

# **Il livello di rete**

## **Routing**

**Liceo G.B. Brocchi - Bassano del Grappa (VI)**  
**Liceo Scientifico - opzione scienze applicate**  
Giovanni Mazzocchin

# Il livello di rete

- Il **livello fisico** si occupa dell'invio dei bit sotto forma di segnali su un canale
- Il **sottolivello MAC** si occupa dell'accesso ad un canale broadcast. È particolarmente importante nelle reti wireless, che sono naturalmente broadcast
- Il livello **data link** organizza i bit in frame ed effettua operazioni di error control e flow control tra macchine direttamente collegate da un canale fisico
- Il **livello di rete** permette lo scambio di pacchetti su un'intera rete, o addirittura su più reti

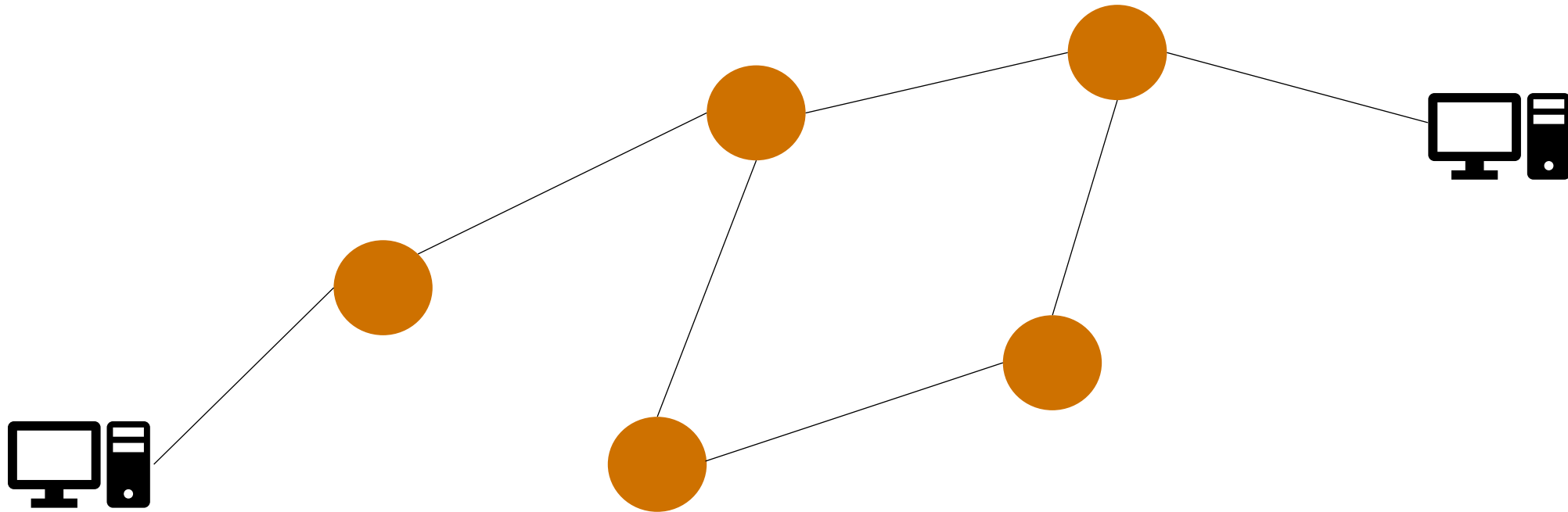
# Il livello di rete

- Compiti del livello di rete:
  - inoltrare i pacchetti
  - evitare congestioni (*congestion control*). Pensate al traffico in un città, va gestito in qualche modo!
  - fornire un servizio *best effort* al livello superiore: sarà il livello di trasporto ad occuparsi di correggere errori, recuperare pacchetti perduti e riordinare sequenze di pacchetti arrivati al destinatario fuori sequenza
  - Quality of Service (QoS): diverse applicazioni richiedono diversi requisiti, ad esempio:
    - il trasferimento di file richiede banda elevata e non ha necessità particolari sulla latenza
    - l'online gaming richiede banda elevata e bassissima latenza
  - trasmettere i pacchetti da una rete ad un'altra. Anche lo switching a livello data link è una forma di instradamento, ma lavora su una singola rete. Internet è una rete di reti, quindi non funzionerebbe senza questo livello
  - nascondere ai livelli superiori i dettagli relativi alla struttura della rete

# Il livello di rete

- **IP** (***I**nternet **P**rotocol*) è il protocollo di rete più importante dello stack TCP/IP
- Fornisce un servizio *connectionless*
- Nei servizi *connectionless* i pacchetti vengono anche detti *datagram*

# Il livello di rete



**i nodi della rete sono i famosi router**

# Il routing (instradamento)

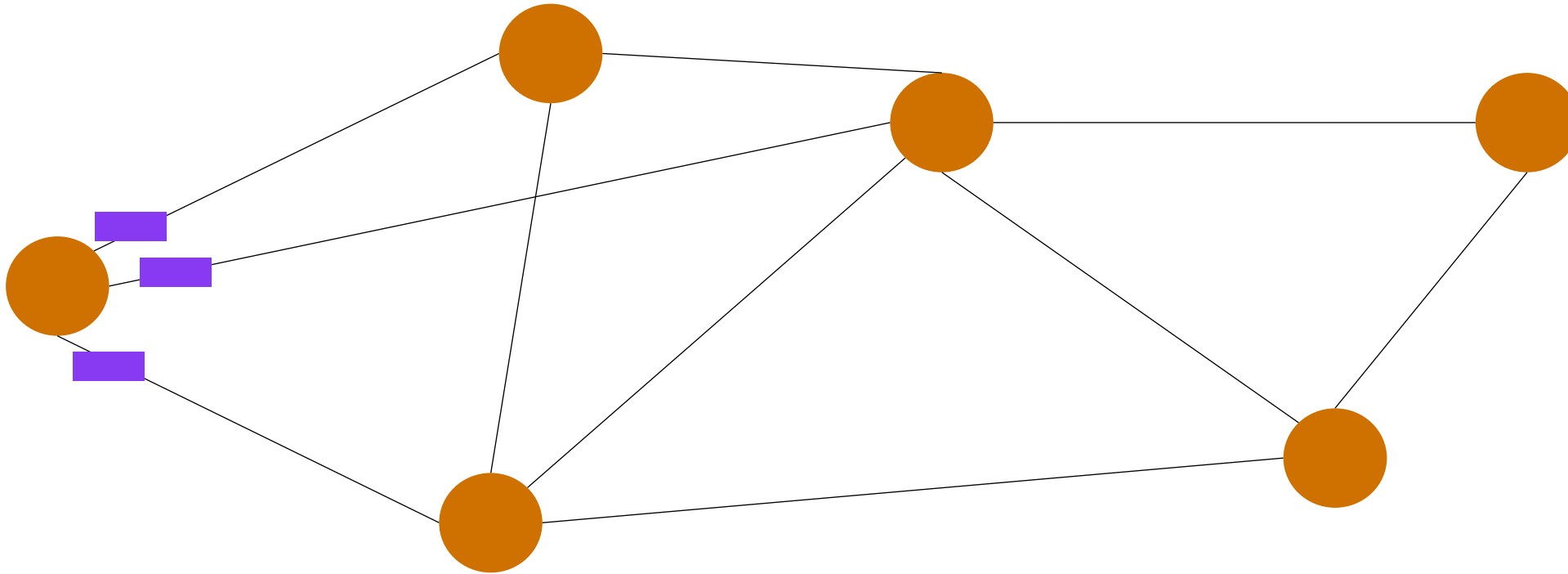
- La funzione principale del livello di rete è fornire un servizio *best effort* di instradamento dei pacchetti dal mittente al destinatario
- Un **algoritmo di routing** è quella componente del livello di rete che stabilisce su quale interfaccia di uscita inoltrare un pacchetto in ingresso
- La commutazione di pacchetto rende indispensabili gli algoritmi di routing: sappiamo già infatti che con il *packet switching*, la rete stabilisce la *route* in modo indipendente per ogni pacchetto

# Il routing (instradamento)

- Un algoritmo di routing deve prima di tutto stabilire quale parametro ottimizzare. Potrebbe minimizzare:
  - la latenza media
  - il numero di hop per pacchetto
- **Static routing**: i path per ciascuna coppia di nodi della rete vengono calcolati a priori e non vengono più modificati
- **Dynamic routing**: i path cambiano in base ai cambiamenti nella topologia della rete (e.g. un router viene spento) e al traffico

# Flooding

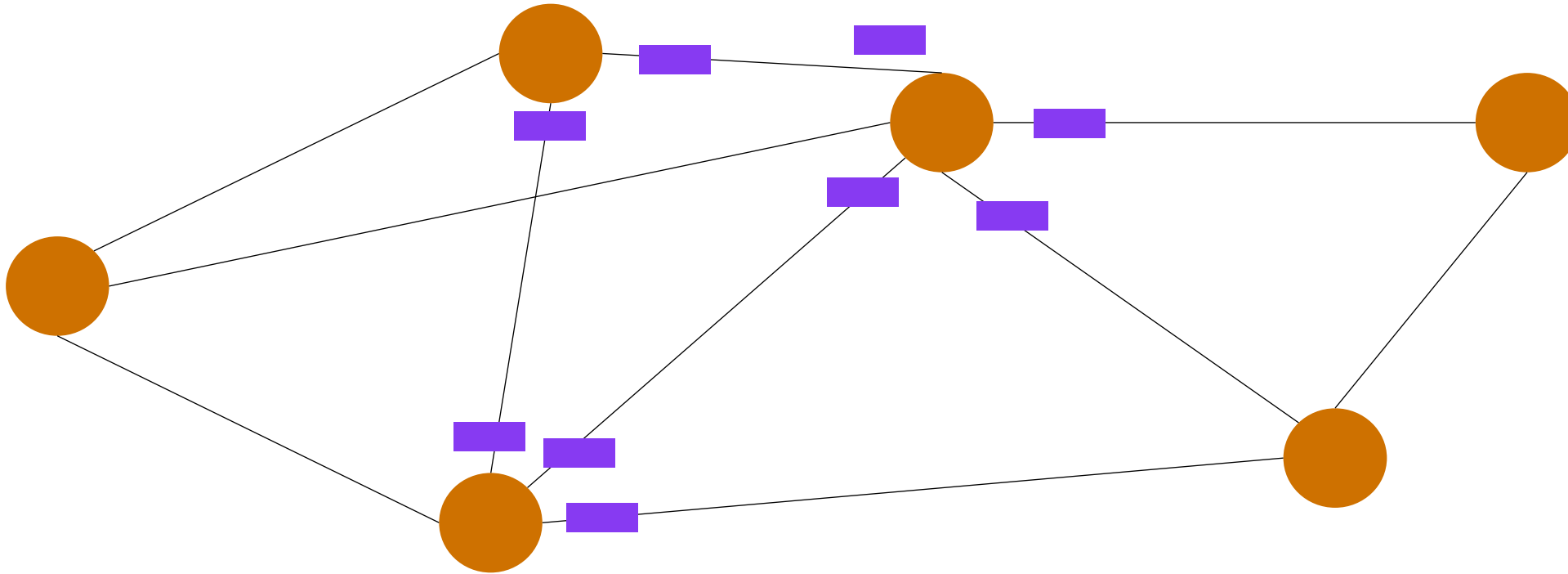
- Nel **flooding**, il router inoltra ciascun pacchetto su tutte le linee in uscita, tranne che su quella su cui è stato ricevuto





# Flooding

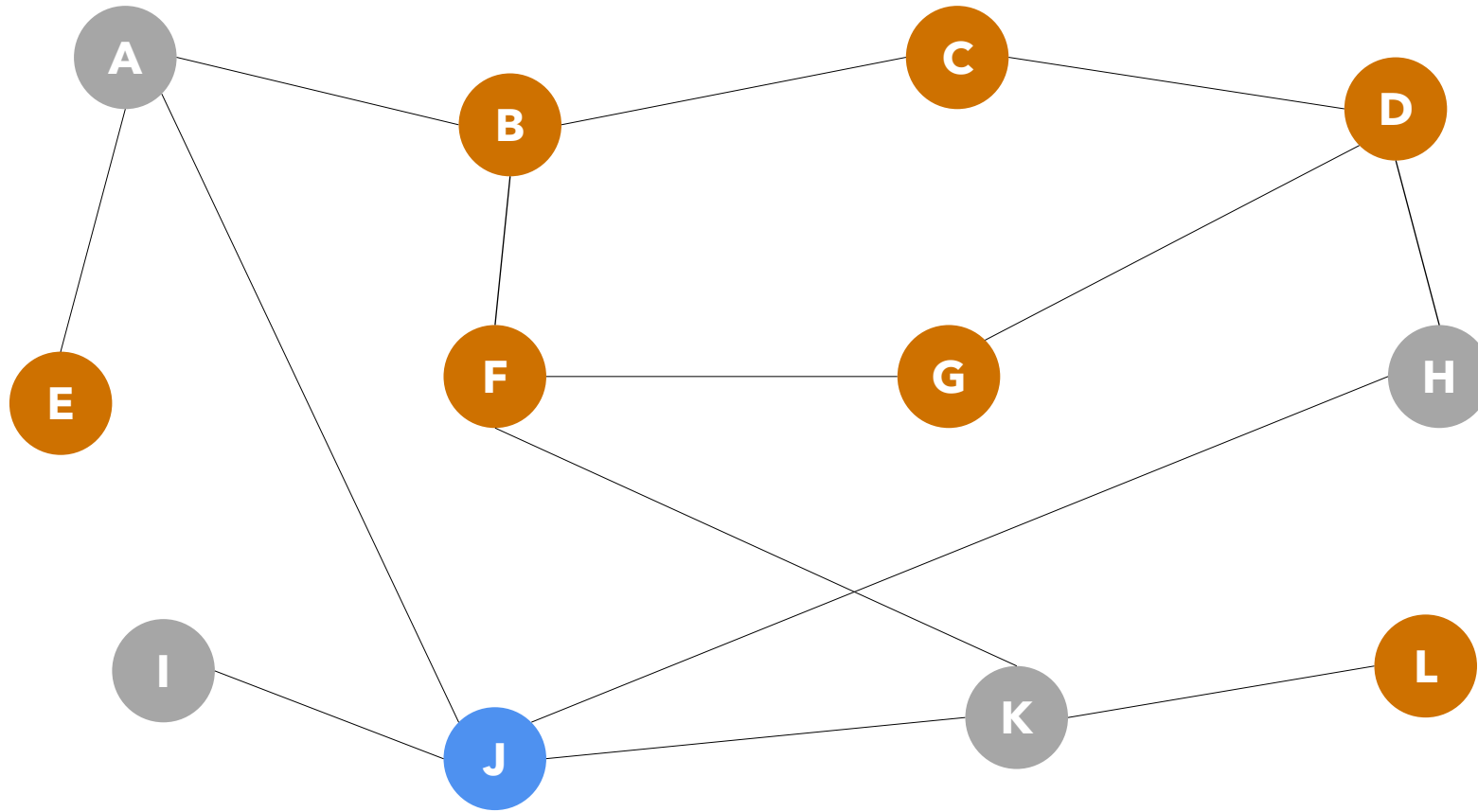
- Nel **flooding**, il router inoltra ciascun pacchetto su tutte le linee in uscita, tranne che su quella su cui è stato ricevuto



# Distance vector routing (routing dinamico)

- È stato il primo algoritmo di routing utilizzato da ARPANET
- È noto anche come **RIP** (**R**outing **I**nformation **P**rotocol)
- Ciascun router mantiene aggiornata una tabella (**routing table**) contenente:
  - il costo della *best route* nota verso ciascun nodo della rete
  - il *next-hop*: il nodo (parte della best route) su cui inoltrare il pacchetto per arrivare a ciascuna destinazione
- Ogni  $t$  millisecondi, ogni router manda la propria tabella aggiornata a tutti i suoi vicini
- La distanza potrebbe essere la latenza, il numero di hop, etc... in questo esempio useremo la latenza
  - ciascun router conosce la latenza verso i router vicini (quelli a cui è direttamente collegato)
- Si basa sull'**algoritmo di Bellman-Ford**

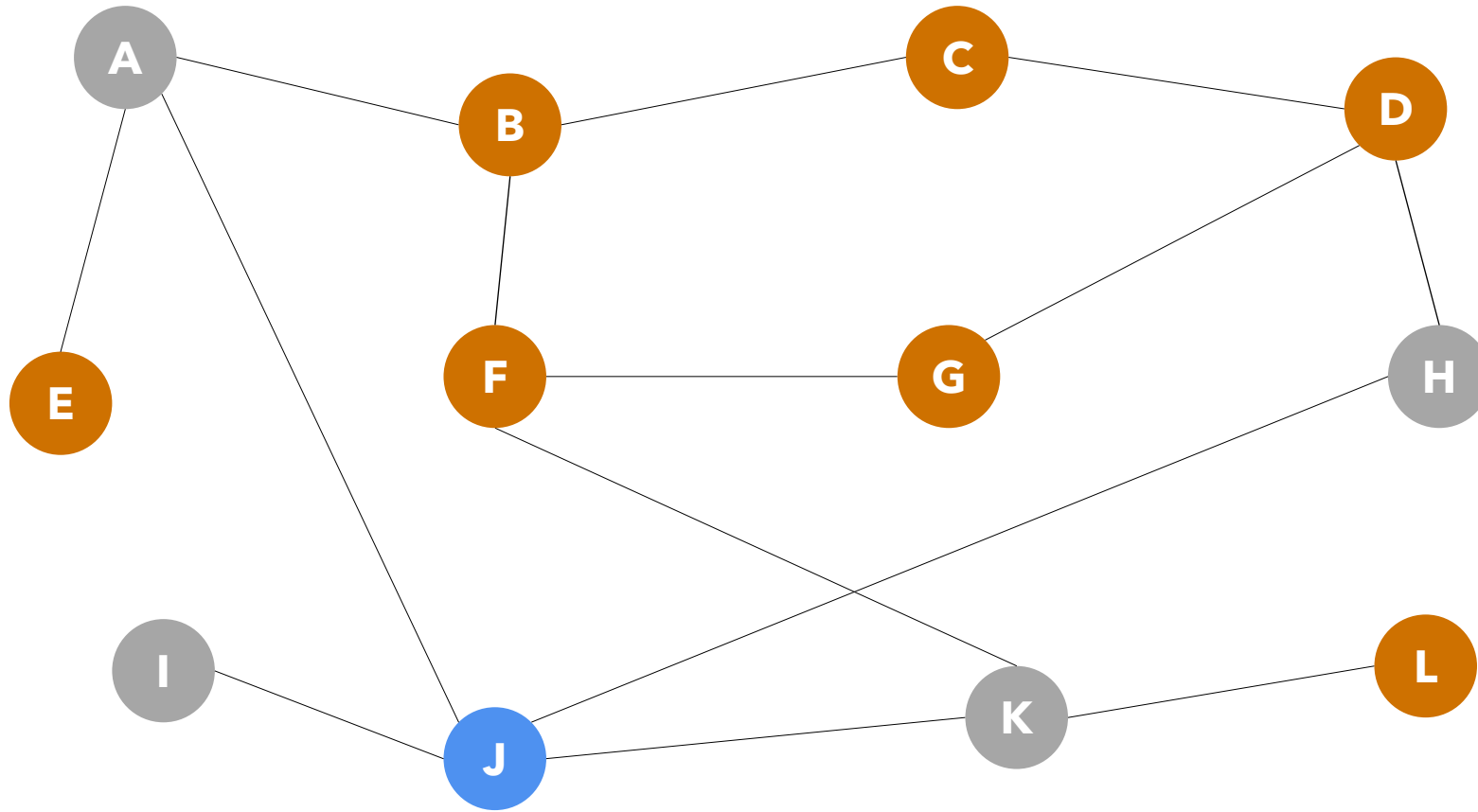
# Distance vector routing (routing dinamico)



**J's table at time t1**

dest	delay	next-hop
A	6	A
B	20	H
C	25	K
D	50	A
E	10	K
etc...		

# Distance vector routing (routing dinamico)



**delays J-neighbours**

<b>neighbour</b>	<b>delay</b>
------------------	--------------

A	6
---	---

I	8
---	---

H	4
---	---

K	12
---	----

# Distance vector routing (routing dinamico)

al tempo  $t_2$ , J riceve i vettori dei suoi 4 vicini

A's vector	
dest	delay
A	0
B	12
...	...
...	...

# Distance vector routing (routing dinamico)

al tempo  $t_2$ , J riceve i vettori dei suoi 4 vicini

I's vector	
dest	delay
...	...
B	36
I	0
...	...

# Distance vector routing (routing dinamico)

al tempo  $t_2$ , J riceve i vettori dei suoi 4 vicini

H's vector	
dest	delay
...	...
B	31
H	0
...	...

# Distance vector routing (routing dinamico)

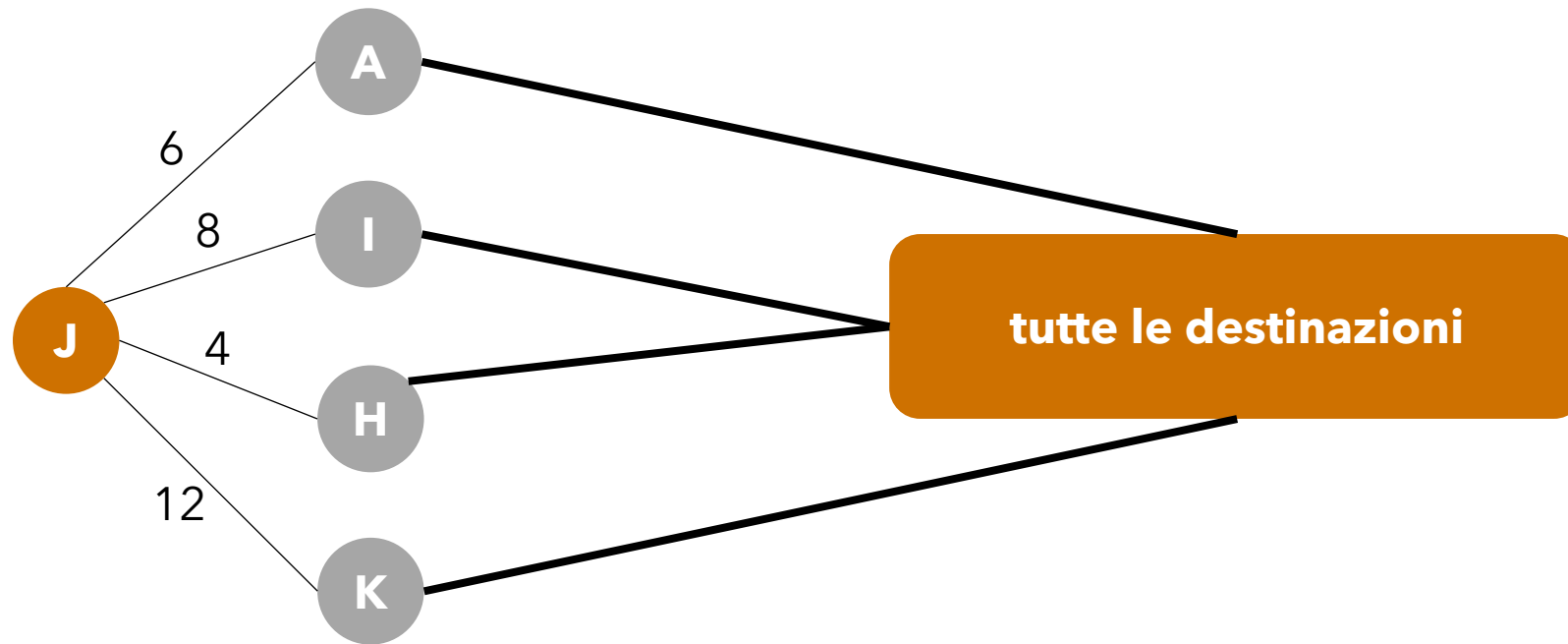
al tempo  $t_2$ , J riceve i vettori dei suoi 4 vicini

K's vector	
dest	delay
...	...
B	28
K	0
...	...



# Distance vector routing (routing dinamico)

- Per ogni destinazione, J calcola un nuovo delay sulla base dei vector ricevuti e dei delay noti verso i vicini
- Il delay verso i vicini va sommato ai delay dei vector ricevuti



# Distance vector routing (routing dinamico)

- Per ogni destinazione, J calcola un nuovo delay sulla base dei vector ricevuti dai vicini
- Il nuovo next-hop corrisponderà al ritardo minimo
- Ricalcolo della route per la destinazione **B**

A's vector		I's vector		H's vector		K's vector	
dest	delay	dest	delay	dest	delay	dest	delay
B	12	B	36	B	31	B	28

$\min(\text{delay J-A} + 12, \text{delay J-I} + 36, \text{delay J-H} + 31, \text{delay J-K} + 28) =$   
 $\min(6 + 12, 8 + 36, 4 + 31, 12 + 28) = 18 \rightarrow \text{next hop: A}$

# Link state routing (routing dinamico)

- Il distance vector routing è stato abbandonato da ARPANET nel 1979, e sostituito dal **link state routing**
- Anche in questo caso serve una metrica di costo per i link che connettono i router (delay, costo inversamente proporzionale alla bandwidth etc...)
- Varianti del link state routing sono utilizzate ancora oggi sotto il nome di **IS-IS** e **OSPF**, componenti fondamentali di Internet
- Nel link state routing, ciascun router conosce l'intera topologia della rete
- Il link state routing si basa sull'**algoritmo di Dijkstra**

# Link state routing (routing dinamico)

- Ciascun router deve:
  1. calcolare il costo per raggiungere tutti i propri vicini
  2. costruire un pacchetto contenente tutte le informazioni relative ai costi per raggiungere i vicini
  3. inviare il pacchetto a tutti i router della rete
  4. ricevere i pacchetti informativi da tutti i router della rete
  5. calcolare i cammini minimi verso tutti gli altri router, utilizzando l'algoritmo di Dijkstra