinguere la parte di indirizzo relativa alla rete da quella relativa all'host. In notazione CIDR, la maschera viene indicata dopo una barra, ad esempio **192.168.1.0/24**.

Figure 1.9: Esempio di maschera di rete (netmask) in notazione decimale e binaria

Riconoscere un indirizzo di rete e un indirizzo di host Per riconoscere un indirizzo di rete e un indirizzo di host, si utilizza la maschera di rete. La maschera di rete è un numero che indica quanti bit dell'indirizzo IP sono dedicati alla parte di rete e quanti alla parte di host.

- Un indirizzo di rete ha tutti zero nella parte di host.

1.1.5 Subnetting

1.1.6 Classificazione degli indirizzi IP

Pubblici

Privati

Statici

Dinamici

1.1.7 Variable Length Subnet Mask (VLSM)

1.1.8 Indirizzi IP speciali

net ID	subnet ID	host ID	Source ?	Dest.?	Description
0		0	OK	Mai	Questo host su questa rete
0		<i>hostID</i>	OK	Mai	Host specifico su questa rete
127		Qualunque	OK	OK	Indirizzo di loopback
255		255	Mai	OK	Broadcast Limitato localmente (mai inoltrato)
Netid		255	Mai	OK	Broadcast diretto alla rete
Netid	subnetID	255	Mai	OK	Broadcast diretto alla sottorete

Figure 1.10: Indirizzi IP speciali

1.1.9 Address Resolution Protocol (ARP) - mappatura IP a MAC

Con ARP si intende un protocollo di rete appartenente alla suite del protocollo internet (IP) versione 4 e operante a livello di accesso alla rete (livello collegamento se si considera nomenclatura ISO/OSI), il cui compito è fornire la "mappatura" tra l'indirizzo IP (32 bit - 4 byte) e l'indirizzo MAC (48 bit - 6 byte) corrispondente di un terminale in una rete locale ethernet.

Cos'è l'indirizzo MAC?

Un indirizzo MAC (dove MAC sta per Media Access Control), detto anche "indirizzo fisico" o "indirizzo Ethernet", è un codice di 48 bit associato ad ogni dispositivo di rete che implementa lo standard Ethernet. Il suo scopo principale è quello di attribuire un'identità univoca a ciascuno dei nodi collegati ad uno stesso segmento di rete, consentendo quindi comunicazioni locali di tipo unicast.

Al Livello 2 del modello ibrido, i pacchetti viaggiano attraverso il collegamento di più nodi, questi nodi sono caratterizzati da indirizzi fisici detti mac. Invece a livello tre l'indirizzo che possiede il pacchetto è l'indirizzo definitivo!(il destinatario, non i nodi intermedi), la metà ultima del pacchetto. Però questi pacchetti viaggiano attraverso dei nodi, quindi a livello 2 i pacchetti avranno come destinazioni i nodi, e per raggiungerli hanno bisogno di conoscere gli indirizzi fisici dei nodi, perciò tramite gli indirizzi MAC e il protocollo arp si risolve il problema locale del collegamento tra i vari nodi che fanno sì che il pacchetto viaggi attraverso quelli.

ARP risponde quindi a questa domanda: ho questo indirizzo ip, mi dici quale sia l'indirizzo di livello due corrispondente a questo ind ip??

Mi dice l'indirizzo del "prossimo nodo", livello 2.

Quindi un protocollo fondamentale che individua da un indirizzo ip(liv 3) uno o più di livello 2

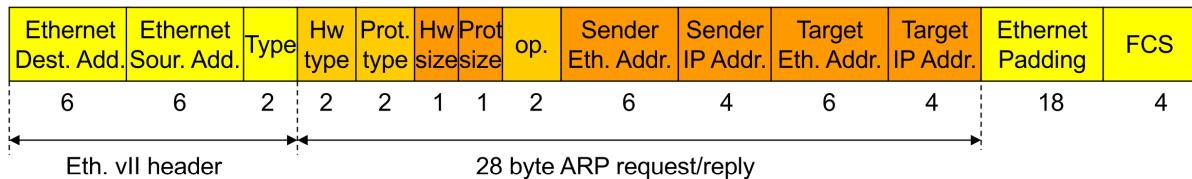


Figure 1.11: Struttura di un frame Ethernet

Questo è il frame ethernet: in giallo ho l'header e la coda di ethernet , la coda svolge il ruolo di verificare che ci siano errori(fcs da 4 byte), il campo padding in coda serve a raggiungere i 64 byte(è un filler) minimi del frame totale tra ethernet e arp(arpa è interno a 1 frame ethernet) il frame deve avere una dimensione minima di 64 byte, il protocollo arp è quello arancione chiaro e scuro al centro.

Struttura del frame ethernet:

- ethernet destination address: ind liv 2 del mittente - uguale a sender eth address
- ethernet source address
- type: tipologia tecnologia del livello 2
- hw type: specifica l'hardware che si sta usando al livello 2
- protocol type: specifica che protocollo si vuole convertire dal liv 3 al 2(quindi non ha utilizzo esclusivo per indirizzi ip)
- hw size: dimensione indirizzo di liv 3
- prot size: dimensione indirizzo liv 2
- options:
- sender ethernet address: indirizzo liv 2 del mittente
- sender ip address: indirizzo liv 3 del mittente
- target ethernet address: indirizzo liv 2 del destinatario
- target ip address: indirizzo liv 3 del destinatario
- op: operazione che si vuole effettuare, 1 per richiesta arp, 2 per risposta arp

Esempio ARP all'interno di una rete

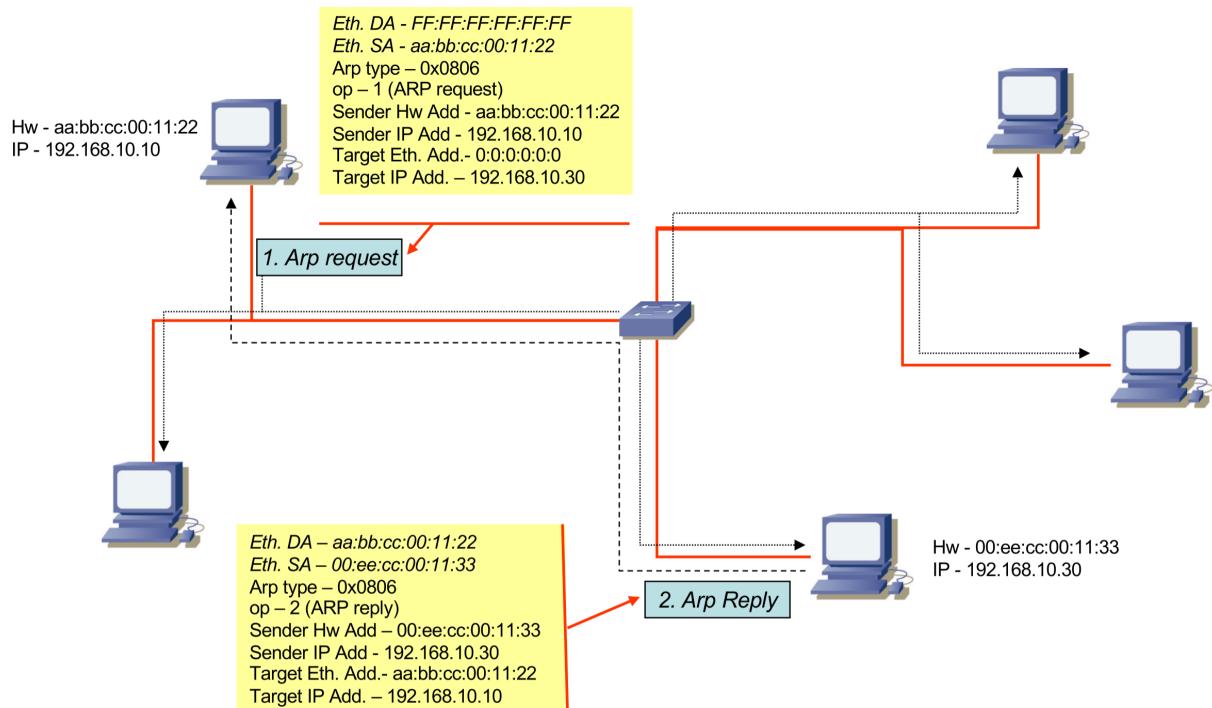


Figure 1.12: Esempio di funzionamento del protocollo ARP

L'indirizzo hw è l'indirizzo fisico anche detto mac, è tipicamente in esadecimali ed è di 6 byte.

I riquadri in giallo sono pacchetti ARP utili alla comunicazione tra i due pc(visto che il dispositivo sta utilizzando arp per inviare un pacchetto allora sta usando un arp request, che viene segnato come op - 1 nel frame ethernet); i pacchetti in giallo quindi encapsulano una serie di informazioni del frame ethernet.

Inoltre l'Ethernet destination address viene impostato a ff:ff:ff:ff:ff:ff (broadcast per indirizzi ARP) se viene effettuata un'ARP Request, questo perchè il mittente sta cercando di individuare l'indirizzo fisico del destinatario, di cui conosce solo l'ip, perciò utilizza ARP.

Nel momento della ARP request non so l'indirizzo ethernet del destinatario, ma so solo l'indirizzo ip, quindi ho tutti i byte del Ethernet address del target a 0.

Manderò quindi in broadcast(quindi multicast: a tutti i dispositivi della rete)(linee non tratteggiate) una richiesta a tutti i computer della rete, aspettando che il computer con l'indirizzo ip corrispondente a quello del target mi invii in unicast(linea tratteggiata) con un altro pacchetto ARP(arp reply, op - 2).

A questo punto il computer che ha inviato inizialmente l'ARP request riceve le info necessarie per poter inviare il pacchetto al destinatario.

Esempio ARP in reti diverse

IMPORTANTE: IL TCP(livello 4) si disinteressa di questo percorso, opera come se A e B fossero già collegati(end to end).

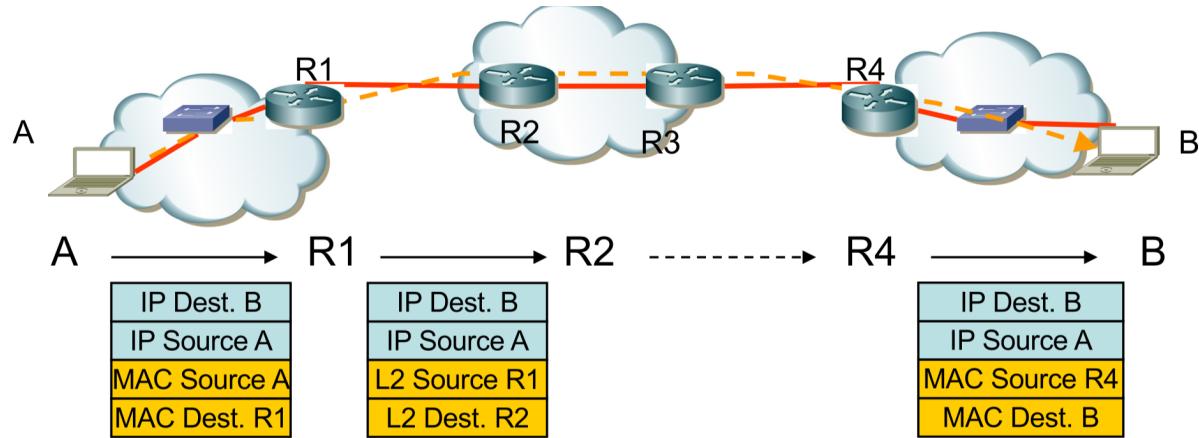


Figure 1.13: Esempio di funzionamento del protocollo ARP tra reti diverse

Default gateway e tabelle di routing per scoprire se la rete del destinatario è la stessa da cui mando la richiesta arp: applico la mia maschera di rete all'ip del destinatario (la mia maschera di rete è un'informazione che posseggo). quindi se la rete del destinatario non corrisponde a quella del mittente, per "uscire" e connettermi all'altra rete devo passare per il router. gli indirizzi di livello 3, quindi l'ip del destinatario e l'ip del mittente sono info che non variano tra un passaggio da un router all'altro, mentre quelli in giallo, gli indirizzi fisici(mac) variano tra router all'altro. quindi il protocollo arp verrà applicato ad ogni passaggio tra un router all'altro cambiando quindi i mac source e mac destinatario, inserendo i mac dei dispositivi(router) tra cui avviene questo passaggio d'informazione. fino ad arrivare al dispositivo con l'indirizzo ip contenuto nel frame (ip dest B) Il primo router a cui ci appoggiamo per l'inolitramento del pacchetto viene chiamato default gateway.

I router servono a connettere reti diverse, quindi se il destinatario non è nella mia rete allora mi appoggio al router, che ha un indirizzo ip di default, che è il default gateway.

Le tabelle di routing sono delle tabelle che i router utilizzano per sapere come inoltrare i pacchetti verso le reti di destinazione. Queste tabelle contengono informazioni sulle reti raggiungibili, i loro indirizzi IP e fisici.

A fine processo B legge i livelli 2 e 3 e riconosce i propri indirizzi MAC e IP, a quel punto può leggere il payload del pacchetto, che contiene i dati che A voleva inviare a B.

ARP nella stessa rete Se nel processo in cui cerco di capire se il destinatario sta nella mia rete o no scopro che sta nella mia rete allora non mi serve il router e posso inviare direttamente il pacchetto al destinatario, ma prima devo scoprire il suo indirizzo fisico(mac), quindi applico arp come fatto inizialmente.

1.1.10 Dynamic Host Configuration Protocol (DHCP) - configurazione automatica degli indirizzi IP tramite server DHCP

Un indirizzo ip può essere statico o dinamico, servers e router hanno tipicamente indirizzi statici.

Gli ind statici si configurano manualmente, quelli dinamici invece sono configurati tramite DHCP, un server che va configurato dall'utente ed è utile perché fornisce:

- indirizzo ip del dispositivo
- netmask
- indirizzo ip del router
- indirizzo ip del server DNS
- lease time

Il lease time è il tempo in cui l'indirizzo ip è riservato al dispositivo, dopo di che il dispositivo deve richiedere un nuovo indirizzo ip; è anche possibile ottenere un indirizzo IP riservato, sulla base dell'indirizzo fisico MAC.

DHCP relay - inoltro delle richieste DHCP Se nella rete locale non è disponibile un DHCP server, allora la richiesta può essere inoltrata verso un'altra rete che dispone della funzione DHCP relay.

DHCP - funzionamento Questo protocollo fa uso dei UDP per la comunicazione tra client e server, e funziona in modo simile a ARP.

L'host che desidera un indirizzo IP si affida a questo protocollo per ottenerlo, inviando un messaggio DHCP discover, incapsulandolo in un pacchetto UDP, che viene inviato in broadcast a tutti i dispositivi della rete locale.

All'interno del pacchetto UDP, il mittente inserisce i seguenti indirizzi IP:

- IP source address: l'indirizzo IP del mittente, che non è ancora stato assegnato, viene impostato momentaneamente a 0.0.0.0
- IP destination address: l'indirizzo IP del server DHCP, viene impostato a 255.255.255.255

Tutti gli eventuali server DHCP presenti nella rete rispondono in broadcast con un messaggio DHCP offer, che contiene le informazioni richieste dal client, come l'indirizzo IP assegnato, la netmask ecc...

Il client riceve le offerte dai server DHCP e sceglie una di esse, inviando un messaggio DHCP request al server scelto.

Infine il server DHCP conferma l'assegnazione dell'indirizzo IP con un messaggio DHCP ack.

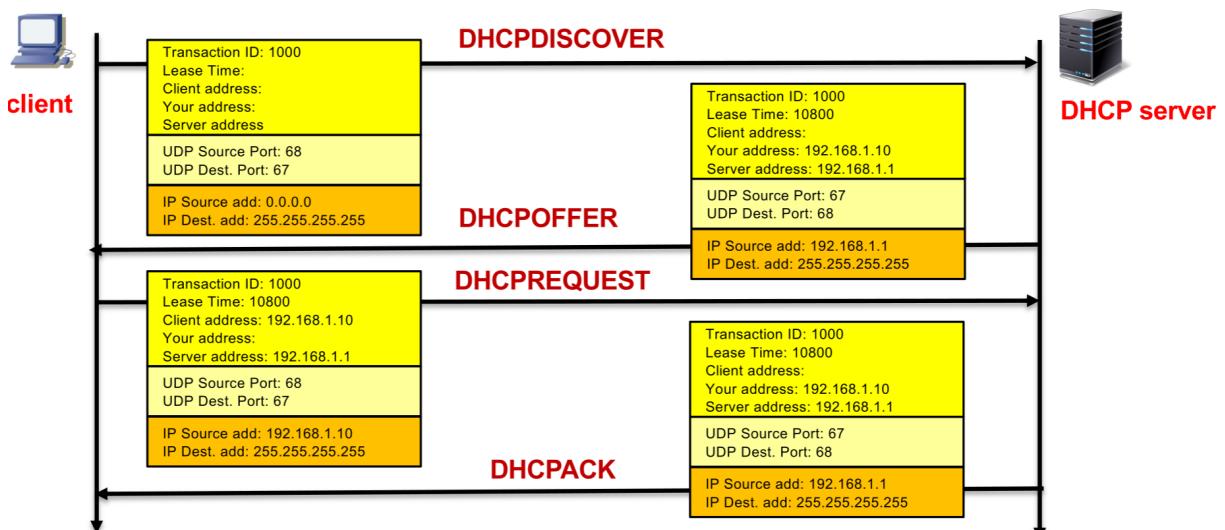


Figure 1.14: Funzionamento del protocollo DHCP

1.1.11 Zero-configuration networking

Il zero-configuration networking è la configurazione della rete in assenza di server e amministratori, è ideale solo per reti piccole e semplici. Non è quindi un protocollo alternativo a DHCP.

Link-Local Address La IANA ha riservato a questo scopo un Link-Local Address: 169.254.0.0/16

Gli indirizzi IP dei dispositivi nella rete vengono assegnati automaticamente, all'interno di un range specifico: [169.254.1.1 - 169.254.254]

ARP probe Quando un dispositivo si connette alla rete, invia un ARP probe per verificare se l'indirizzo IP scelto è già in uso. Se non riceve risposta, assume che l'indirizzo sia disponibile e lo utilizza. Se un altro host ha lo stesso indirizzo, sceglie casualmente un altro indirizzo IP all'interno dello stesso range.

ARP announcement Dopo aver scelto un indirizzo IP, il dispositivo invia un ARP announcement per informare gli altri dispositivi della rete del suo nuovo indirizzo.

Questi indirizzi sono di tipo riservato e non possono essere usati per comunicare fuori dalla rete locale.

1.1.12 Network Address Translation (NAT) - da IP privato a IP pubblico

Il NAT è un software eseguito dal router di frontiera (ha il compito di far uscire dalla rete locale i pacchetti destinati ad indirizzi ip di altre reti).

Quindi è quel servizio che converte l'indirizzo privato del router di frontiera, in uno pubblico, così da comunicare con "l'esterno". Perciò i router di frontiera delle altre reti non vedono l'indirizzo privato ma quello pubblico, convertito grazie al NAT.

Più precisamente, il NAT è una famiglia di tecniche/protocolli.

Funzionamento del NAT e tabelle di traduzione Tutti i datagram IP che viaggiano da (verso) la LAN hanno il medesimo indirizzo IP mittente (destinazione), poiché il router di frontiera ha un solo indirizzo IP pubblico, che viene utilizzato per tutte le comunicazioni in uscita (entrata) dalla rete locale(LAN).

Il NAT utilizza una tabella di traduzione per mappare gli indirizzi IP privati della rete locale con l'indirizzo IP pubblico del router di frontiera; utilizzando le porte come ulteriore identificativi.

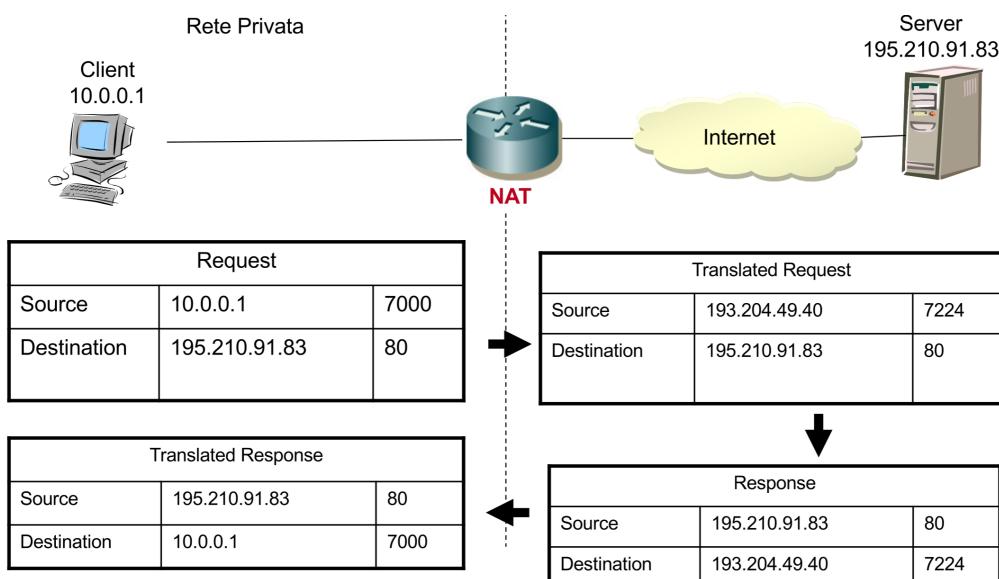


Figure 1.15: Funzionamento del NAT: traduzione degli indirizzi IP privati in indirizzi IP pubblici

1.1.13 Internet control message protocol (ICMP) - diagnostica della rete

I messaggi ICMP si dividono messaggi di errore e messaggi di richiesta; vengono inviati attraverso i pacchetti IP(protocol type 1); questo protocollo serve a segnalare informazioni di controllo relative al livello di rete, un messaggio ICMP è composto dai campi type e code:

- type: indica il tipo di messaggio ICMP
- code: dettagli sul tipo di messaggio

ICMP type	Code	Description
0	0	Echo replay – risposta al messaggio di eco (ping)
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	6	Destination network unknown
3	7	Destination host unknown
4	0	Source quench – riduzione data rate
8	0	Echo request
9	0	Router advertisement
10	0	Router discovery
11	0	TTL expired
12	0	IP header bad

Figure 1.16: Principali comandi ICMP e loro utilizzo

Esempio utilizzo di ICMP - ping Ad esempio, echo request e echo reply sono alla base del comando ping. Type 8 corrisponde a echo request, mentre type 0 corrisponde a echo reply.

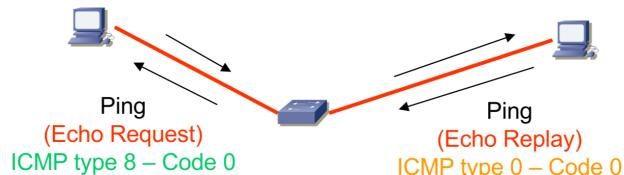


Figure 1.17: Esempio di funzionamento di ICMP tramite il comando ping

Esempio utilizzo di ICMP - traceroute Tramite il traceroute(utilizzo dei ICMP type 11) scopro il percorso che sta seguendo il pacchetto comandi: traceroute sitoweb

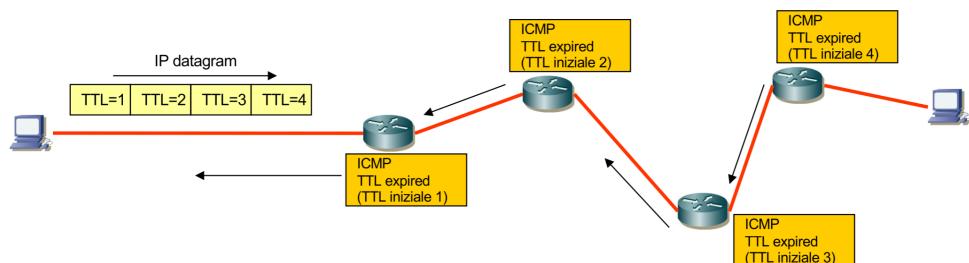


Figure 1.18: Esempio di funzionamento di ICMP tramite il comando traceroute

1.2 Algoritmi di routing

1.2.1 Introduzione agli algoritmi di routing - tipologie

Gli algoritmi di routing sono utilizzati dai router per determinare il percorso migliore che i pacchetti devono seguire per raggiungere la loro destinazione.

Questi algoritmi possono essere classificati in base al loro uso o meno delle tabelle di routing o se sono gerarchici.

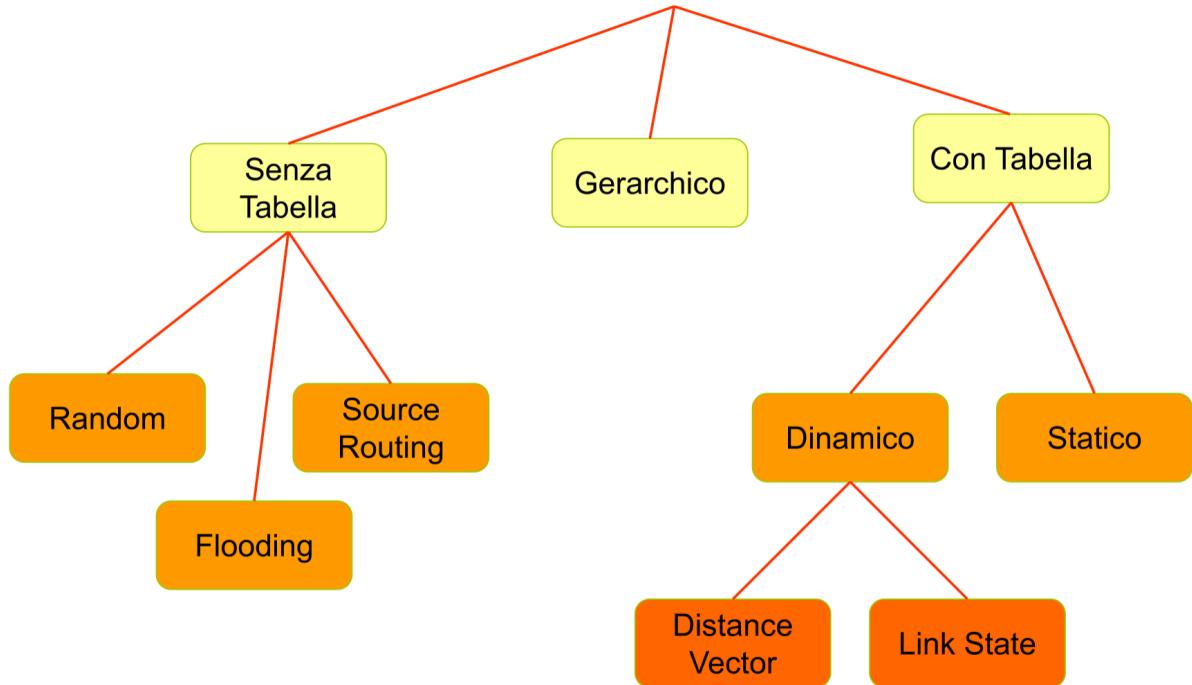


Figure 1.19: Tipologie di algoritmi di routing

- Con tabella: il router conserva in memoria una tabella per capire qual è la strada migliore, contiene dati per scegliere la strada migliore. Come costo si intende il numero di hop, la banda, l'affidabilità,

Indirizzo Rete Destinazione	Costo	Linea di Uscita/Next hop
192.168.59.0/24	L1
....
Default Route	L2

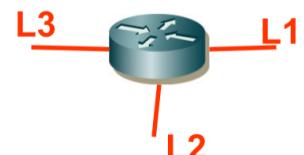


Figure 1.20: Esempio di tabella di routing

il ritardo o anche un valore inserito manualmente; dipende strettamente dal metodo che si vuole utilizzare.

- Senza tabella: includono il routing random, il flooding e il source routing
- Gerarchici: i router affidano le scelte di routing ad altri router "più intelligenti"?

!Instradare significa scegliere il percorso migliore per raggiungere la destinazione, il forwarding invece è l'azione di inviare il pacchetto per la strada scelta!

1.2.2 Metriche per il costo del percorso

Il costo del percorso viene calcolato sulla base di diverse metriche:

- **Bandwidth**: la capacità di un link (es. generalmente un link a 10 Mbps è preferibile rispetto ad una linea a 64 kbps)
- **Delay**: il tempo necessario ad ogni pacchetto per andare dalla sorgente alla destinazione
- **Load**: il carico di lavoro degli elementi della rete come i router o i link
- **Reliability**: l'affidabilità, generalmente riferita al tasso di errore di ogni singolo link
- **Hop count**: il numero di router che un pacchetto deve attraversare per raggiungere la destinazione
- **Ticks**: il ritardo di un collegamento dati in multipli di un *IBM PC clock tick* (circa 55 ms)
- **Cost**: un valore arbitrario, generalmente basato sulla banda, sul costo economico di un link, o su altre misure stabilite dall'amministratore di rete

Cosa fa il router? ?

1.2.3 Instradamento gerarchico - protocolli intra e inter AS

Al fine di comprendere il routing è indispensabile introdurre il concetto di Autonomous System(AS).

Un AS è l'insieme di router amministrati dallo stesso gestore; è possibile distinguere i vari protocolli di routing proprio sulla base di questo.

Infatti i protocolli che lavorano all'interno di un AS vengono chiamati intra AS, mentre quelli che operano tra Autonomous Systems differenti vengono detti inter AS.

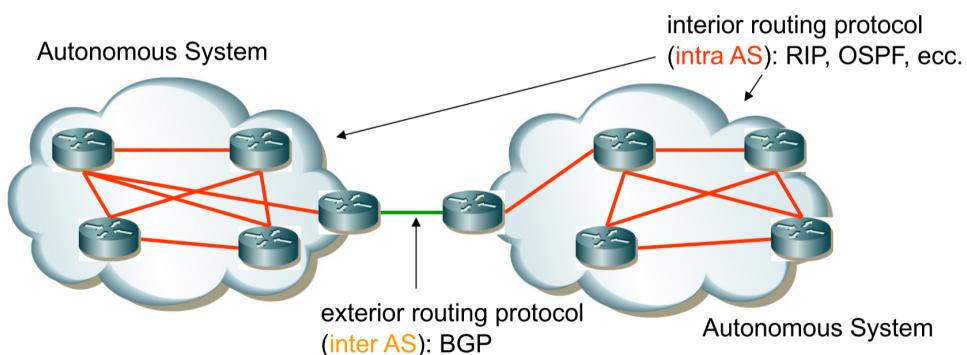


Figure 1.21: Esempio di gerarchia tra Autonomous Systems (AS)

1.2.4 Distance Vector

È una famiglia di algoritmi di routing in cui ogni router scambia informazioni con i vicini, usa l'algoritmo di Bellman-Ford per calcolare il percorso più breve. Un algoritmo di routing che fa uso di tabelle dinamiche è il distance vector:

ogni router prende decisioni in modo autonomo; ogni router invia in modo periodico ai router vicini i "vettori delle distanze" (distance vector).

Le colonne della tabella che interessano particolarmente questo algoritmo sono quella dei costi e quella delle destinazioni note.

In questo modo tra i router ci si scambiano informazioni utili al routing stesso, prendendo informazioni dai router vicini

Il vettore distanza quindi comunica ai vicini i percorsi che conosce e il costo di tali percorsi (percorsi che passano dal router che condivide tali percorsi) le tabelle sono dinamiche perché la topologia della rete può cambiare nel tempo.

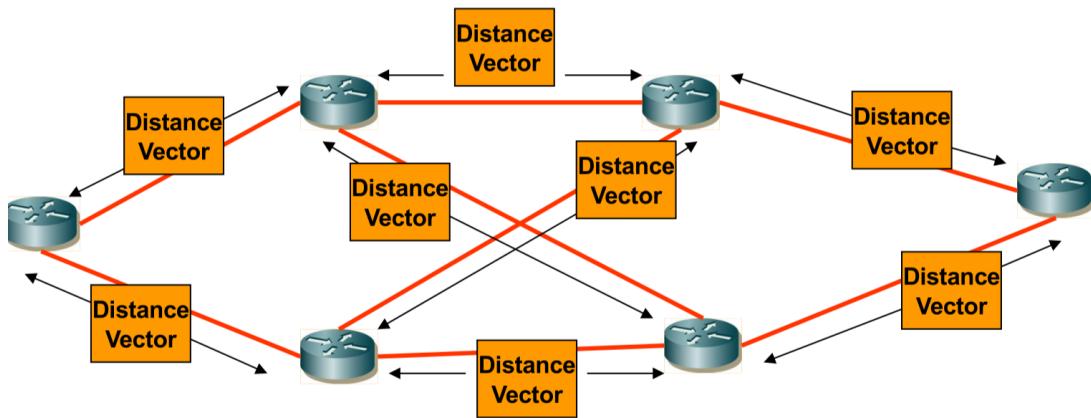


Figure 1.22: Esempio di funzionamento dell'algoritmo Distance Vector

1.2.5 Algoritmo Bellman-Ford

Il funzionamento dell'algoritmo di routing distance-vector porta ad una soluzione che si otterrebbe equivalentemente applicando l'algoritmo di Bellman-Ford per la ricerca del cammino a costo minimo in un grafo.

I router non applicano questo algoritmo, ma la soluzione finale cui si giunge è identica.

Questo algoritmo lavora su una rappresentazione globale del grafo, cosa che nei router reali non accade poiché non conoscono tutta la rete, ma solo le info sui e dei vicini diretti.

I protocolli distance vector sono una versione distribuita del Bellman-Ford, cioè: l'algoritmo non è eseguito da un singolo punto centrale, ma ogni nodo della rete (es. ogni router) esegue una parte dell'algoritmo!

Albero dei cammini a costo minimo

Variabili

- s è il router da cui parte l'algoritmo e di cui interessa calcolare il costo minimo
- h indica il numero dei salti
- j indica un nodo
- i indica un nodo differente da j
- $D_j^{(h)}$ è il costo del cammino minimo tra la sorgente s e il nodo j con massimo h salti
- d_{ij} è il costo del collegamento diretto tra i nodi i e j ; questo costo è infinito se i nodi non sono collegati direttamente

Fasi dell'algoritmo

1. Inizializzazione: $h = 0$

- La distanza del nodo da se stesso $D_s^{(0)} = 0$
- La distanza tra s e i nodi non diretti è infinita $D_j^{(0)} = \infty$ per tutti $j \neq s$

2. Iterazione: per ogni $h > 0$

Hop successivo $h = h + 1$

Calcolo dei cammini al salto successivo $D_j^{(h)} = \min_i [D_i^{(h-1)} + d_{ij}^{(h-1)}]$

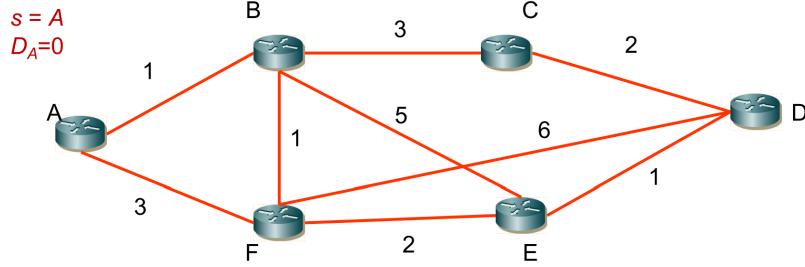
3. Verifica: tramite questo controllo capisco se il cammino $D_j^{(h)}$ calcolato è uguale al precedente, se così fosse allora non ha senso continuare e si setta come h_{max} quello precedente, si ferma l'algoritmo; in caso contrario l'algoritmo prosegue dal passaggio due.

If $D_j^h = D_j^{h-1} \quad \forall j \neq s$

then $h_{max} = h - 1$; stop

else go to 2

Esempio Il primo passo è calcolare i costi dei cammini quando $h = 0$, ricordando che quando non c'è collegamento diretto avrà costo infinito.



$$D_B = D_C = D_D = D_E = D_F = \infty$$

Figure 1.23: Esempio di applicazione dell'algoritmo di Bellman-Ford - Passo 1

Poi si calcolano i costi dei cammini quando $h = 1$, quindi con al massimo un salto.

- $D_B^{(1)} = \min[D_A^{(0)} + d_{AB}, D_C^{(0)} + d_{CB}, D_D^{(0)} + d_{DB}, D_E^{(0)} + d_{EB}, D_F^{(0)} + d_{FB}, D_B^{(0)}] = \min[0 + 1, \infty + 3, \infty + \infty, \infty + 5, \infty + 1, \infty] = 1$
- $D_C^{(1)} = \min[D_A^{(0)} + d_{AC}, D_B^{(0)} + d_{BC}, D_D^{(0)} + d_{DC}, D_E^{(0)} + d_{EC}, D_F^{(0)} + d_{FC}, D_C^{(0)}] = \min[0 + 3, \infty + 3, \infty + \infty, \infty + 2, \infty + \infty, \infty] = 3$
- $D_D^{(1)} = \min[D_A^{(0)} + d_{AD}, D_B^{(0)} + d_{BD}, D_C^{(0)} + d_{CD}, D_E^{(0)} + d_{ED}, D_F^{(0)} + d_{FD}, D_D^{(0)}] = \min[0 + \infty, \infty + \infty, \infty + \infty, \infty + 1, \infty + 5, \infty] = \infty$
- $D_E^{(1)} = \min[D_A^{(0)} + d_{AE}, D_B^{(0)} + d_{BE}, D_C^{(0)} + d_{CE}, D_D^{(0)} + d_{DE}, D_F^{(0)} + d_{FE}, D_E^{(0)}] = \min[0 + \infty, \infty + 5, \infty + 2, \infty + 1, \infty + \infty, \infty] = \infty$
- $D_F^{(1)} = \min[D_A^{(0)} + d_{AF}, D_B^{(0)} + d_{BF}, D_C^{(0)} + d_{CF}, D_D^{(0)} + d_{DF}, D_E^{(0)} + d_{EF}, D_F^{(0)}] = \min[0 + \infty, \infty + 1, \infty + \infty, \infty + 5, \infty + \infty, \infty] = \infty$

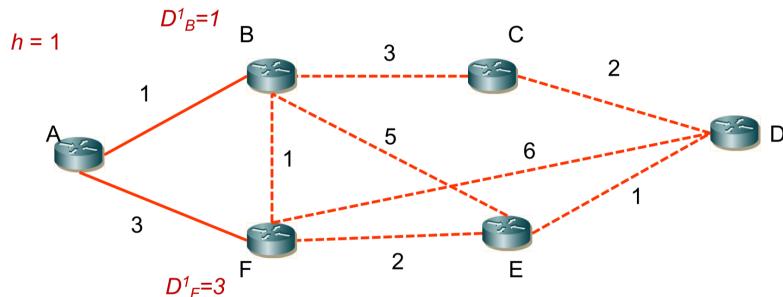


Figure 1.24: Esempio di applicazione dell'algoritmo di Bellman-Ford - Passo 2

- $D_B^{(2)} = \min[0 + 1, 3 + 3, \infty + \infty, \infty + 5, \infty + 1, 1] = 1$
- $D_C^{(2)} = \min[0 + 3, 1 + 3, \infty + \infty, \infty + 2, \infty + \infty, 3] = 3$
- $D_D^{(2)} = \min[0 + \infty, 1 + \infty, 3 + \infty, \infty + 1, \infty + 5, \infty] = \infty$
- $D_E^{(2)} = \min[0 + \infty, 1 + 5, 3 + 2, \infty + 1, \infty + \infty, \infty] = 5$

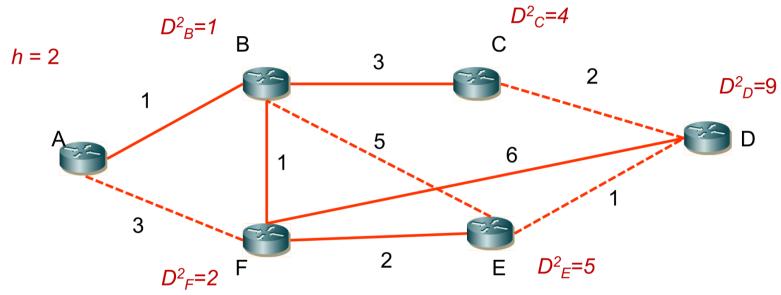


Figure 1.25: Esempio di applicazione dell'algoritmo di Bellman-Ford - Passo 3

- $D_F^{(2)} = \min[0 + \infty, 1 + 1, 3 + \infty, \infty + 5, \infty + \infty, \infty] = 2$
- $D_B^{(3)} = \min[0 + 1, 3 + 3, \infty + \infty, 5 + 5, 2 + 1, 1] = 1$
- $D_C^{(3)} = \min[0 + 3, 1 + 3, \infty + \infty, 5 + 2, 2 + \infty, 3] = 3$
- $D_D^{(3)} = \min[0 + \infty, 1 + \infty, 3 + \infty, 5 + 1, 2 + 5, \infty] = 6$
- $D_E^{(3)} = \min[0 + \infty, 1 + 5, 3 + 2, \infty + 1, 2 + \infty, 5] = 5$
- $D_F^{(3)} = \min[0 + \infty, 1 + 1, 3 + \infty, \infty + 5, 5 + \infty, 2] = 2$

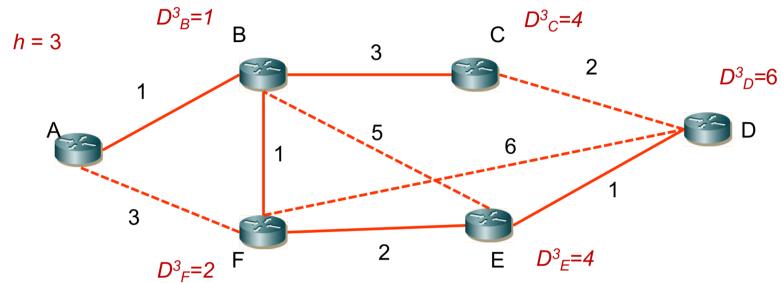


Figure 1.26: Esempio di applicazione dell'algoritmo di Bellman-Ford - Passo 4

Poi si calcolano i costi dei cammini quando $h = 4$

- $D_B^{(4)} = \min[0 + 1, 3 + 3, 6 + \infty, 5 + 5, 2 + 1, 1] = 1$
- $D_C^{(4)} = \min[0 + 3, 1 + 3, 6 + \infty, 5 + 2, 2 + \infty, 3] = 3$
- $D_D^{(4)} = \min[0 + \infty, 1 + \infty, 3 + \infty, 5 + 1, 2 + 5, 6] = 6$
- $D_E^{(4)} = \min[0 + \infty, 1 + 5, 3 + 2, 6 + 1, 2 + \infty, 5] = 5$
- $D_F^{(4)} = \min[0 + \infty, 1 + 1, 3 + \infty, 6 + 5, 5 + \infty, 2] = 2$

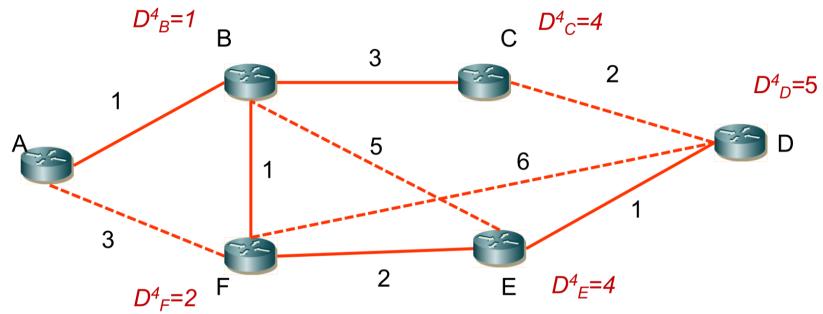


Figure 1.27: Esempio di applicazione dell'algoritmo di Bellman-Ford - Passo 5

Poi si calcolano i costi dei cammini quando $h = 5$, quindi con al massimo un salto.

- $D_B^{(5)} = \min[0 + 1, 3 + 3, 6 + \infty, 5 + 5, 2 + 1, 1] = 1$
- $D_C^{(5)} = \min[0 + 3, 1 + 3, 6 + \infty, 5 + 2, 2 + \infty, 3] = 3$
- $D_D^{(5)} = \min[0 + \infty, 1 + \infty, 3 + \infty, 5 + 1, 2 + 5, 6] = 6$
- $D_E^{(5)} = \min[0 + \infty, 1 + 5, 3 + 2, 6 + 1, 2 + \infty, 5] = 5$
- $D_F^{(5)} = \min[0 + \infty, 1 + 1, 3 + \infty, 6 + 5, 5 + \infty, 2] = 2$

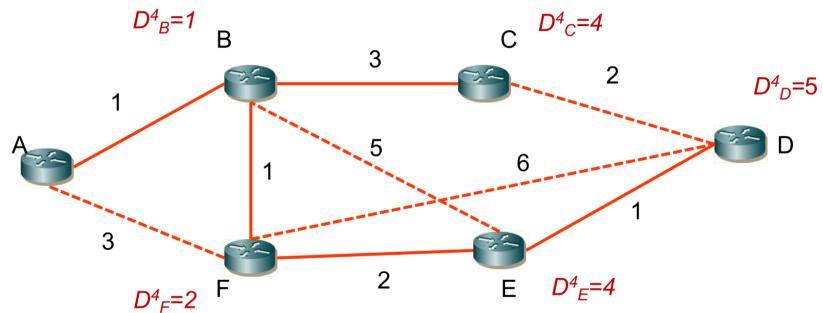


Figure 1.28: Esempio di applicazione dell'algoritmo di Bellman-Ford - Passo 6

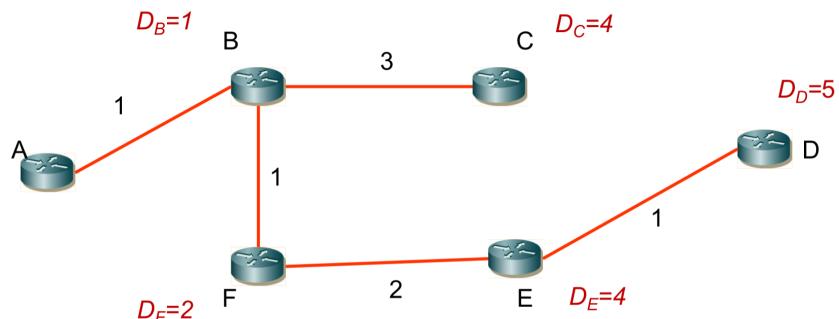


Figure 1.29: Albero dei cammini a costo minimo

1.2.6 Link State

L'algoritmo link-state rappresenta un'alternativa al vistance vector, si utilizza per reti di dimensioni maggiori.

Il presupposto su cui si basa questo algoritmo è che i router conoscano la topologia della rete, non come avveniva nel distance vector; infatti nel distance vector i router conoscevano solamente le info sui vicini, nel link state invece i router conoscono l'intera topologia della rete.

Flooding Si sfrutta il concetto di flooding: il flooding è un protocollo di instradamento usato dai router che inoltrano un pacchetto in ingresso su tutte le linee ad eccezione di quella da cui proviene. Ogni pacchetto in arrivo viene inoltrato su ogni linea di uscita eccetto quella da cui è arrivato; quindi tutti i router della rete comunicano in flooding tutte le informazioni dei vicini, con relativi costi. Inoltre il link state periodicamente ritrasmette l'informazione.

La topologia della rete è cosa nota a tutti e quindi si possono applicare strategie proprie della teoria dei grafi, più nello specifico applichiamo l'algoritmo di Dijkstra (complessità non lineare, $n \log(n)$, ossia con l'aumentare dei nodi la complessità di questo algoritmo aumenta alla $n \log(n)$): permette di trovare la strada nel modo più efficiente possibile (considerando tempo e costo: migliore end to end).

Algoritmo di Dijkstra

1.2.7 Problemi di routing - routing loop

Soluzioni: split horizon, poison reverse, hold-down

Tramite split horizon decido che i router non condividono con i vicini le informazioni ottenute da altri vicini c'è anche lo split horizon con poisoned reverse, invece di non comunicare ai vicini le info di altri vicini, glieli comunico con costo infinito, metrica "infinity", così che non vengano mai scelte poiché irraggiungibili un'altra soluzione è il trigger update: ogni volta che avviene un cambiamento nella topologia della rete allora vengono aggiornate le tabelle in base a questo cambiamento le reti vengono dichiarate irraggiungibili non nell'esatto momento in cui non sono più raggiungibili, ma allo scadere di un time detto "hold-down timer"

1.2.8 Protocolli di routing intra AS (RIP - OSPF)

RIP (Distance Vector)(UDP)

Caratteristiche principali Il RIP è il protocollo di routing più semplice ed è adatto a reti di piccole dimensioni; usa il Distance Vector.

Come metrica del costo utilizza l'hop count, ossia valuta il costo dei percorsi in base al numero dei santi che vengono effettuati. Il valore massimo di hop è 15, perciò quando questo valore arriva a 16 per un certo percorso, questo viene indicato come irraggiungibile (per evitare il counting tot infinity). Il routing update (aggiornamento delle tabelle contenenti le informazioni per il routing) avviene ogni 30s oppure ad ogni cambiamento topologico.

RIPv1 e RIPv2 Nel RIPv1 l'indirizzamento IP era classful, perciò i pacchetti scambiati non contenevano informazioni sulla maschera di rete; quindi come destinazione si scriveva solamente l'ind. IP.

RIPv2 invece ha integrato anche la riga Subnet Mask, così da permettere l'uso degli indirizzi classless.

I messaggi RIP vengono incapsulati all'interno di UDP (porta 520), perciò tramite datagrammi. Ognuno di questi pacchetti può contenere al massimo 25 destinazioni.

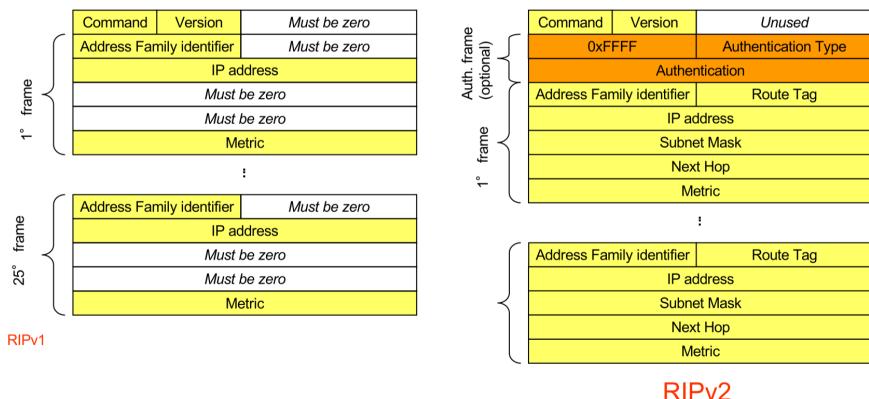


Figure 1.30: Confronto tra RIPv1 e RIPv2

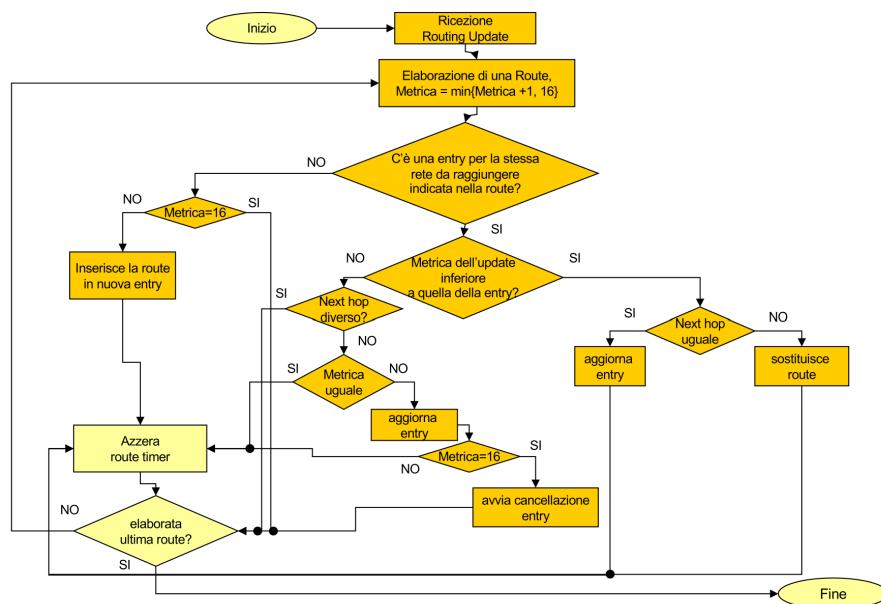


Figure 1.31: Processo di aggiornamento in RIP

OSPF (Link State) e Link State Advertisement (LSA)

Un protocollo che utilizza il link state come alg di routing è il OSPF(open shortest path first); in OSPF si può specificare quale sia il costo(banda, affidabilità, ritardo, anche un valore inserito manualmente).

Quando i router diventano tanti allora OSPF non fa altro che dividere gli AS in aree, ognuna delle quali definita da 4 ottetti, in modo gerarchico;

Area border routers i router al confine con due aree vengono detti “area border router” e devono avere almeno un’interfaccia verso la backbone area(rete 0.0.0.0) questa backbone area(area 0) fa da ponte l’idea è quella di usare l’algoritmo dijkstra in modo modulare in ogni singola area.

LSA I Link State Packet: sono generati dal protocollo OSPF, sono pacchetti di aggiornamento che contengono i vari nodi a cui il router è collegato ogni volta che collega una rete viene detta LSA(LINK STATE ADVERTISEMENT) nella rete locale LAN è importante definire un router che inoltri gli lsas, questo router viene chiamato “designated” per ridurre l’overhead da segnalazione, l’lsa viene inviato ogni 30 minuti o ad ogni qual volta ci sia un cambiamento dei collegamenti della rete differenza tra lsa e pacchetto OSPF LSA = informazione vera e propria sullo stato dei link. Pacchetto OSPF (come il Link State Update) = trasporta una o più LSA tra router OSPF. AUTENTICAZIONE DEL PEER E AUTENTICAZIONE DEL MESSAGGIO - DIFFERENZA aut peer è riferita a chi manda il pacchetto: esempio, io mando un pacchetto a michele, verifico che questo pacchetto sia effettivamente per michele(aut peer) aut messaggio è una verifica sul “messaggio” del pacchetto che viene inviato, per assicurarsi che sia proprio quello inviato e non differente il pacchetto OSPF è incapsulato in pacchetti IP(type 59(cinque nove, non è cinquantanove) in esadecimale) del pacchetto OSPF mi serve sapere: version, Router ID(IP address del router), Area ID, link state checksum.

tre tipologie di LSP:

- link state request: richiesta di invio di uno o più LSA
- link state update: ogni 30 min o quando avviene un cambiamento topologico
- link state acknowledgment: all’avvenuta ricezione del LSA

cinque tipologie di LSA:

- | | |
|---------------|-------------------|
| • router LSA | • AS Boundary |
| • Network LSA | • router LSA |
| • Summary LSA | • AS external LSA |

! un LSA “vecchio” di circa un’ora viene rimosso

Utilizzo di Dijkstra in diverse aree Ad esempio, nell’area 1 uso il dijkstra tra quei 4 router, i router di bordo area avranno sicuramente dei collegamenti con le altre aree, però durante il calcolo dijkstra in area 1 questi altri collegamenti “terminano” sui router di bordo area. questo ragionamento viene applicato ad ogni area, anche alla backbone area, che mi va a risolvere il fatto che i router di bordo area precedentemente non integravano nel dijkstra i collegamenti fuori dalla propria area. così ho calcolato le strade più efficienti di ogni area, il che porta a vantaggi anche nel quadro globale della cosa. ci sono anche router detti boundary router che si collegano con “internet”, con il “mondo esterno”, infatti tutte le aree precedenti fanno parte della stessa rete, è divisa solo in aree. così abbasso il costo computazionale dell’algoritmo dijkstra perchè non lo applico a tutta la rete, ma a tutte le aree che compongono la rete, poichè è un algoritmo la cui complessità è $n \log(n)$, perciò meglio usarlo più volte con un numero inferiore di nodi(n) che una sola volta con un numero elevato di n

Ogni router conserva gli LSA più recenti

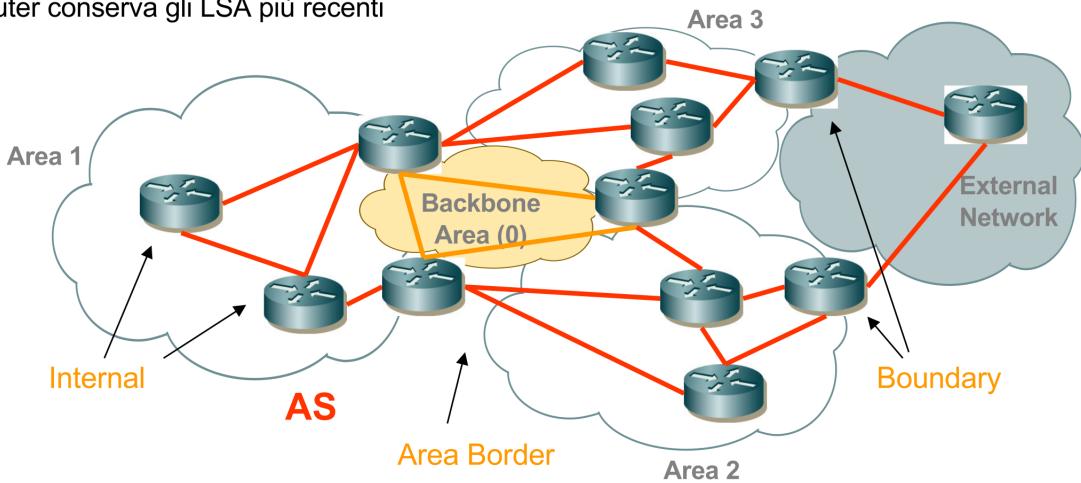


Figure 1.32: Backbone della rete OSPF

Pacchetti OSPF

version	packet type	Packet Length
Router ID (IP address)		
Area ID		
Checksum	Auth. Type	
Authentication Data		
Link state age	options	Link state type
Link state ID		
Advertising router		
Link State sequence number		
Link state checksum	Length	
Link state Data		

Figure 1.33: Pacchetto OSPF

1.2.9 Protocolli di routing inter-AS - (BGP)

BGP (Path Vector)