

Livello di Rete:
IP forwarding, ICMP, routing, protocollo OSPF

Prof.ssa Gaia Maselli
maselli@di.uniroma1.it

Parte di queste slide sono state prese dal materiale associato ai libri:

- 1) B.A. Forouzan, F. Mosharraf – Reti di calcolatori. Un approccio top-down. Copyright © 2013 McGraw-Hill Education Italy srl. Edizione italiana delle slide a cura di Gabriele D'Angelo e Gaia Maselli
- 2) Computer Networking: A Top Down Approach , 6th edition. All material copyright 1996-2009 J.F Kurose and K.W. Ross, All Rights Reserved

Livello di rete

- ❑ Forwarding e routing
- ❑ Struttura dei router
- ❑ IPv4
 - Formato dei datagrammi IPv4
 - Frammentazione
 - Indirizzamento IPv4 (con classi e senza classi)
 - DHCP
 - NAT
- ❑ Forwarding dei datagrammi IP
- ❑ ICMP
- ❑ Routing

Forwarding datagrammi IP

- ❑ **Inoltrare significa collocare il datagramma sul giusto percorso (porta di uscita del router) che lo porterà a destinazione (o lo farà avanzare verso la destinazione)**
- ❑ **Inviare il datagramma al prossimo hop**
- ❑ Quando un host ha un datagramma da inviare lo invia al router della rete locale
- ❑ Quando un router riceve un datagramma da inoltrare, accede alla tabella di routing per trovare il successivo hop a cui inviarlo
- ❑ L'inoltro richiede una riga nella tabella per ogni blocco di rete

esempio

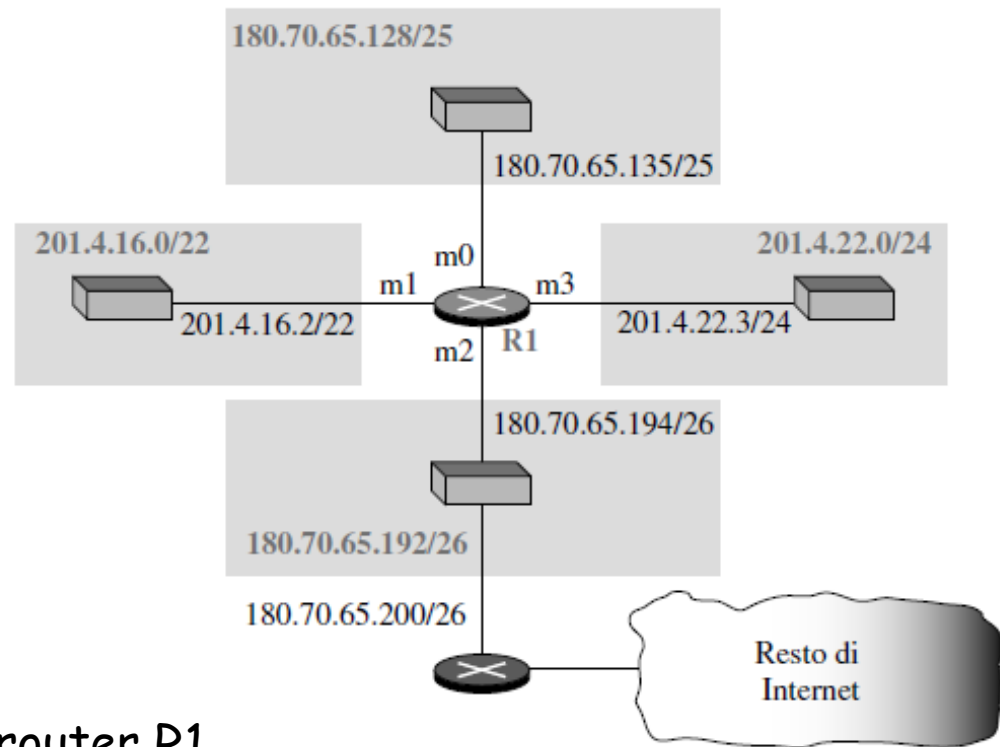


Tabella di inoltro per il router R1

Indirizzo di rete	Hop successivo	interfaccia
180.70.65.192/26	-	m2
180.70.65.128/25	-	m0
201.4.22.0/24	-	m3
201.4.16.0/22	-	m1
default	180.70.65.200	m2

Altra rappresentazione della tabella d'inoltro

<i>Bit a sinistra nell'indirizzo di destinazione</i>	<i>Salto successivo</i>	<i>Interfaccia</i>
10110100 01000110 01000001 11	–	m2
10110100 01000110 01000001 1	–	m0
11001001 00000100 00011100	–	m3
11001001 00000100 000100	–	m1
Default	180.70.65.200	m2

- All'interno della tabella di routing sono presenti indirizzi di rete (lunghezza inferiore a 32 bit)
- Ma un datagramma contiene l'indirizzo IP dell'host di destinazione (pari a 32 bit) e non indica la lunghezza del prefisso di rete
- Come si esegue l'instradamento?

Altra rappresentazione della tabella d'inoltro

<i>Bit a sinistra nell'indirizzo di destinazione</i>	<i>Salto successivo</i>	<i>Interfaccia</i>
10110100 01000110 01000001 11	–	m2
10110100 01000110 01000001 1	–	m0
11001001 00000100 00011100	–	m3
11001001 00000100 000100	–	m1
Default	180.70.65.200	m2

Quando arriva un datagramma in cui i 26 bit a sinistra nell'indirizzo di destinazione combaciano con i bit della prima riga, il pacchetto viene inviato attraverso l'interfaccia m2. Analogamente negli altri casi.

La tabella mostra chiaramente che la prima riga ha un prefisso più lungo (che matcha con il successivo) che indica uno spazio di indirizzi più piccolo

esempio

Mostrare il processo d'inoltro di un datagramma, con indirizzo di destinazione 180.70.65.140 (10110100 01000110 01000001 10001100), nel caso arrivi a R1.

Soluzione

Il router esegue i seguenti passaggi:

1. La prima maschera (/26) ovvero 10110100 01000110 01000001 11 è applicata all'indirizzo di destinazione.

10110100 01000110 01000001 10001100 (indirizzo destinazione)

10110100 01000110 01000001 11000000 (maschera)

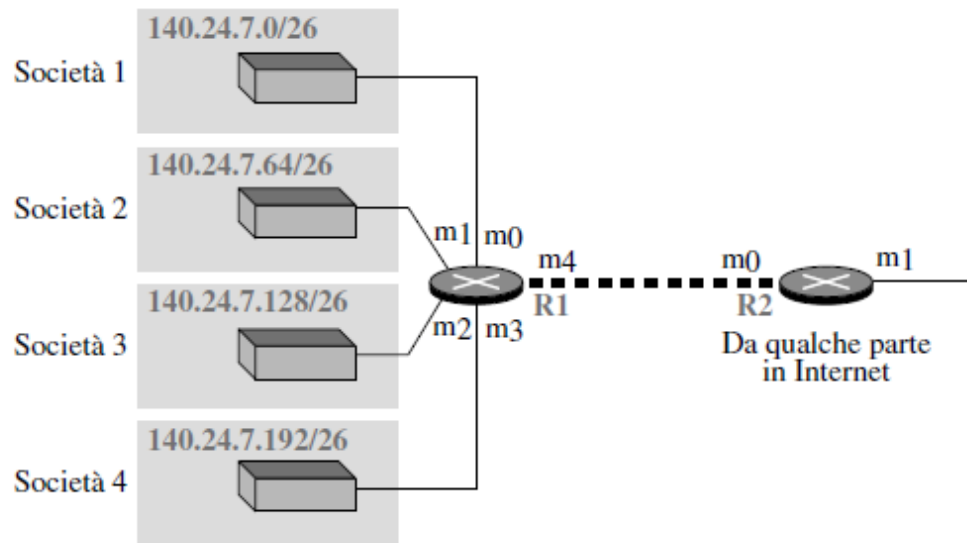
10110100 01000110 01000001 10000000 (indirizzo di rete della destinazione)

Il risultato è 180.70.65.128, che non combacia con l'indirizzo di rete corrispondente.

2. La seconda maschera (/25) è applicata all'indirizzo di destinazione. Il risultato è 180.70.65.128 che combacia con l'indirizzo di rete corrispondente. L'indirizzo del salto successivo e il numero di interfaccia m0 vengono estratti dalla tabella e usati per inoltrare il datagramma.

Aggregazione degli indirizzi

- ❑ Inserire nella tabella una riga per ogni blocco può portare a tabelle molto lunghe, con aumento del tempo necessario per fare la ricerca
- ❑ Soluzione: aggregazione degli indirizzi



Aggregazione di indirizzi
nella tabella di R2

Tabella d'inoltro per R1

Indirizzo di rete/maschera	Indirizzo del salto successivo	Interfaccia
140.24.7.0/26	-----	m0
140.24.7.64/26	-----	m1
140.24.7.128/26	-----	m2
140.24.7.192/26	-----	m3
0.0.0.0/0	Indirizzo di R2	m4

Tabella d'inoltro per R2

Indirizzo di rete/maschera	Indirizzo del salto successivo	Interfaccia
140.24.7.0/24	-----	m0
0.0.0.0/0	Router di default	m1

Corrispondenza con la maschera più lunga

- Cosa accade se l'organizzazione 4 è connessa al router R2?

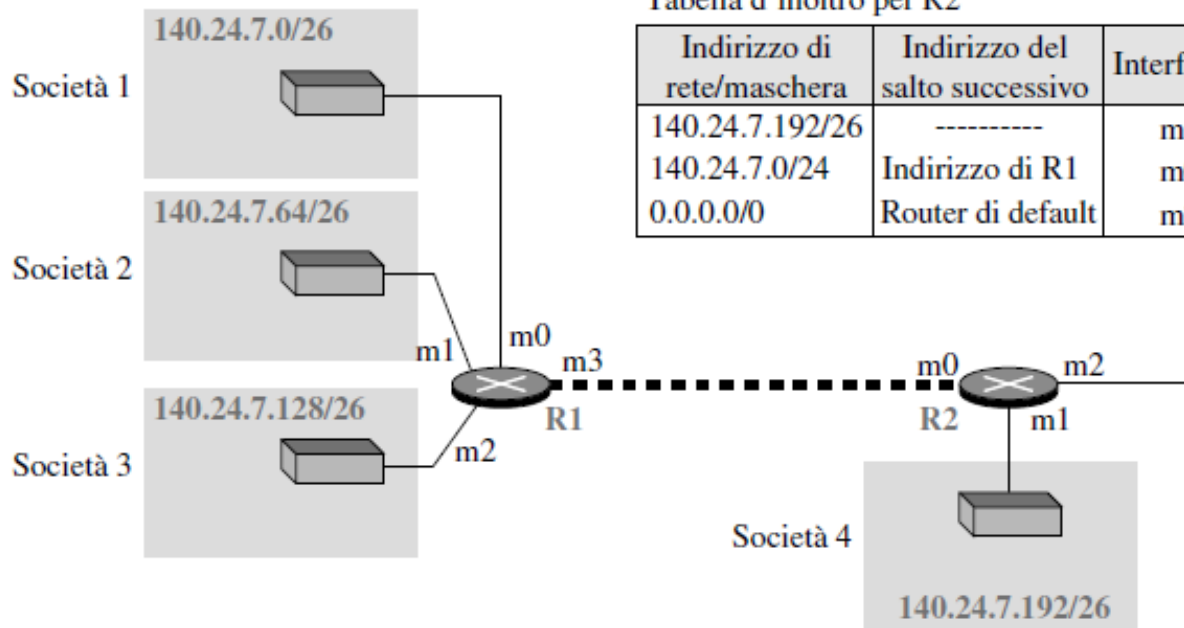


Tabella d'inoltro per R2

Indirizzo di rete/maschera	Indirizzo del salto successivo	Interfaccia
140.24.7.192/26	-----	m1
140.24.7.0/24	Indirizzo di R1	m0
0.0.0.0/0	Router di default	m2

Tabella d'inoltro per R1

Indirizzo di rete/maschera	Indirizzo del salto successivo	Interfaccia
140.24.7.0/26	-----	m0
140.24.7.64/26	-----	m1
140.24.7.128/26	-----	m2
0.0.0.0/0	Router di default	m3

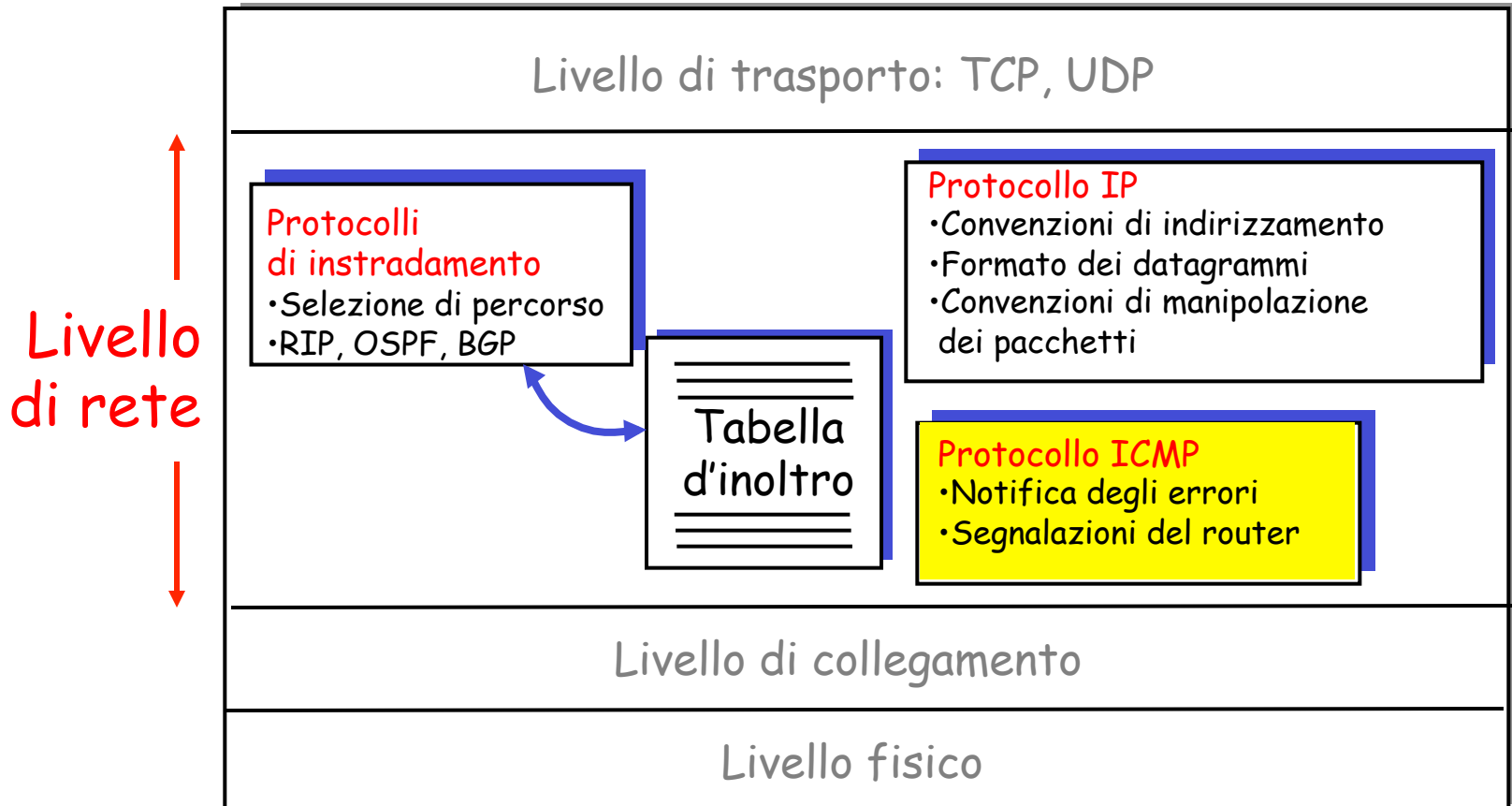
Corrispondenza con la maschera più lunga!!!

Livello di rete

- ❑ Forwarding e routing
- ❑ Struttura dei router
- ❑ IPv4
 - Formato dei datagrammi IPv4
 - Frammentazione
 - Indirizzamento IPv4 (con classi e senza classi)
 - DHCP
 - NAT
- ❑ Forwarding dei datagrammi IP
- ❑ **ICMP**
- ❑ Routing

Livello di rete

Uno sguardo al livello di rete Internet:



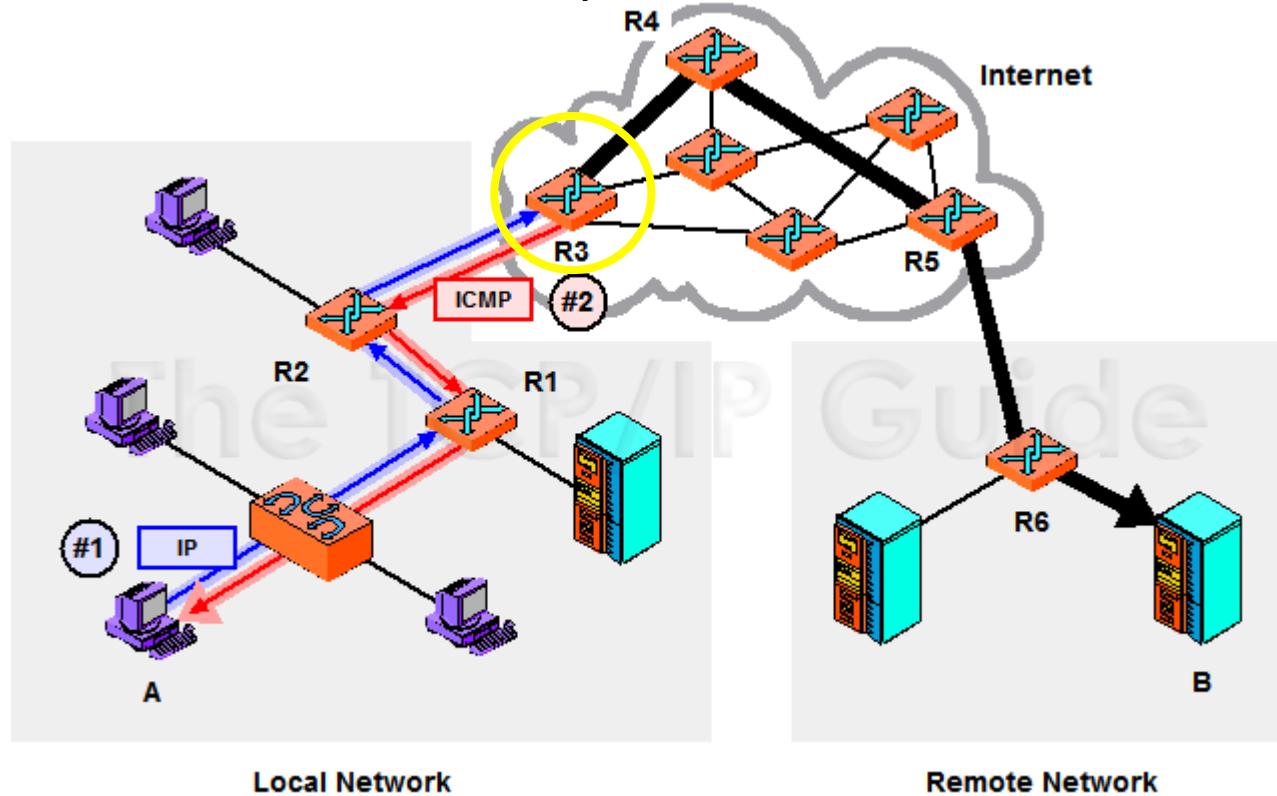
Il campo dati dei datagrammi IP può contenere un messaggio ICMP

Gestione errori?

- ❑ Cosa accade se un router deve scartare un datagramma perché non riesce a trovare un percorso per la destinazione finale?
- ❑ Cosa accade se un datagramma ha il campo TTL pari a 0?
- ❑ E se un host di destinazione non ha ricevuto tutti i frammenti di un datagramma entro un determinato limite di tempo?
- ❑ Situazioni di errore che IP non gestisce!

Internet Control Message Protocol (ICMP)

- Viene usato da host e router per scambiarsi informazioni a livello di rete.



A typical use of ICMP is to provide a feedback mechanism when an IP message is sent. In this example, device *A* is trying to send an IP datagram to device *B*. However, when it gets to router *R3* a problem of some sort is detected that causes the datagram to be dropped. *R3* sends an ICMP message back to *A* to tell it that something happened, hopefully with enough information to let *A* correct the problem, if possible. *R3* can only send the ICMP message back to *A*, not to *R2* or *R1*.

Internet Control Message Protocol (ICMP)

- Viene usato da host e router per scambiarsi informazioni a livello di rete.
 - report degli errori: host, rete, porta, protocollo irraggiungibili.
 - **echo request/reply** (usando il programma ping).
- Livello di rete "sopra" IP:
 - ICMP è considerato parte di IP anche se **usa IP per inviare i suoi messaggi**
- Messaggi ICMP: hanno un campo tipo e un campo codice, e contengono l'intestazione e i primi 8 byte del datagramma IP che ha provocato la generazione del messaggio.

<u>Tipo</u>	<u>Codice</u>	<u>Descrizione</u>
0	0	Risposta eco (a ping)
3	0	rete destin. irraggiungibile
3	1	host destin. irraggiungibile
3	2	protocollo dest. irraggiungibile
3	3	porta destin. irraggiungibile
3	6	rete destin. sconosciuta
3	7	host destin. sconosciuto
4	0	riduzione (controllo di congestione)
8	0	richiesta eco
9	0	annuncio del router
10	0	scoperta del router
11	0	TTL scaduto
12	0	errata intestazione IP

Ping

- Il programma *ping* si basa sui messaggi di richiesta e risposta *echo* di ICMP

```
$ ping pads.cs.unibo.it
PING chernobog.pads.cs.unibo.it (130.136.132.11) 56(84) bytes of data.
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=1 ttl=52 time=34.2 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=2 ttl=52 time=33.1 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=3 ttl=52 time=34.0 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=4 ttl=52 time=33.9 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=5 ttl=52 time=33.3 ms
--- chernobog.pads.cs.unibo.it ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 33.177/33.758/34.220/0.417 ms
```

Traceroute e ICMP

- ❑ Il programma invia una serie di datagrammi IP alla destinazione ciascuno contenente un segmento UDP con un numero di porta inutilizzata.
 - Il primo pari a $TTL = 1$
 - Il secondo pari a $TTL = 2$, ecc.
 - Numero di porta improbabile
 - L'origine avvia un timer per ogni datagramma
- ❑ Quando l' n -esimo datagramma arriva all' n -esimo router:
 - Il router scarta il datagramma.
 - Invia all'origine un messaggio di allerta ICMP (tipo 11, codice 0).
 - Il messaggio include il nome del router e l'indirizzo IP.

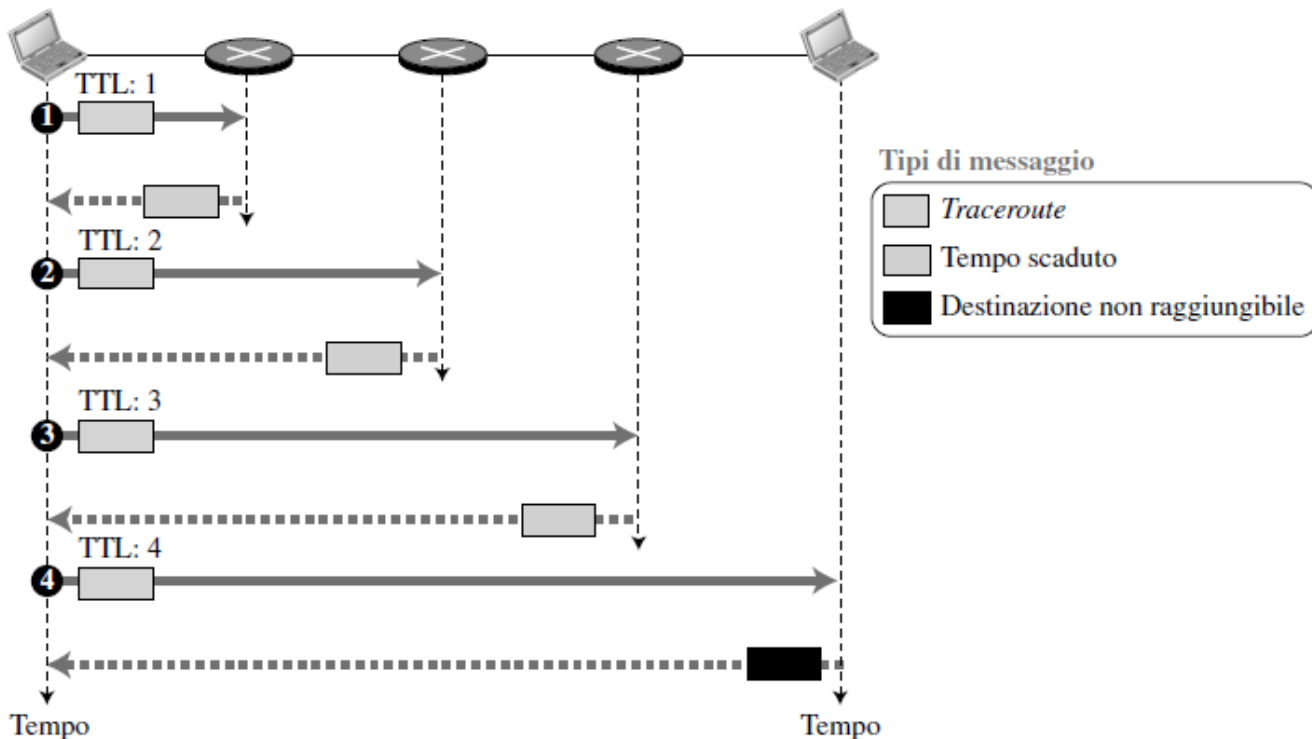
- ❑ Quando il messaggio ICMP arriva, l'origine può calcolare RTT
- ❑ Traceroute lo fa per 3 volte

Criteri di arresto dell'invio

- ❑ Quando un segmento UDP arriva all'host di destinazione.
- ❑ L'host di destinazione restituisce un messaggio ICMP di porta non raggiungibile (tipo 3, codice 3).
- ❑ Quando l'origine riceve questo messaggio ICMP, si blocca.

traceroute

- ❑ Non c'è un programma server (ma solo un client)
- ❑ Le risposte arrivano da ICMP (tempo scaduto dai router intermedi e porta non raggiungibile dall'host di destinazione)



Livello di rete

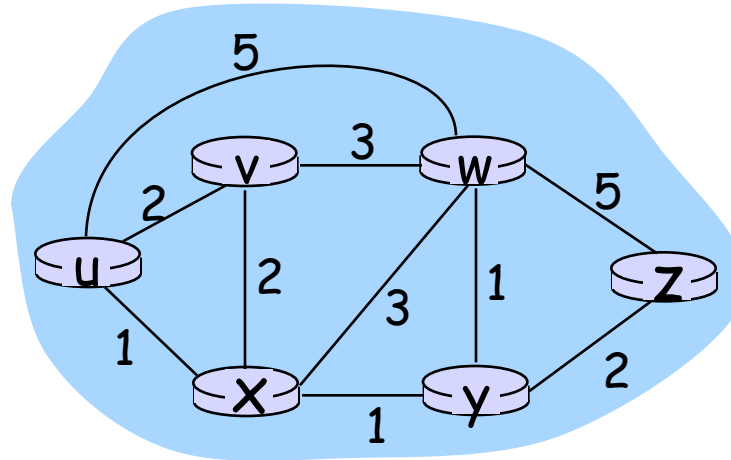
- ❑ Forwarding e routing
- ❑ Struttura dei router
- ❑ IPv4
 - Formato dei datagrammi IPv4
 - Frammentazione
 - Indirizzamento IPv4 (con classi e senza classi)
 - DHCP
 - NAT
- ❑ Forwarding dei datagrammi IP
- ❑ ICMP
- ❑ **Routing**
 - Link state
 - Distance vector

Routing

- ❑ Quale percorso deve seguire un pacchetto che viene instradato da un router sorgente a un router destinazione?
- ❑ Sono disponibili più percorsi... quale si sceglie?
- ❑ Il routing si occupa di trovare il miglior percorso (o rotta, path, route) e inserirlo nella tabella di routing (o forwarding)

Il routing costruisce le tabelle, il forwarding le usa!

Grafo di una rete di calcolatori



Grafo: $G = (N,E)$

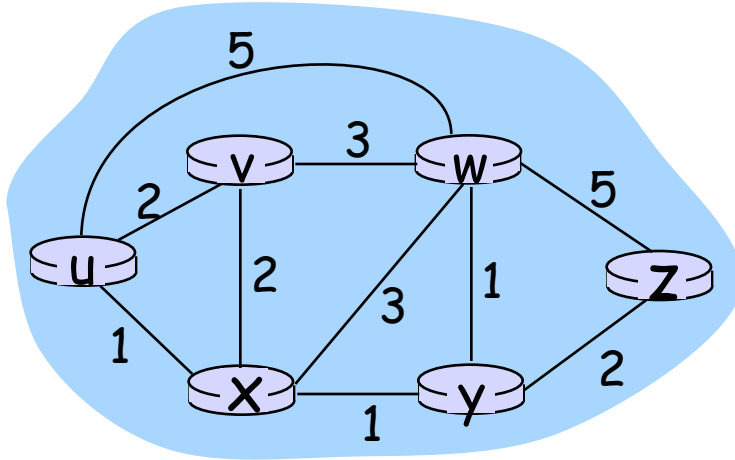
N = insieme di nodi (router) = $\{ u, v, w, x, y, z \}$

E = insieme di archi (collegamenti) = $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

N.B.: Il grafo è un'astrazione utile anche in altri contesti di rete

Esempio: P2P, dove N è un insieme di peer ed E è un insieme di collegamenti TCP

Grafo di una rete : costi



- $c(x,x')$ = costo del collegamento (x,x')
 - es., $c(w,z) = 5$
- il costo di un cammino è semplicemente la somma di tutti i costi degli archi lungo il cammino

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

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

Algoritmo d'instradamento: determina il cammino a costo minimo.

Classificazione degli algoritmi d'instradamento

Globale o decentralizzato?

Globale:

- ❑ L'algoritmo riceve in ingresso tutti i collegamenti tra i nodi e i loro costi.
- ❑ **Algoritmi a stato del collegamento (*link-state algorithm*).**

Decentralizzato:

- ❑ Ogni nodo elabora un vettore di stima dei costi (distanze) verso tutti gli altri nodi nella rete.
- ❑ Il cammino a costo minimo viene calcolato in modo distribuito e iterativo.
- ❑ **Algoritmo a vettore distanza (*VC, distance-vector algorithms*)**

Statico o dinamico?

Statico:

- ❑ I cammini cambiano molto raramente.

Dinamico:

- ❑ Determinano gli instradamenti al variare di:
 - Volume di traffico
 - Topologia della rete

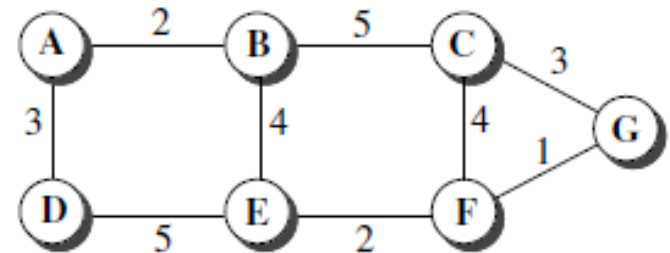
Livello di rete

- ❑ Forwarding e routing
- ❑ Struttura dei router
- ❑ IPv4
 - Formato dei datagrammi IPv4
 - Frammentazione
 - Indirizzamento IPv4 (con classi e senza classi)
 - DHCP
 - NAT
- ❑ Forwarding dei datagrammi IP
- ❑ ICMP
- ❑ **Routing**
 - **Link state**
 - Distance Vector

Algoritmo d'instradamento a stato del collegamento (LS)

Algoritmo di Dijkstra:

- La topologia di rete e tutti i costi dei collegamenti sono noti a tutti i nodi
 - attraverso il "link-state broadcast".
 - tutti i nodi dispongono delle stesse informazioni
- Calcola il cammino a costo minimo da un nodo (origine) a tutti gli altri nodi della rete.
 - Crea una **tabella d'inoltro** per quel nodo
- È iterativo: dopo la k -esima iterazione i cammini a costo minimo sono noti a k nodi di destinazione.



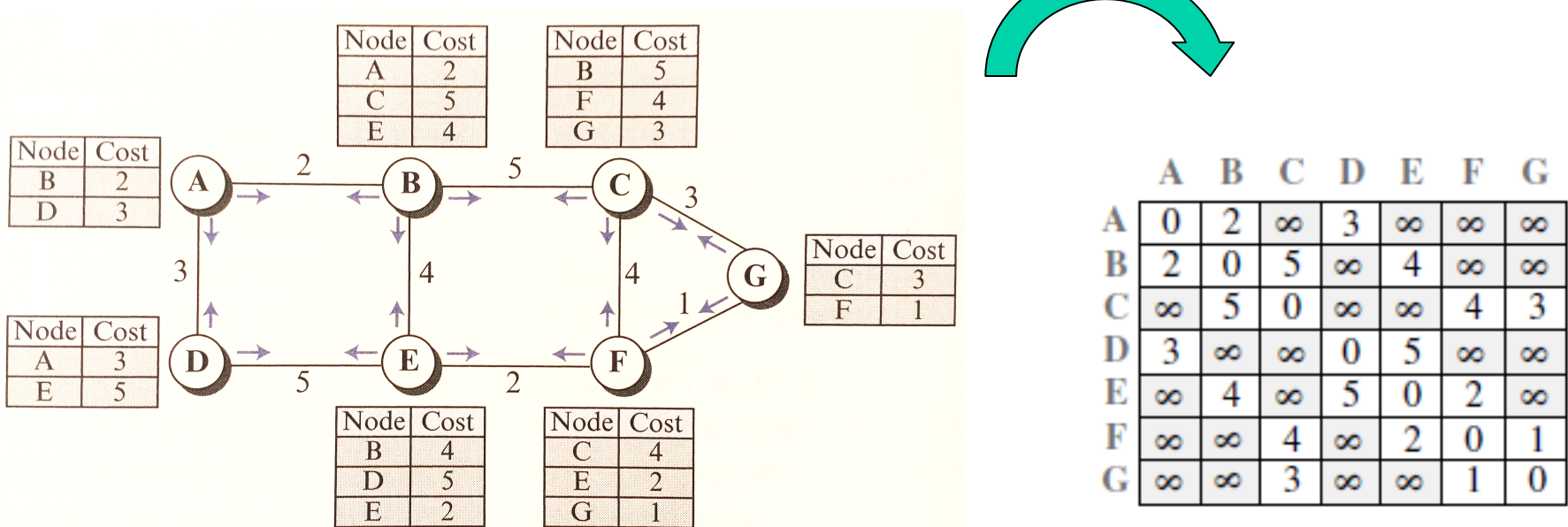
a. Il grafo pesato

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Il link-state database

Link-state database (LSDB)

- ❑ LSDB è unico per tutta la rete
- ❑ Si costruisce mediante flooding
- ❑ Ogni nodo invia messaggi di benvenuto ai propri vicini per scoprire identità del vicino e costo del nodo (creando il LS packet - LSP)
- ❑ Flooding del LSP (invio su ogni interfaccia ad esclusione di quella da cui si è ricevuto il pacchetto)



Costruire la tabella di routing

- ❑ Per costruire il suo albero a costo minimo utilizzando il LSDB ogni nodo deve eseguire l'algoritmo di Dijkstra
- ❑ Ogni nodo sceglie se stesso come radice dell'albero
- ❑ Ogni nodo applica l'algoritmo indipendentemente

Algoritmo di routing a stato del collegamento (LS)

Definiamo la seguente notazione:

- N : insieme dei nodi della rete
- $c(x,y)$: costo dei collegamenti dal nodo x al nodo y
- $D(v)$: costo del cammino minimo dal nodo origine alla destinazione v per quanto riguarda l'iterazione corrente.
- $p(v)$: immediato predecessore di v lungo il cammino.
- N' : sottoinsieme di nodi per cui il cammino a costo minimo dall'origine è definitivamente noto.

Algoritmo di Dijkstra

- Calcola i percorsi di costo minimo da un nodo u a tutti gli altri nodi della rete
- È iterativo e viene eseguito un numero di volte pari al numero di nodi nella rete

Algoritmo di Dijkstra

1 **Inizializzazione:**

2 $N' = \{r\}$ /* r è il nodo che esegue l'algoritmo (radice dell'albero)*/

3 per tutti i nodi n

4 se n è adiacente a r

5 allora $D(n) = c(r,n)$

6 altrimenti $D(n) = \infty$

7

8 **Ciclo**

9 determina un n non in N' tale che $D(n)$ sia minimo

10 aggiungi n a N'

11 per ogni nodo a adiacente a n e non in N' aggiorna $D(a)$:

12 $D(a) = \min(D(a), D(n) + c(n,a))$

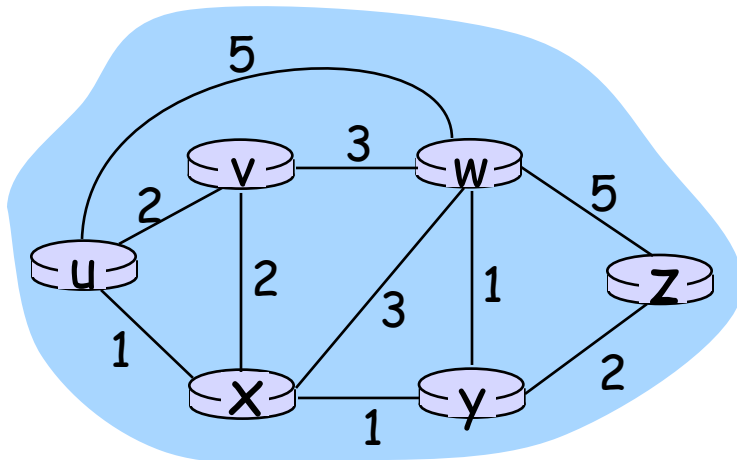
13 /* il nuovo costo verso a è il vecchio costo verso a oppure

14 il costo del cammino minimo noto verso a più il costo da n a a */

15 **Finché $N' = N$**

Algoritmo di Dijkstra: esempio

passo	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux					
2	uxy					
3	uxyv					
4	uxyvw					
5	uxyvwz					



1 **Inizializzazione:**

- 2 $N' = \{r\}$ /* r è il nodo che esegue l'algoritmo (radice dell'albero)*/
- 3 per tutti i nodi n
- 4 se n è adiacente a r
- 5 allora $D(n) = c(r,n)$
- 6 altrimenti $D(n) = \infty$

8 **Ciclo**

- 9 determina un n non in N' tale che $D(n)$ sia minimo
- 10 aggiungi n a N'
- 11 per ogni nodo a adiacente a n e non in N' aggiorna $D(a)$:
- 12 $D(a) = \min(D(a), D(n) + c(n,a))$
- 13 /* il nuovo costo verso a è il vecchio costo verso a oppure
- 14 il costo del cammino minimo noto verso a più il costo da n a a */
- 15 **Finché** $N' = N$

Algoritmo di Dijkstra: un altro esempio

Cammino a costo minimo da u:

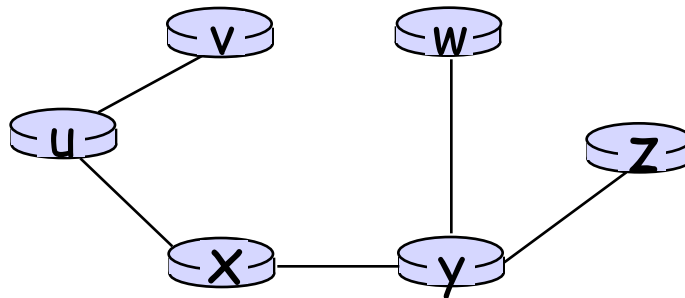
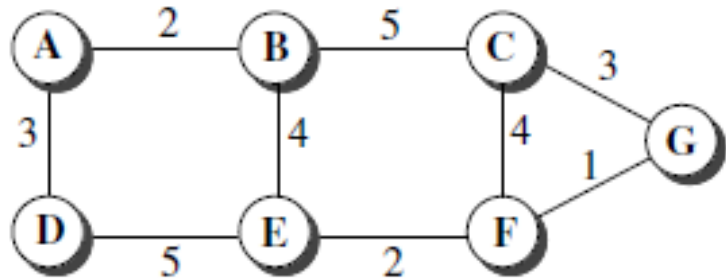


Tabella d'inoltro in u:

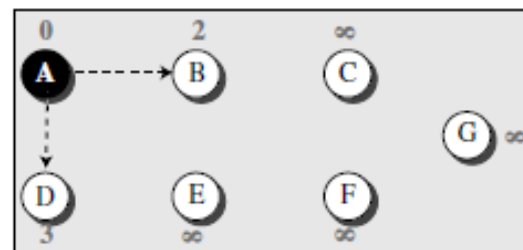
destinazione	collegamento
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Altro esempio

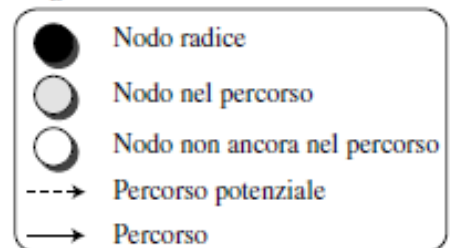


a. Il grafo pesato

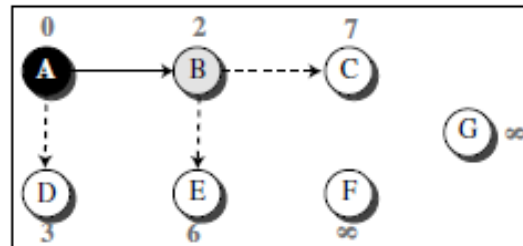
Inizializzazione



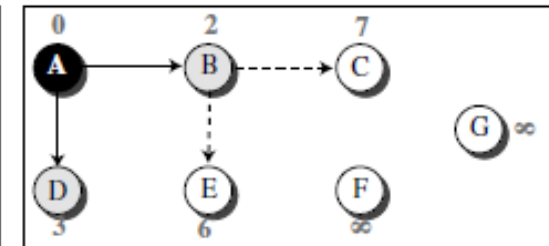
Legenda



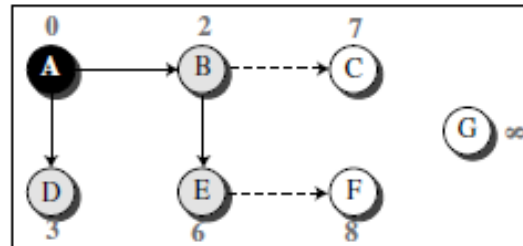
Iterazione 1



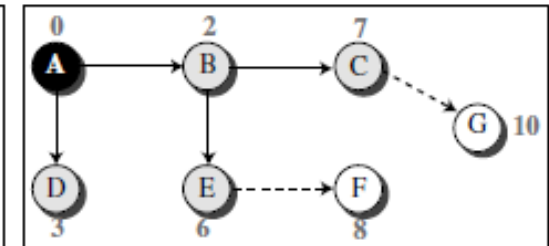
Iterazione 2



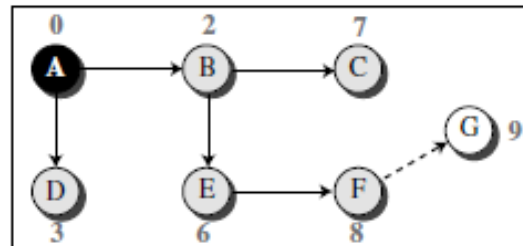
Iterazione 3



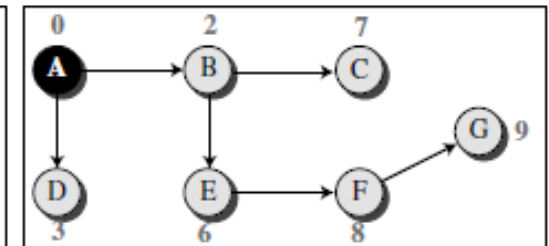
Iterazione 4



Iterazione 5



Iterazione 6



Protocollo di routing

- ❑ Un protocollo è qualcosa di più di un algoritmo!
- ❑ Un protocollo deve definire il suo ambito di funzionamento, i messaggi che vengono scambiati, la comunicazione tra router e l'interazione con gli altri protocolli
- ❑ **OSPF: Open Shortest Path First** basato su algoritmo LS

OSPF (Open Shortest Path First)

- ❑ “open”: le specifiche del protocollo sono pubblicamente disponibili.
- ❑ È un protocollo a stato del collegamento:
 1. Utilizza il **flooding** di informazioni di stato del collegamento
 2. Utilizza l'algoritmo di Dijkstra per la determinazione del percorso a costo minimo.
- ❑ Flooding
 - ❑ Con OSPF, ogni volta che si verifica un cambiamento nello stato di un collegamento, il router manda informazioni d'instradamento a *tutti* gli altri router.
 - ❑ Invia periodicamente (ogni 30 minuti) messaggi OSPF all'intero sistema autonomo, utilizzando il flooding.
- ❑ I messaggi OSPF vengono trasportati direttamente in datagrammi IP usando il numero di protocollo 89 nel campo IP protocol

Messaggi OSPF

- ❑ **Hello:** usato dai router per annunciare la propria esistenza
- ❑ **Database description:** risposta ad hello (consente di ottenere il LSDB a chi si è appena connesso)
- ❑ **Link-state request:** usato per richiedere specifiche informazioni su un collegamento
- ❑ **Link-state update:** messaggio principale usato da OSPF per la costruzione del LSDB
- ❑ **Link-state ack:** riscontro ai link-state update