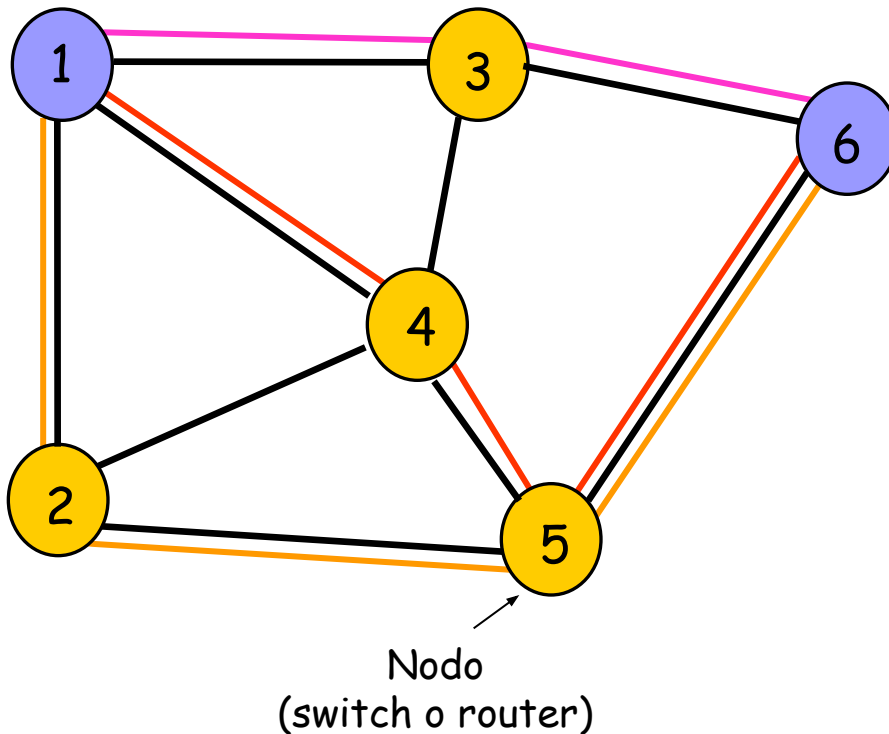


Strato di rete (parte 3)

**Funzione di instradamento:
generalità**

Instradamento in reti a pacchetto



- Tre possibili (loopfree) cammini dal nodo 1 al nodo 6
 - 1-3-6, 1-4-5-6, 1-2-5-6
- Qual'è il cammino migliore ?
 - Minimo ritardo
 - Minimo numero di hop
 - Minimo costo
 - Massima affidabilità

Creazione delle tabelle di routing

- **E' necessario definire la tipologia di informazioni sullo stato dei link**
 - Link up/down; stato di congestione; delay o altre metriche
- **Occorre distribuire le informazioni di stato dei link usando un protocollo di routing**
 - Quali informazioni devono essere scambiate ?
 - Con quale frequenza ?
 - Scambio di informazioni con i vicini, broadcast, flooding
- **Occorre calcolare i cammini migliori**
 - Algoritmo di instradamento
 - Metriche singole o multiple

Requisiti

- **Risposta alle variazioni di stato**
 - Variazioni di topologia o banda dei link
 - Stato di congestione
 - Rapida convergenza
 - Assenza di loop
- **Ottimalità**
 - Utilizzazione ottima delle risorse di rete
 - minimizzazione della lunghezza dei cammini
- **Robustezza**
 - Continuità di servizio in presenza di condizioni anomale (alto carico, congestione di rete, guasti, errori di implementazione)
- **Semplicità**
 - Basso carico di elaborazione

Instradamento Centralizzato o Distribuito

- **Routing Centralizzato**

- I cammini sono determinati da un elemento (nodo) centralizzato
- Le informazioni di stato sono inviate al nodo centrale
- Difficili adattamento ai cambi di topologia
- Soluzione non scalabile e di scarsa affidabilità

- **Routing Distribuito**

- I router determinano i cammini usando un algoritmo distribuito
- Le informazioni di stato sono scambiate tra i router
- Maggiore adattabilità alle variazioni di stato della rete
- Alta scalabilità

Instradamento Statico o Dinamico

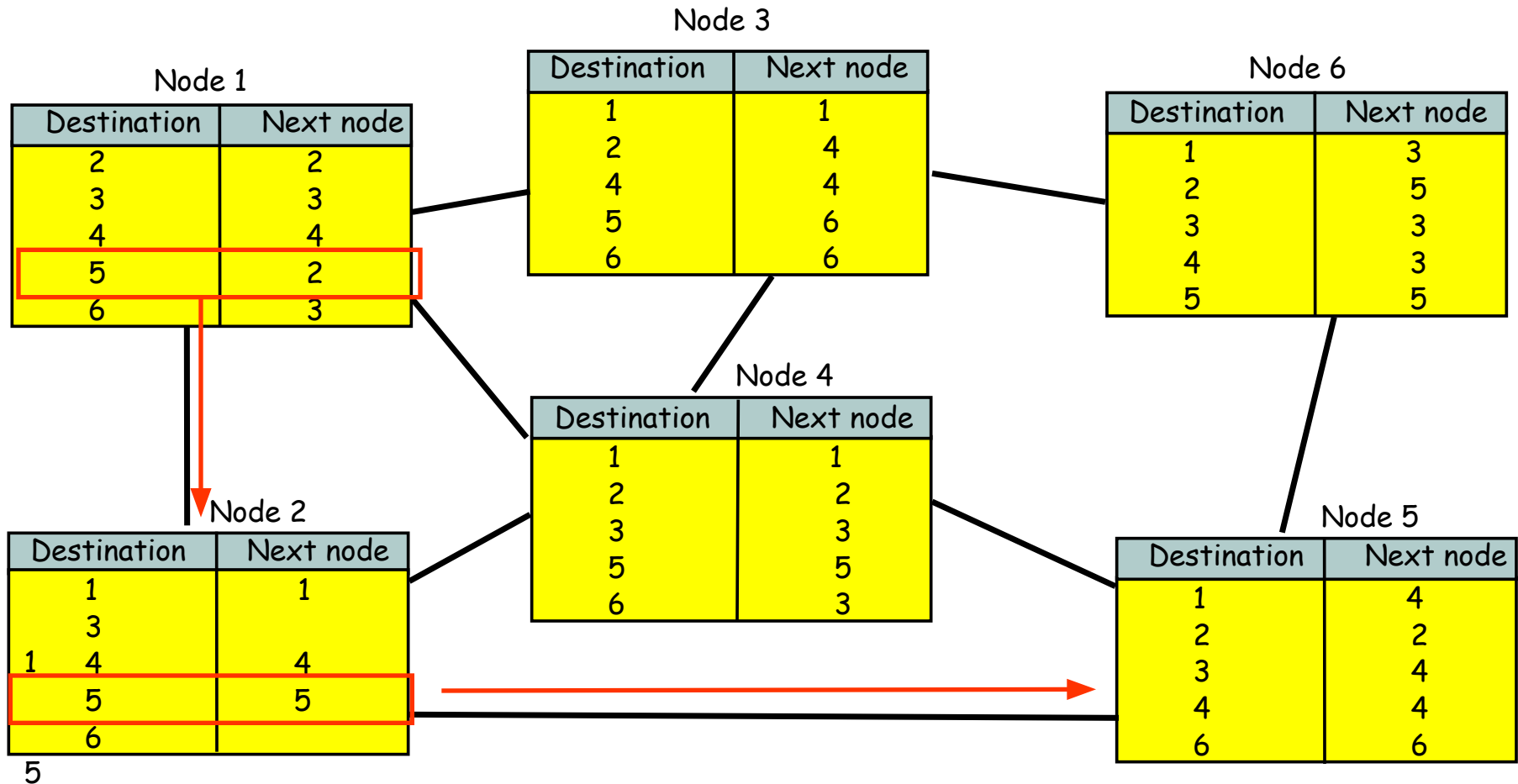
- **Instradamento Statico**

- Cammini configurati manualmente, non variano nel tempo
- Adatto al caso di reti semplici con traffico predicibile
- Usato per imporre alcuni cammini particolari
- Usato per fornire un instradamento di default (default router)

- **Instradamento Dinamico**

- Adatto a sostenere variazioni dello stato della rete
- Calcolo automatico dei cammini
- Cammini determinati in base alle informazioni di stato della rete ricevute per mezzo di un protocollo di instradamento

Routing Table: principio di funzionamento

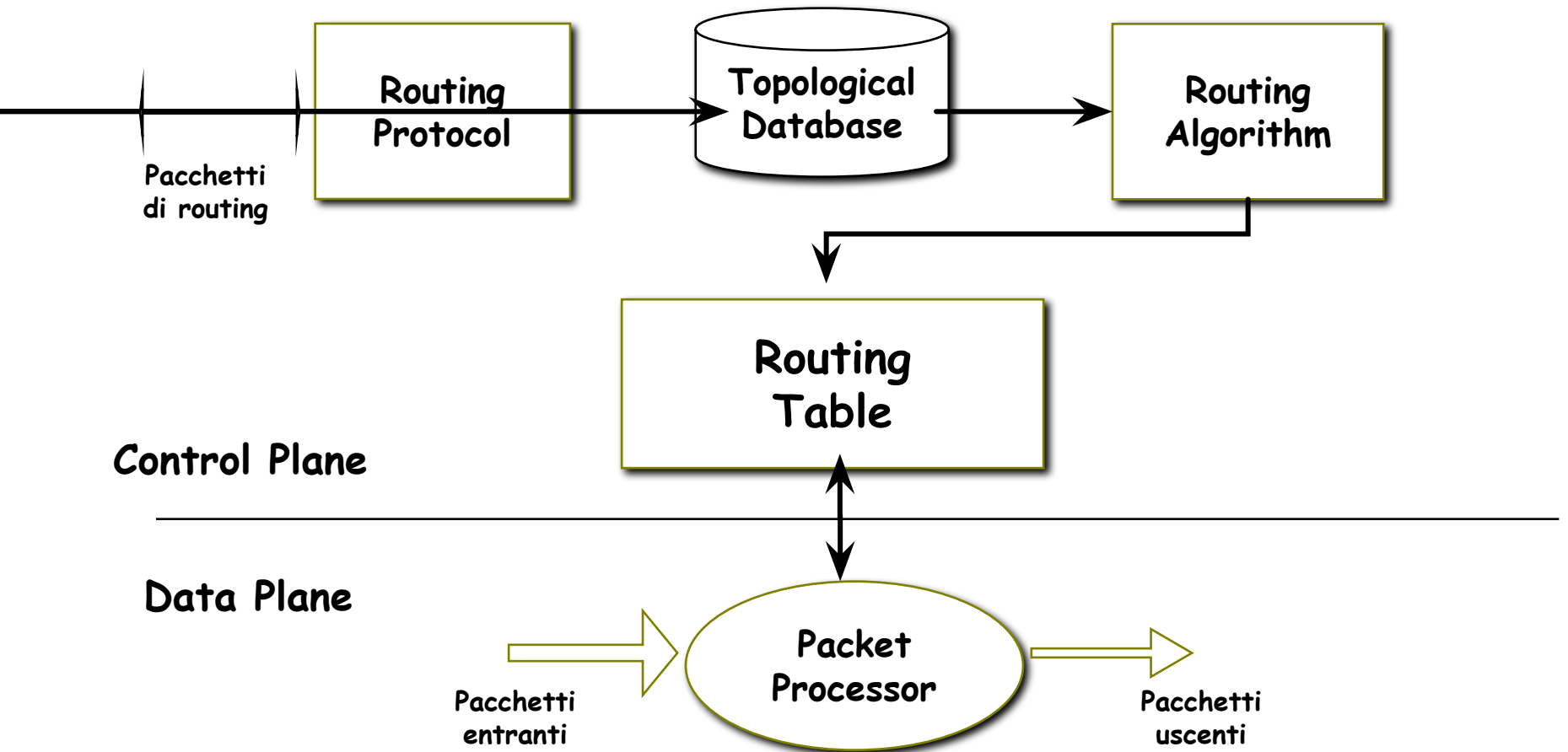


Instradamento in reti IP

Instradamento in reti IP

- Un router possiede un Database Topologico in cui sono memorizzate le informazioni sulla topologia della rete
 - Le informazioni sulla topologia di rete sono aggiornate dai messaggi del protocollo di routing
- L'algoritmo di routing, sulla base delle informazioni contenute nel Database Topologico, determina periodicamente i percorsi a costo minimo tra il router e le possibili reti di destinazione (network prefix)
- La Routing Table è costruita inserendo, per ogni destinazione, sulla base dei risultati del passo precedente, l'informazione relativa al next hop verso cui instradare il pacchetto

Instradamento in reti IP



Instradamento in reti IP

- **Le Routing Table sono dinamiche**
 - ogni router ed ogni host aggiornano nel tempo le informazioni relative alla topologia di rete
- **L'aggiornamento dinamico è necessario perché:**
 - Internet non può essere considerata stabile
 - in caso di guasti alcuni cammini non sono utilizzabili
 - È consigliabile scegliere il cammino in base allo stato di occupazione delle risorse di rete
- **Le RT devono essere aggiornate continuamente (anche ad intervalli di pochi secondi)**
- **L'aggiornamento delle RT è attuato mediante protocolli di colloquio tra i router (*Routing Protocol*)**

Sistemi autonomi

- Un sistema autonomo (*Autonomous System - AS*) è un insieme di host e router controllato da una singola autorità amministrativa (es. ISP)
 - un particolare AS è detto "Core AS" e costituisce il backbone di Internet
 - un router del core AS è detto *Core Router*
 - gli altri AS sono detti "Stub AS"
- Ogni AS ha il proprio protocollo di instradamento
- Uno Stub AS deve aver almeno un router connesso ad un core router; questi router sono detti *Exterior Gateway*
- Un router interno ad un AS è detto *Interior Gateway*

IGP e EGP

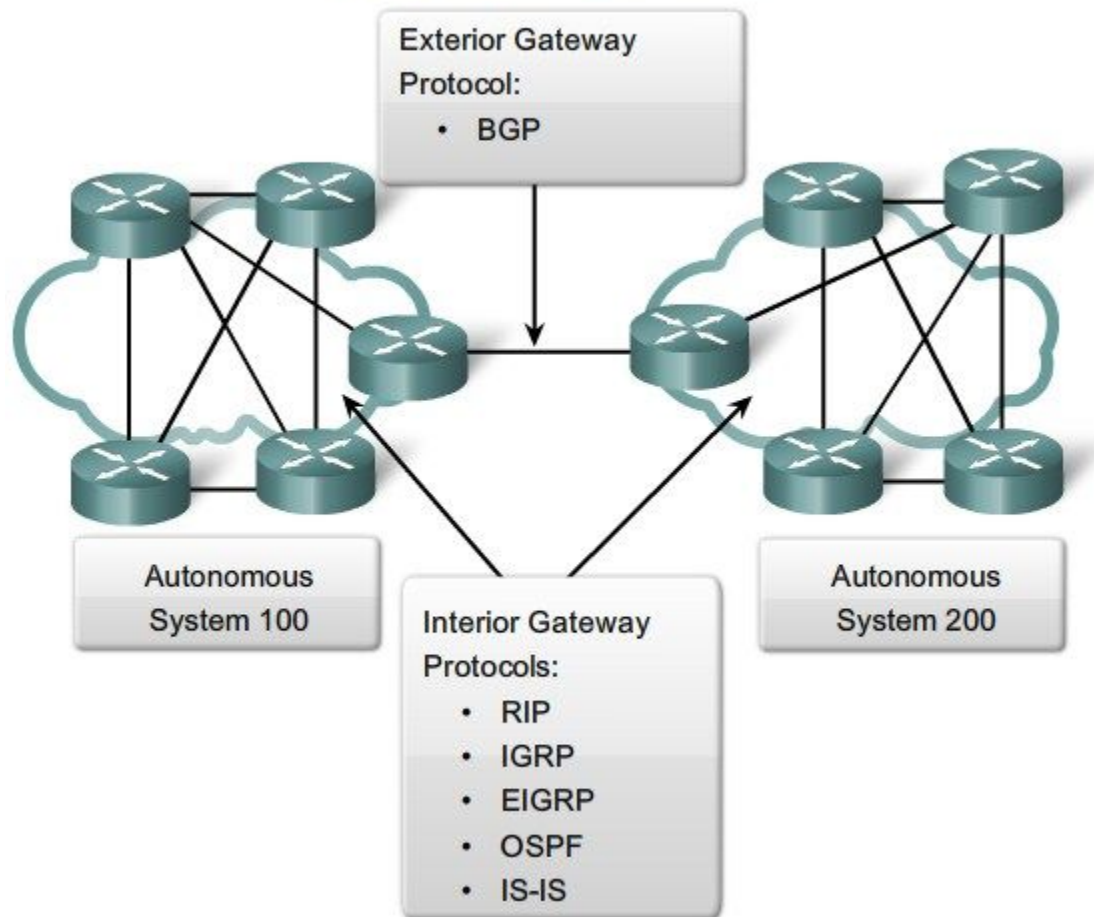
- I protocolli di instradamento all'interno di un AS sono detti *Interior Gateway Protocols* (IGP)
- Le informazioni di instradamento che coinvolgono più di un sistema autonomo sono gestite mediante gli *Exterior Gateway Protocols* (EGP)
- Le informazioni di instradamento degli EGP vengono inviate agli Exterior Gateway di ogni sistema autonomo
- L'instradamento all'interno di un sistema autonomo e la raccolta di dati da inviare ai core router avviene per mezzo degli IGP

IGP e EGP

- **Un EGP svolge tre funzioni**
 - individuazione dei router adiacenti con cui scambiare le informazioni di instradamento
 - verifica continua della funzionalità dei router interlocutori
 - scambio periodico delle informazioni di instradamento, queste riguardano la sola raggiungibilità delle reti, non la distanza

IGP e EGP

IGP vs. EGP Routing Protocols

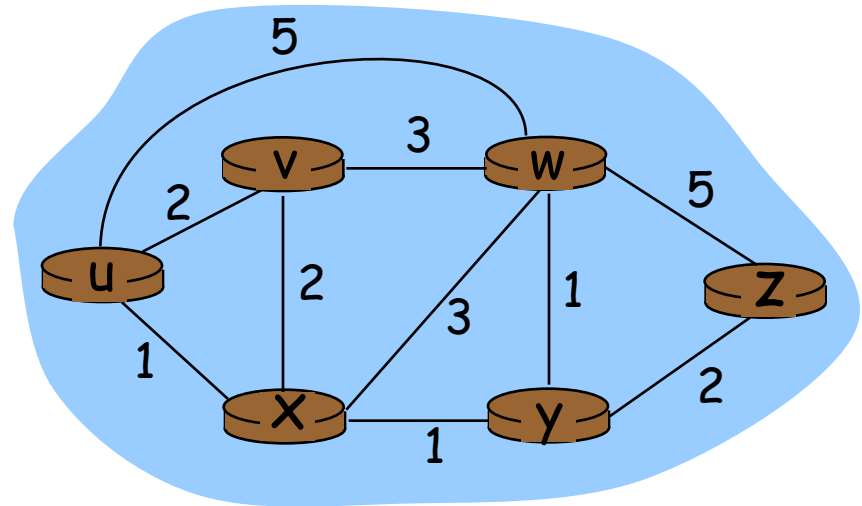


Algoritmi di instradamento

Generalità

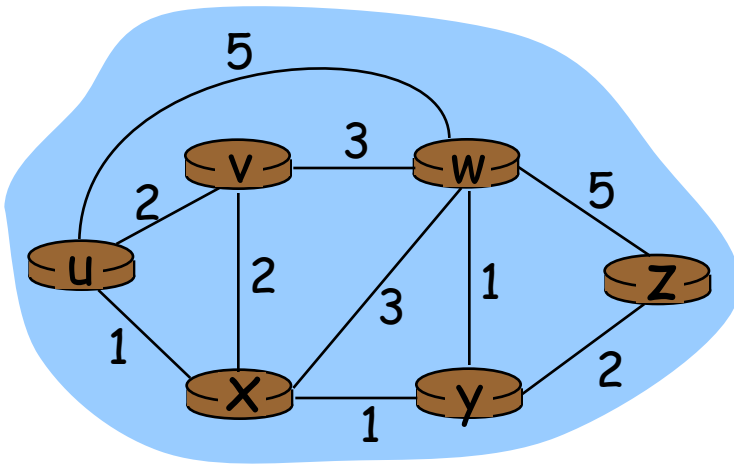
Modello a grafo di una rete

Grafo Pesato
 $G = (N, E, c)$



- **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)}
- **c** = insieme dei costi associati ai rami
 - $c(x,x')$ = costo associato ramo (x,x')

Costo di un cammino



Il costo di un cammino è definito
la somma di tutti
i costi degli archi lungo il cammino

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

- Il **protocollo di instradamento** mette in grado ogni router di determinare il modello a grafo della rete
- L'**algoritmo di instradamento** determina il cammino a costo minimo tra due nodi della rete

Metriche

- **Misurano la “qualità” di un link o di un cammino**
 - **Costo basso**: link ad alta qualità (es. banda elevata), da includere se possibile nei cammini
 - **Costo elevato**: link di bassa qualità (es. banda limitata), da escludere se possibile nei cammini
- **Lunghezza di un cammino (Path Length) = somma dei costi dei link componenti (**Distanza**)**
- **Possibili metriche**
 - **Numero di hop**: misura approssimata delle risorse utilizzate
 - **Affidabilità**: grado di disponibilità del cammino; BER
 - **Ritardo**: somma dei ritardi lungo il path
 - **Bandwidth**: capacità disponibile lungo un path
 - **Carico**: Grado di utilizzazione dei link e dei router lungo il path

Approcci Shortest Path

- Distance Vector Protocol
 - Nodi adiacenti si scambiano la lista delle distanze verso le destinazioni
 - Viene determinato il next-hop migliore per ogni destinazione
 - **Algoritmo di Bellman-Ford**
- Link State Protocol
 - Le informazioni sullo stato dei link (costi) sono diffuse in rete (flooding)
 - I router conoscono l'intera topologia della rete
 - Ogni router calcola lo shortest path ed il next-hop verso ogni destinazione
 - **Algoritmo di Dijkstra**

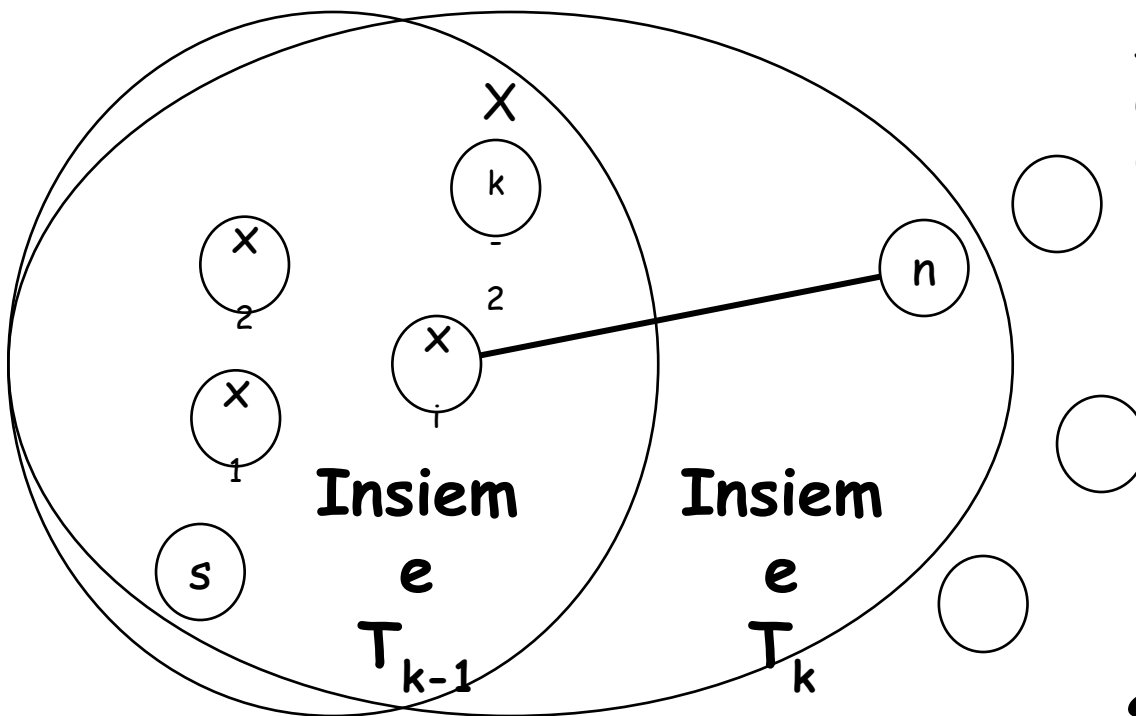
Algoritmo di Dijkstra

Algoritmo di Dijkstra

- Individua il cammino a lunghezza minima tra un nodo s e tutti gli altri nodi di un grafo G procedendo in modo da aumentare progressivamente la distanza
- L'algoritmo procede a passi successivi
 - al passo k -mo sono individuati i k nodi raggiungibili dal nodo sorgente tramite i cammini a costo più basso
 - tali k nodi formano l'insieme T_k
 - al passo $k+1$ -mo si individua il nodo n che è caratterizzato dal cammino dal costo più basso dal nodo s che transita esclusivamente nei nodi dell'insieme T_k
 - viene formato l'insieme T_{k+1} aggiungendo il nodo n all'insieme T_k
 - l'algoritmo termina quando sono stati esplorati tutti i nodi

Algoritmo di Dijkstra

Al passo k viene aggiunto all'insieme T_{k-1} il nodo n caratterizzato dal cammino a costo minimo con il nodo sorgente s che transita esclusivamente in nodi dell'insieme T_{k-1}



Situazione al passo
 k

Algoritmo di Dijkstra

- **Notazioni**

- N : insieme dei nodi del grafo
- s : nodo sorgente
- T_k : insieme dei nodi raggiunti dall'algoritmo al passo k
- $c(i,j)$: peso (costo) del ramo (i,j)
 - $c(i,i) = 0$
 - $c(i,j) \geq 0$ se i vertici i e j sono connessi direttamente
 - $c(i,j) = \infty$ se i vertici i e j non sono connessi direttamente
- $L_k(n)$: costo del cammino minimo, individuato dall'algoritmo fino al passo k , tra il nodo s ed un generico nodo n

Algoritmo di Dijkstra

- **Inizializzazione ($k=1$)**

- $T_1 = \{s\}$
- $L_1(n) = c(s,n)$ per $n \neq s$

- **Aggiunta di un nodo (passo $1 \leq k \leq N$)**

- trovare $x \notin T_{k-1}$ tale che

$$L_{k-1}(x) = \min_{j \notin T_{k-1}} \{L_{k-1}(j)\}$$

- aggiungere all'insieme T_{k-1} il nodo x ed il ramo incidente a x

- **Aggiornamento dei cammini minimi**

- $L_k(n) = \min [L_{k-1}(n), L_{k-1}(x) + c(x,n)]$ per tutti i valori di $n \notin T_k$

Algoritmo di Dijkstra

- **Al termine**

- l'insieme T_N è uno spanning tree del grafo di partenza contenente i cammini a costo minimo tra il nodo sorgente e tutti gli altri nodi del grafo
- $L_N(n)$ indica il costo del cammino a costo minimo tra il nodo s ed il nodo n

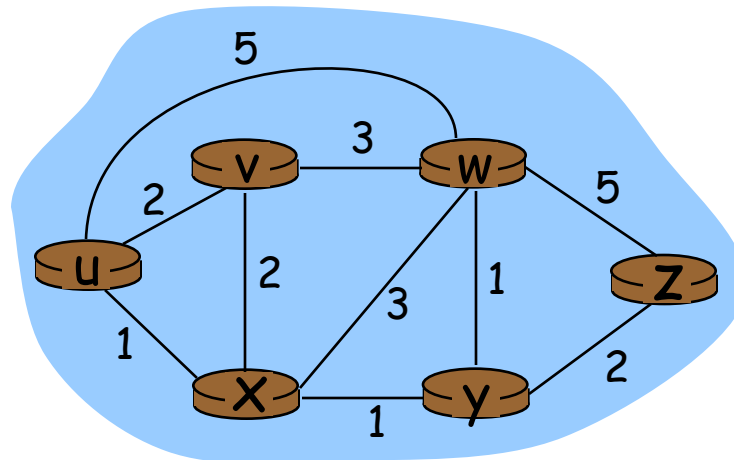
- **Si noti che**

- al passo k -mo viene aggiunto all'insieme T_{k-1} il k -mo nodo ed è individuato il cammino a costo minimo tra il tale nodo ed il nodo sorgente
- questo cammino transita esclusivamente attraverso i nodi sinora compresi nell'insieme T_{k-1}

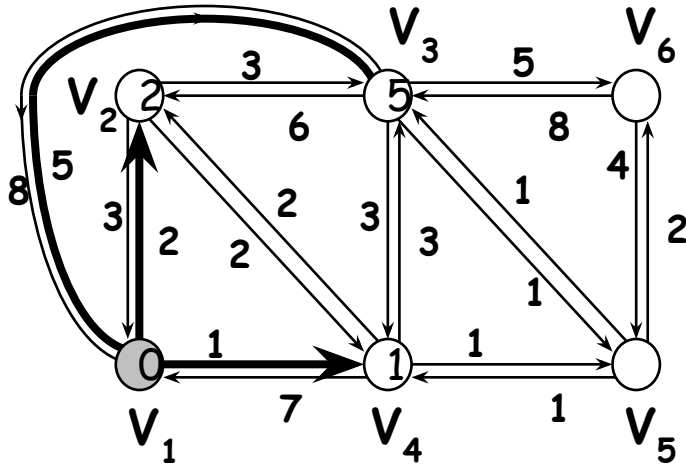
- **La complessità dell'algoritmo è $O(N^2)$**

Algoritmo di Dijkstra: esempio

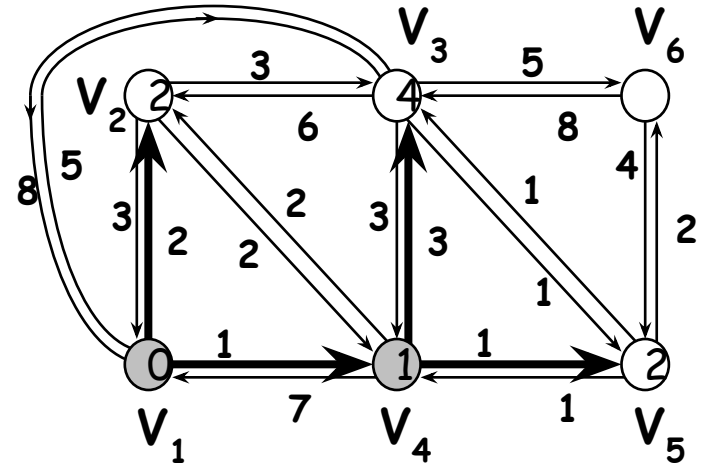
passo	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Esempio Dijkstra Algorithm - 1(2)

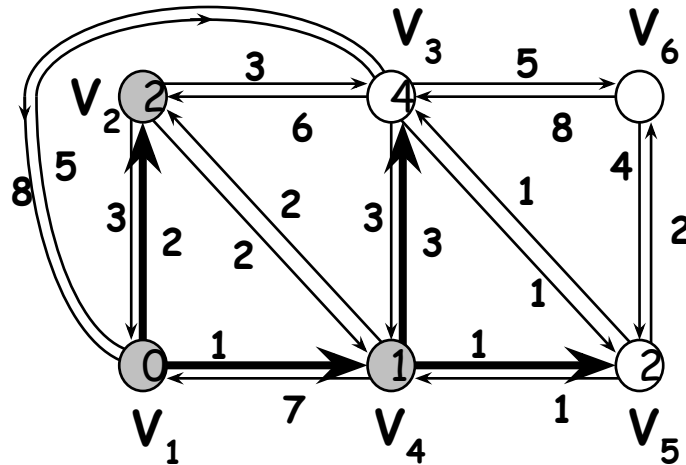


$$T_1 = \{1\}$$

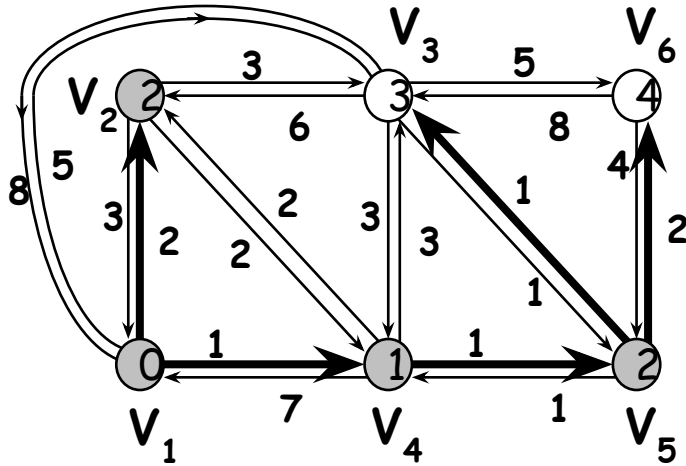


$$T_2 = \{1, 4\}$$

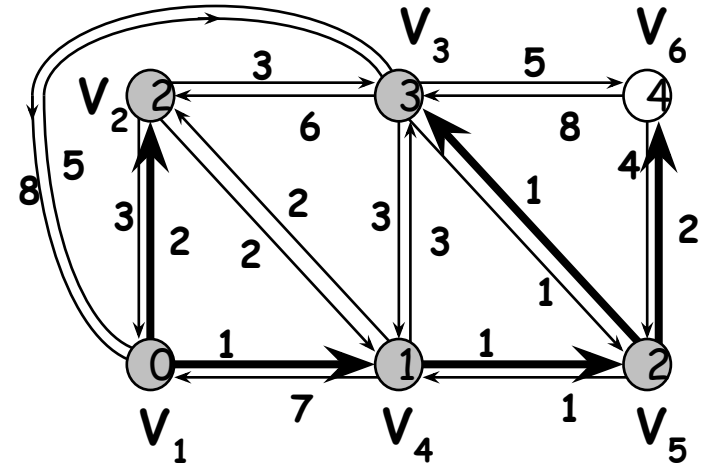
$$T_3 = \{1, 2, 4\}$$



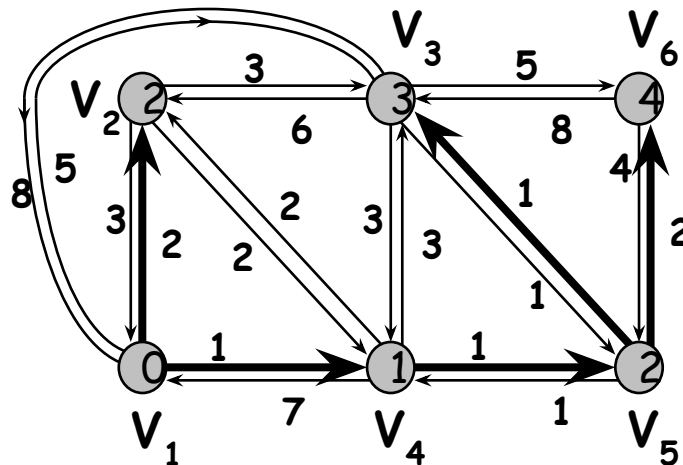
Esempio Dijkstra Algorithm - 2(2)



$$T_4 = \{1, 2, 4, 5\}$$



$$T_5 = \{1, 2, 3, 4, 5\}$$



$$T_6 = \{1, 2, 3, 4, 5, 6\}$$

Algoritmo di Dijkstra

- **Complessità dell'algoritmo (n nodi)**
 - Ciascuna iterazione: controllo su tutti i nodi, w, non in N
$$\frac{n(n+1)}{2} \rightarrow O(n^2)$$
- **La più efficiente implementazione possibile**
 $O(n \log n)$
- **Possibili oscillazioni**
- **Es. costo del collegamento = quantità di traffico trasportato**

