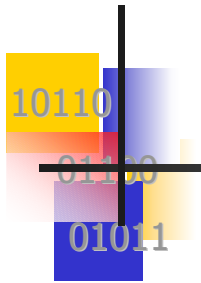


## IL LIVELLO RETE



# Il Livello RETE

---

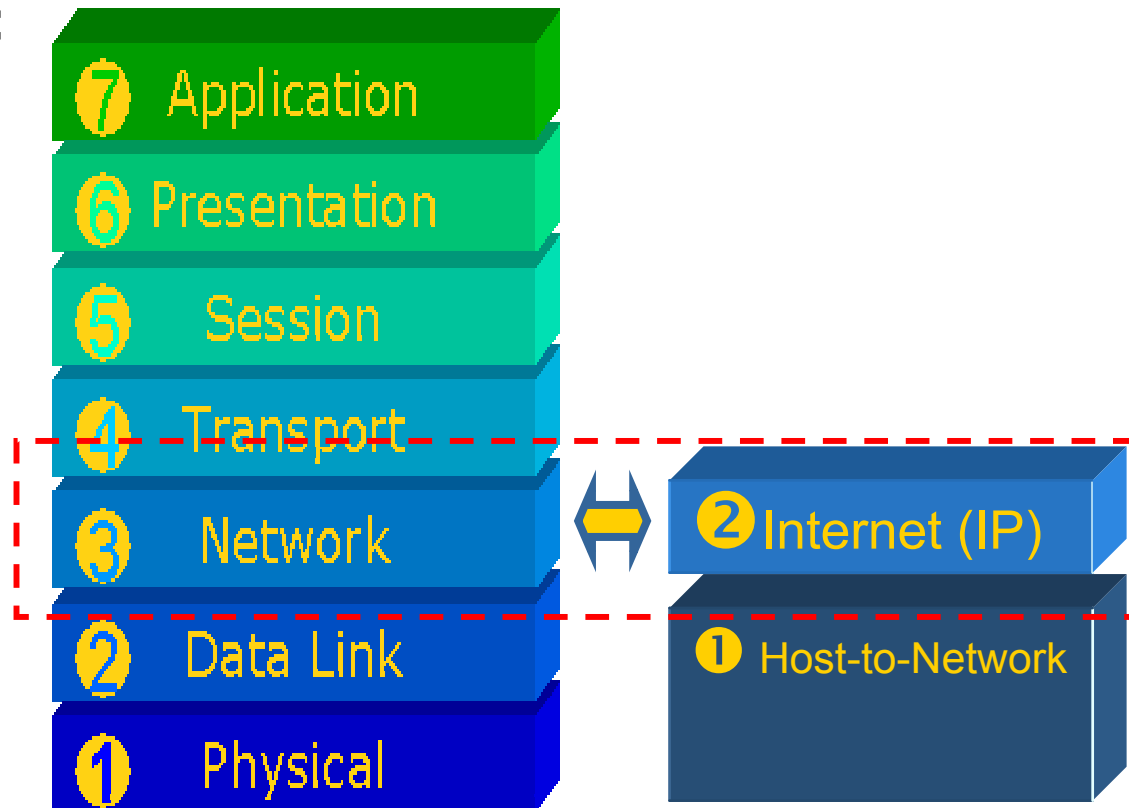
- Servizi del livello Rete
- Organizzazione interna
- Livello Rete basato su Circuito Virtuale
- Livello Rete basato su Datagram
- Algoritmi di Routing
- Problema della congestione
- Internetworking



# Livello RETE

Funzionalità del livello Rete:

1. Concatena le reti data link.
2. Controlla il cammino e il flusso di pacchetti (routing su rete e indirizzamento su rete).
3. Gestisce la congestione della rete.
4. Implementa l'interfaccia necessaria alla comunicazione di reti di tipo diverso.

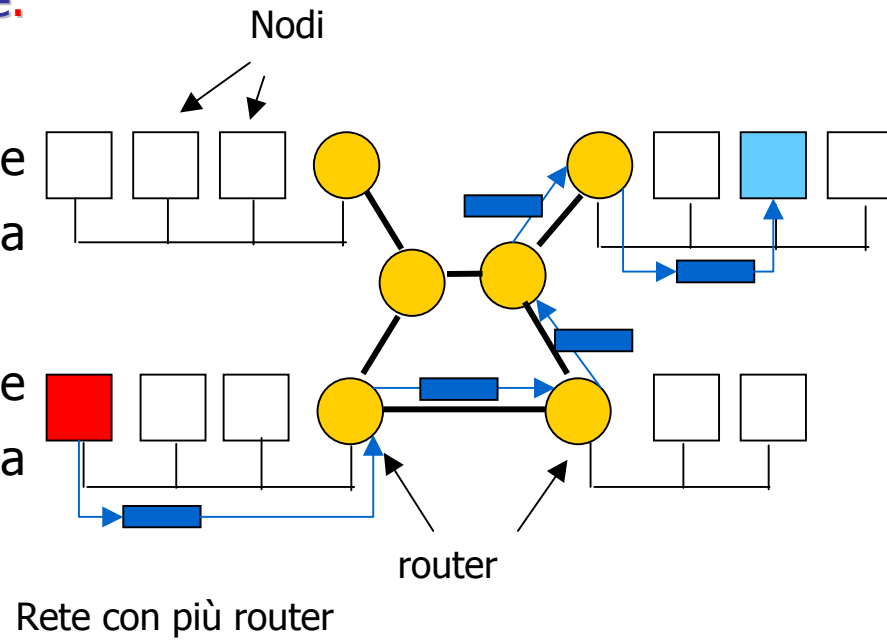




# Obiettivi

1. Trasmissione punto-a-punto di pacchetti dalla sorgente alla destinazione.
2. Deve essere nota la topologia della rete (insieme dei **router**)
3. Connessione di reti diverse (**internetworking**)
4. Schema di **indirizzamento globale**.

- ➔ I router instradano i pacchetti sulle linee di comunicazione in modo da consegnarli alla destinazione.
- ➔ I percorsi sono scelti in modo ottimale (es. cammino minimo, cammino senza sovraccarichi)





# Approcci e Organizzazione del livello Rete

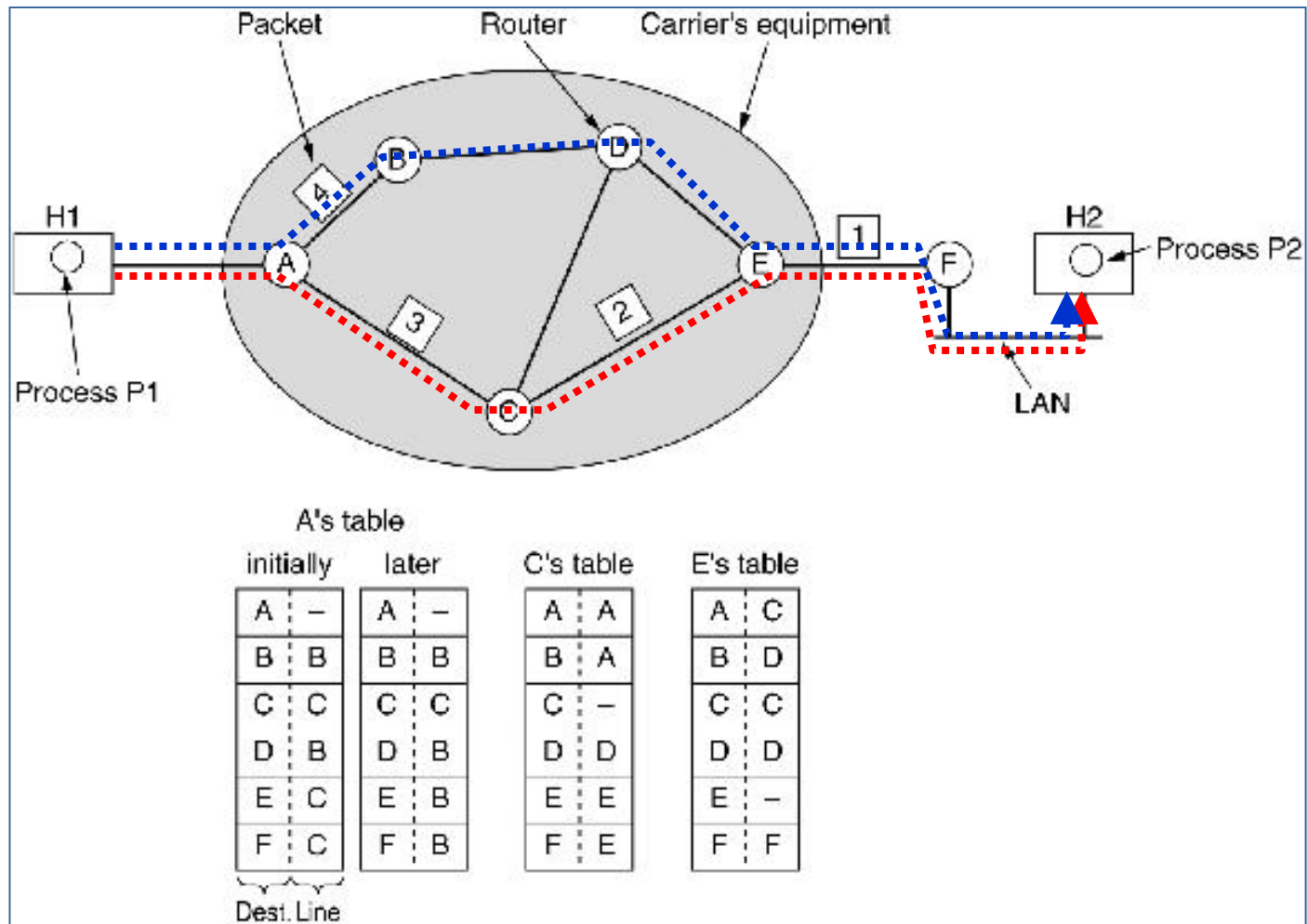
## ■ Con connessione (circuiti virtuali)

- Non si deve scegliere il percorso per ogni pacchetto.
- La creazione della connessione prevede la definizione del percorso di consegna dalla sorgente alla destinazione.
- I router devono mantenere memoria dei circuiti virtuali che passano attraverso di essi.
- I pacchetti contengono indicazione del circuito virtuale a cui appartengono.

## ■ Senza connessione (datagram)

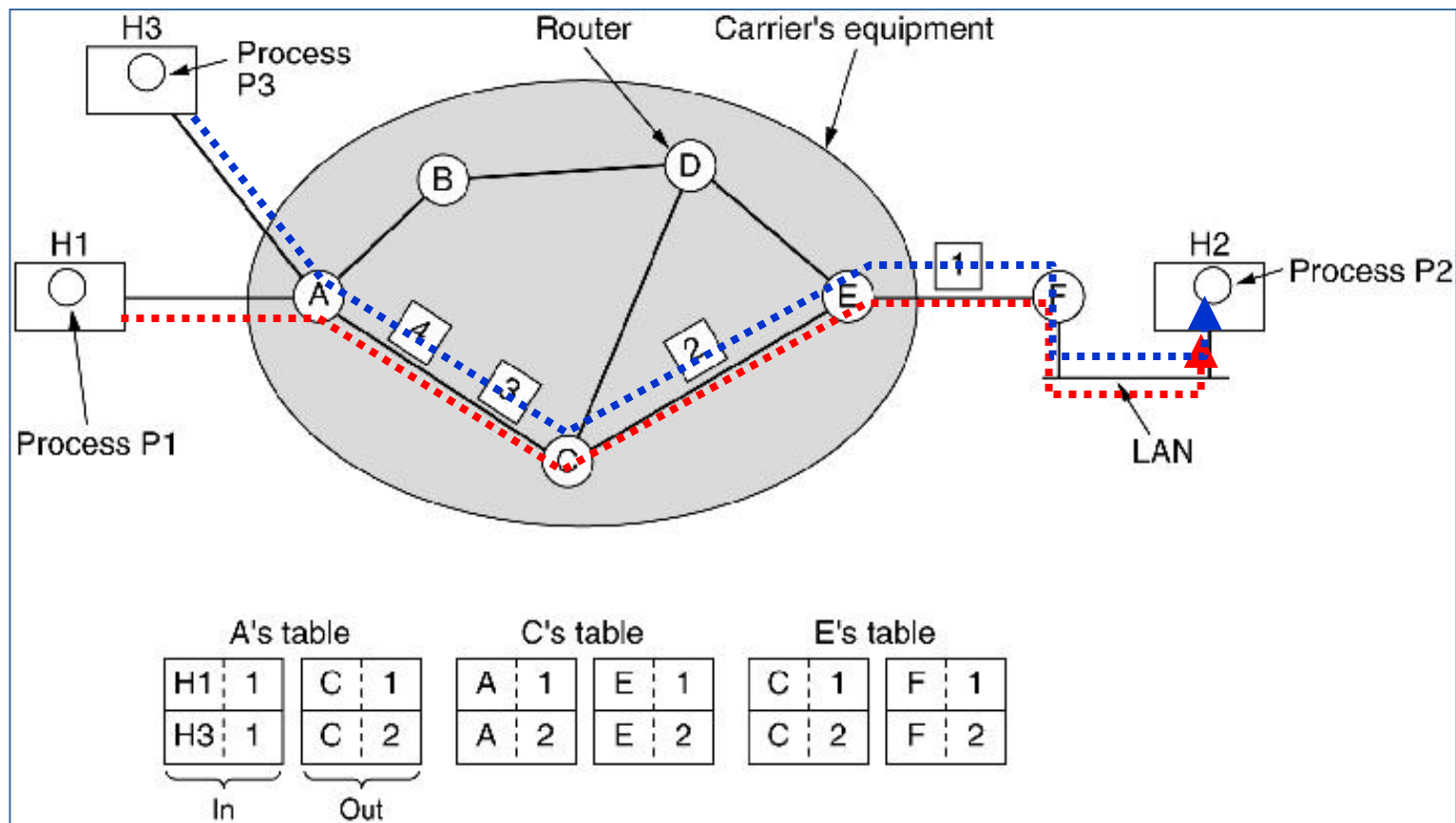
- I pacchetti fra due host possono seguire anche percorsi diversi.
- I router hanno tabelle che indicano quale linea di uscita utilizzare per ogni possibile router destinazione.

# Implementazione di un servizio Datagram



## Tabelle di instradamento

# Implementazione di un servizio con Circuito Virt.



## Tabelle dei circuiti virtuali



# Reti basate su Datagram o su Circuiti Virtuali

Caratteristica	Reti basate su datagram	Reti basate su circuito virtuale
Creazione circuito	Non richiesto	Richiesto
Indirizzamento	Ogni pacchetto contiene gli indirizzi sorgente e destinazione completi	Ogni pacchetto contiene un piccolo numero VC (Virtual Circuit)
Informazioni di stato	La sottorete non conserva informazioni di stato	Ogni circuito virtuale richiede spazio di tabella nella sottorete
Instradamento	Ogni pacchetto è instradato indipendentemente	Percorso scelto alla creazione del circuito virtuale: tutti i pacchetti seguono questo percorso
Effetti dei guasti nei router	Nessuno, a parte i pacchetti persi durante il guasto	Tutti i circuiti virtuali che passano attraverso il router guasto vengono terminati
Controllo della congestione	Complesso	Semplice se può essere allocato spazio sufficiente in anticipo per ogni circuito virtuale





# Routing: Definizione e Requisiti

## → Definizione

**Un algoritmo di routing decide quale connessione usare per instradare i pacchetti dalla macchina sorgente alla macchina destinazione.**

## → Requisiti:

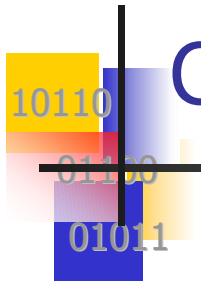
- **Correttezza** nel routing dei datagram.
- **Semplicità** e **efficienza** delle implementazioni del routing.
- **Robustezza**: una rete importante deve continuare a funzionare senza guasti globali
- **Stabilità**: le informazioni nelle tabelle di routing devono raggiungere un punto di equilibrio
- **Imparzialità** ed **Ottimalità** nel routing dei datagrammi



# Generazione delle Tabele di Instradamento

---

- **Manuale**
  - Tabella creata a direttamente ad hoc
  - Utile in piccole reti
  - Utile se l'instradamento non cambia.
- **Automatico**
  - Il software crea/aggiorna le tabelle
  - Necessario in grandi reti
  - L'instradamento cambia in caso di guasti.



# Classi di algoritmi di routing

Gli algoritmi automatici possono essere:

- **Non adattivi** (o **Routing Statico**):  
questi algoritmi calcolano i percorsi quando la rete non è ancora attiva.
- **Adattivi** (o **Routing Dinamico**):  
questi algoritmi modificano i percorsi in base alle situazioni di traffico ed ad altre informazioni locali come congestione, guasti, ecc.



# Tipi di algoritmi di routing

## Algoritmi di routing

- **Isolati**

routing calcolato con sole informazioni locali, indipendentemente dal resto (stato degli altri nodi e stato della rete). Es. SNA/IBM

- **Centralizzati**

un centro di controllo conosce lo stato globale e calcola il cammino ottimo per ogni coppia (mittente,destinatario) e dirama le tabelle. Es. TYMNET

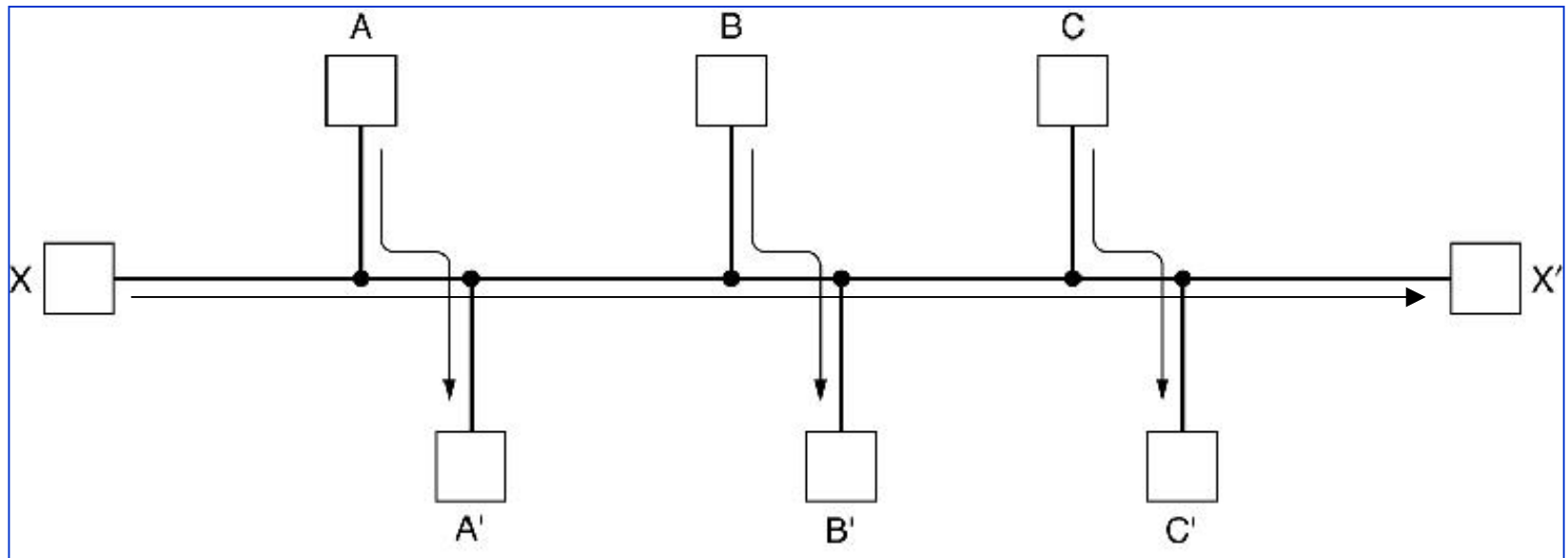
- **Distribuiti**

i nodi cooperano e comunicano frequentemente il proprio stato e quello della rete. Es. Internet

- **Misti**

combinazione di politiche isolate e centralizzate. Es. TRANSPAC

# Imparzialità e ottimalità



Compromesso tra efficienza globale e imparzialità verso connessioni locali.



# Principio di ottimalità e sink tree

**Principio di ottimalità:** se il router  $j$  è nel cammino ottimo fra  $i$  e  $k$ , allora anche il cammino ottimo fra  $j$  e  $k$  è sulla stessa strada.

**sink tree** : l'insieme dei cammini ottimali da tutte le sorgenti ad una destinazione.

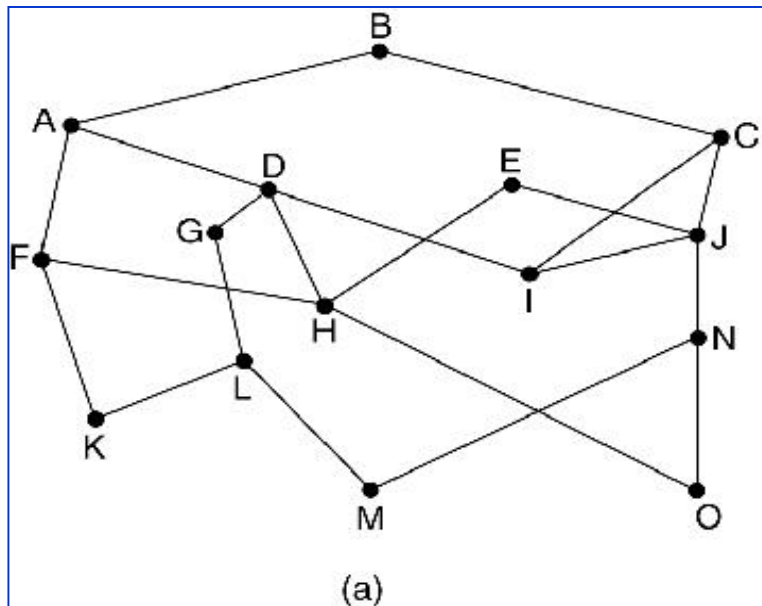
L'obiettivo di tutti gli algoritmi di routing è **quello di scoprire e utilizzare i sink tree di tutti i router.**

10110

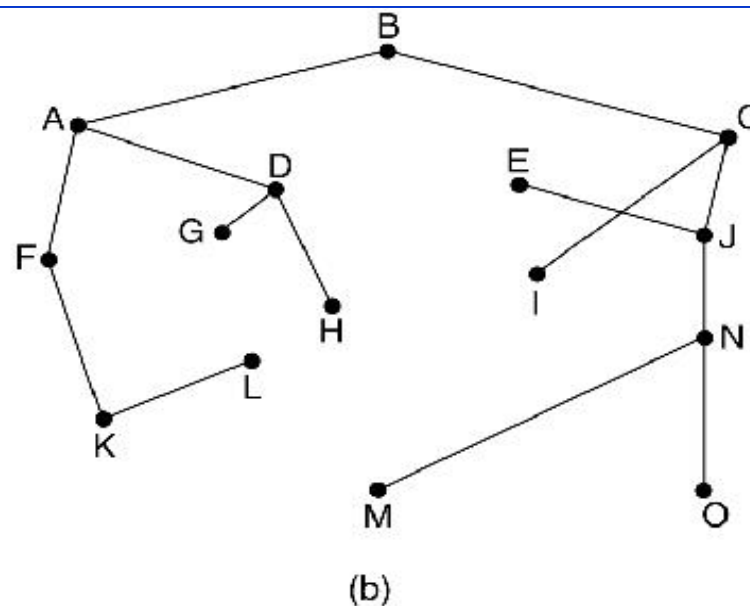
01100

01011

# Sink tree : esempio



(a) Esempio di rete

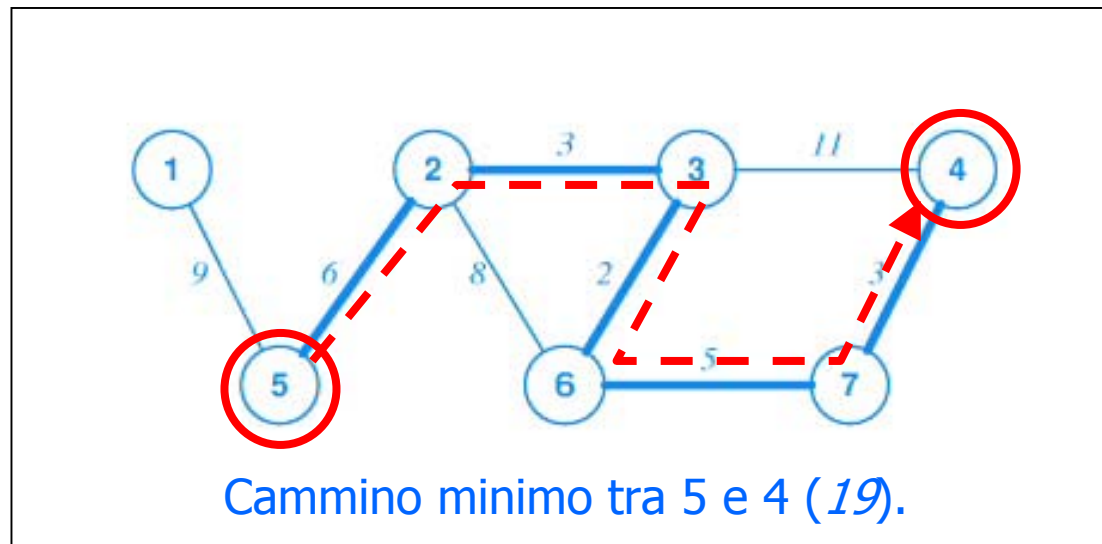


(b) Un sink tree per il router B

Il **principio di ottimalità** e i **sink tree** forniscono un modello per misurare gli algoritmi di routing.

# Routing con cammino minimo: l'algoritmo di Dijkstra

- **L'algoritmo di Dijkstra** (1959) lavora su grafi orientati, che hanno pesi non negativi sui collegamenti.
- Questo algoritmo trova i percorsi più brevi tra un nodo di partenza e tutti gli altri.



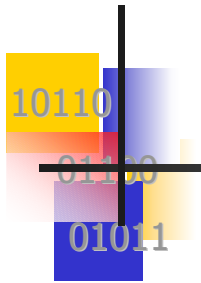




# Routing su cammino minimo

---

- Il principio di base è quello di costruire un grafo della rete, dove ogni nodo del grafo rappresenta un router ed ogni arco del grafo rappresenta una linea di comunicazione (chiamata anche **canale**).
- Per scegliere un percorso tra due router, l'algoritmo cerca nel grafo il cammino più breve tra di essi.
- Metriche possibili
  - **distanza**,
  - **costi**,
  - **capacità**.



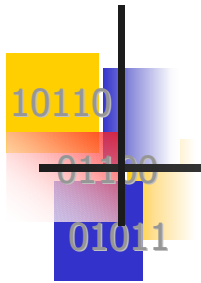
# Routing Flooding

---

Ogni pacchetto in arrivo viene inoltrato su ogni linea in uscita eccetto quella da cui è arrivato. Per prevenire la duplicazione eccessiva dei pacchetti:

- Questi vengono dotati di un contatore. Quando questo contatore raggiunge lo 0, il pacchetto viene eliminato.
- I router tengono traccia dei messaggi ricevuti e ritrasmessi, e non duplicano messaggi già replicati.
- Nel flooding selettivo invece, i pacchetti in arrivo vengono replicati ma solo sulle linee che approssimativamente vanno nella direzione richiesta dalla sorgente.

Gli algoritmi di flooding vengono utilizzati come benchmark, perché scelgono sempre il cammino più breve, in quanto lo ricercano in parallelo.



# Routing basato sullo stato dei canali

---

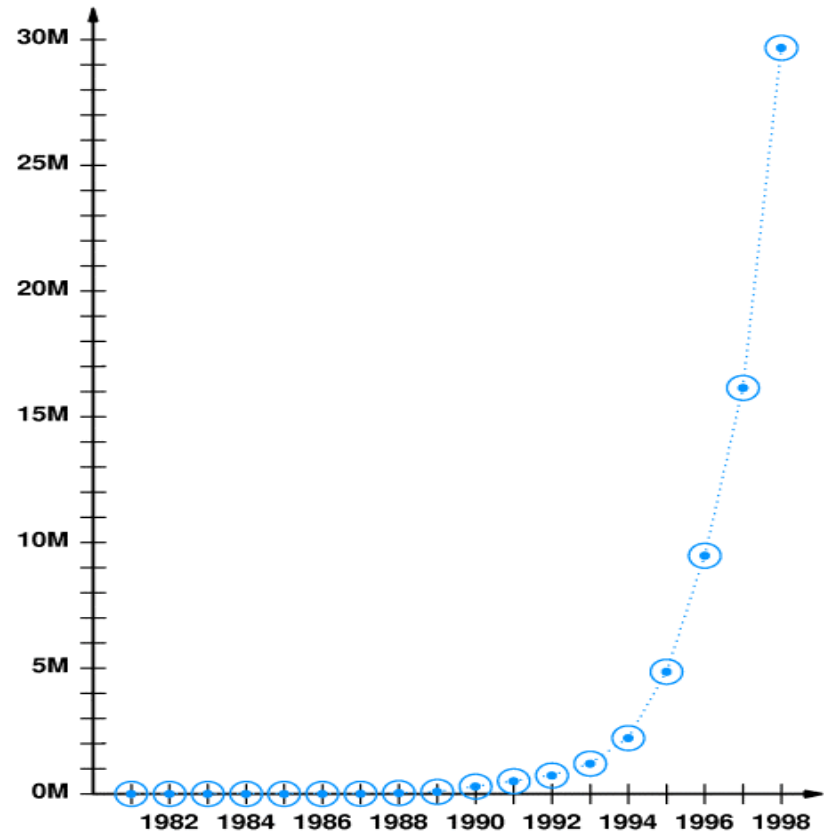
Il routing basato sullo stato dei canali venne attivato nel 1979 su ARPANET.

1. Scoprire i propri vicini e i loro indirizzi di rete spedendo uno speciale pacchetto di HELLO su ogni linea.
2. Misurare il ritardo o il costo per ognuno dei suoi vicini con speciali pacchetti ECHO.
3. Costruire un pacchetto contenente tutto quello che ha appena scoperto (link state). In ogni pacchetto vengono aggiunti: l'identità del router ed un numero di sequenza.
4. Spedire questo pacchetto a tutti i router effettuando un flooding.
5. Calcolare il cammino minimo per ogni altro router utilizzando l'algoritmo di Dijkstra



# Routing gerarchico

- Le tabelle di routing di Internet diventano sempre più grandi.
- Quindi si divide il gruppo di router in regioni.
- Ogni router conosce i dettagli della propria regione e come comunicare con le altre, ma non conosce la loro struttura interna.



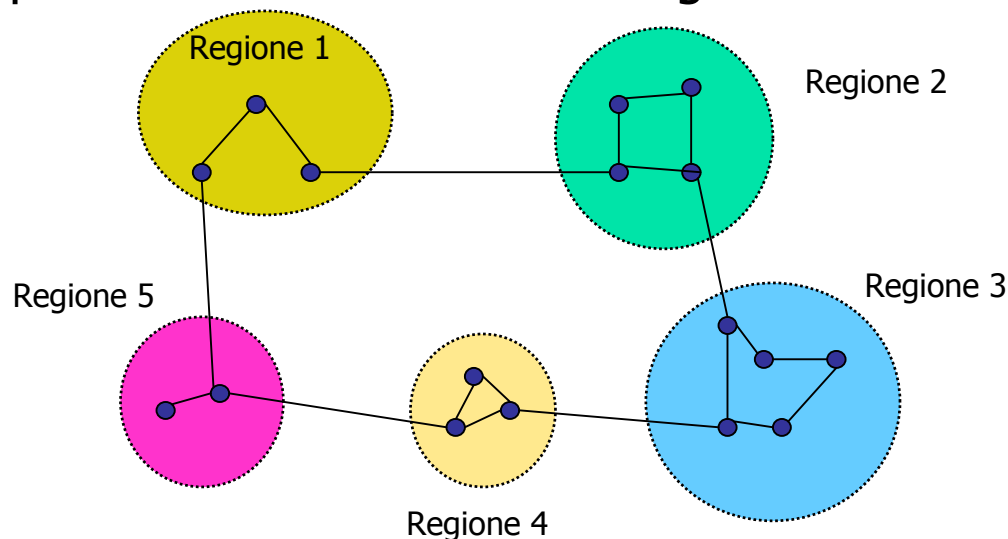


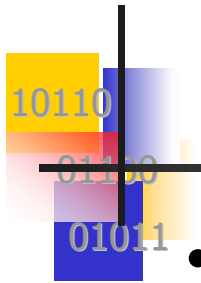
# Routing gerarchico

Due (o più) livelli di routing:

- un primo livello di **routing interno** ad ogni regione
- un secondo livello di routing fra tutti i **router di confine**.

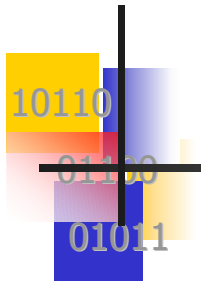
- **router interni**: sanno come arrivare a tutti gli altri router della regione;
- **router di confine**: particolari router a cui i router interni che devono spedire a router di un'altra regione inviano i dati.





# Routing per Host Mobili

- Servono per servire **utenti mobili** che si connettono alla rete da punti diversi.
- Si usa una **locazione base** che non cambia mai, un **agente base** che tiene traccia degli host mobili in altre aree e un **agente straniero** che tiene traccia degli host mobili nell'area.
- Un **host mobile** si connette con **l'agente straniero** nella area dove si trova che contatterà **l'agente base** della sua locazione base.
- Una comunicazione verso **l'agente mobile** passa dal suo **agente base** verso **l'agente straniero** dell'area in cui esso si trova.
- Le successive vanno direttamente all'agente straniero dell'area interessata senza andare verso la locazione base.



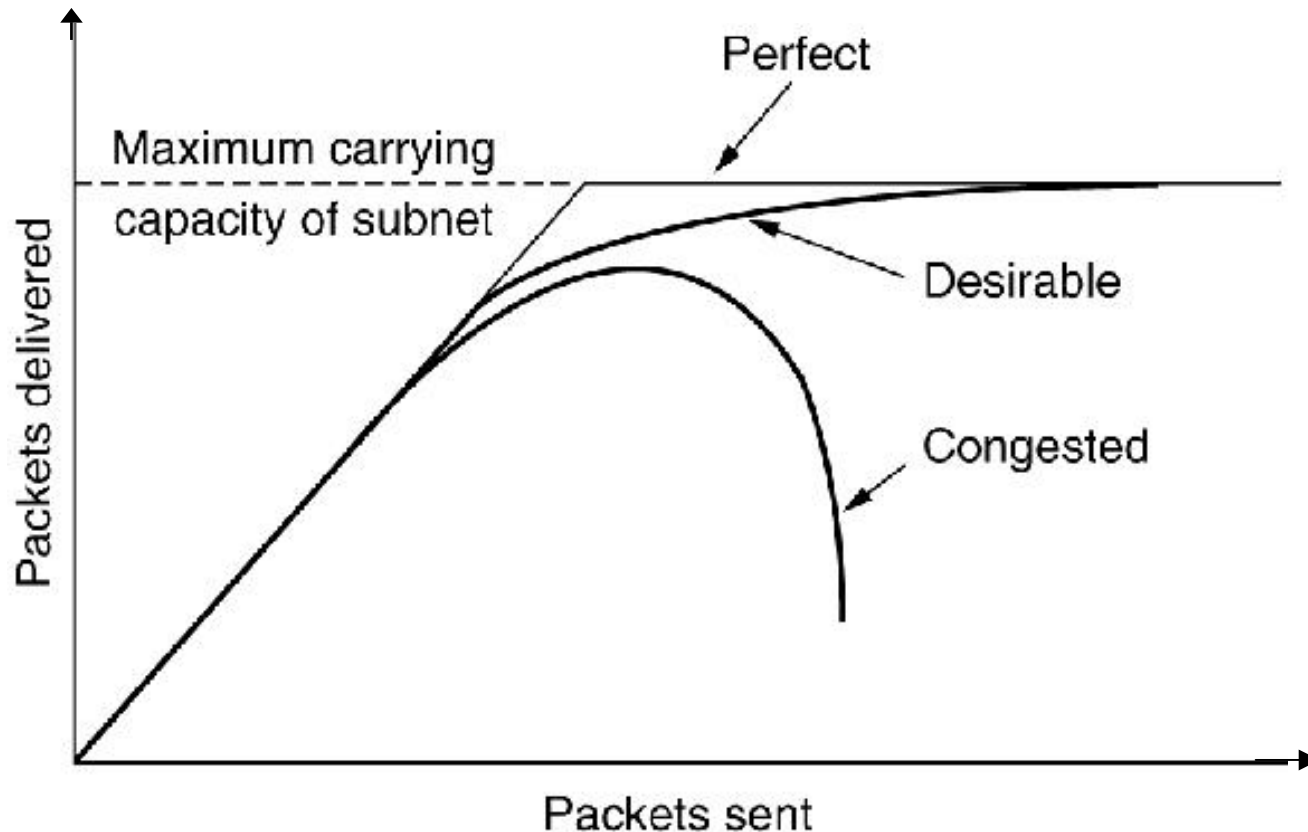
# Routing Broadcast

- E' necessario quando un host deve inviare un messaggio a molti nodi (**multicast**) o a tutti i nodi (**broadcast**) della rete.
- Non conviene inviare il messaggio a tutte le destinazioni direttamente.
- Approcci:
  - Flooding,
  - routing miltidestinazione,
  - routing con spanning tree (albero di ricoprimento).

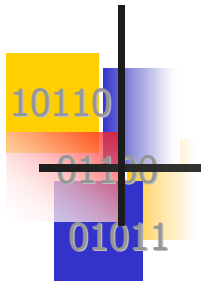


# Controllo della congestione

- Quando nella rete (o in una sottorete) sono presenti troppi pacchetti le prestazioni degradano (sia ha **congestione**).







# Controllo della congestione

## Problema di gestione della rete

Congestione del traffico e degradazione delle prestazioni

- buffer limitati nel router
- processore troppo lento nel router
- linea di trasmissione troppo lenta.

## Propagazione

***controllo della congestione*** ***è diverso*** dal controllo di flusso nei livelli 2, 3 e 4 (singola connessione sorgente-destinazione)

Approcci :

- ***ciclo aperto*** (senza controreazione)
- ***ciclo chiuso*** (con controreazione).

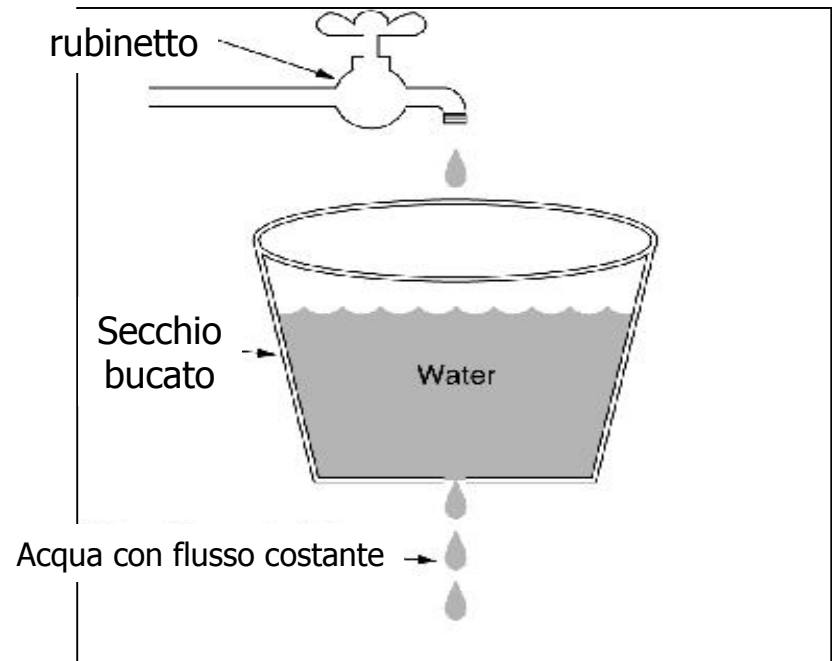


# Algoritmo del secchio bucato

## Algoritmo leaky bucket

Regolazione dell'input a ritmo costante

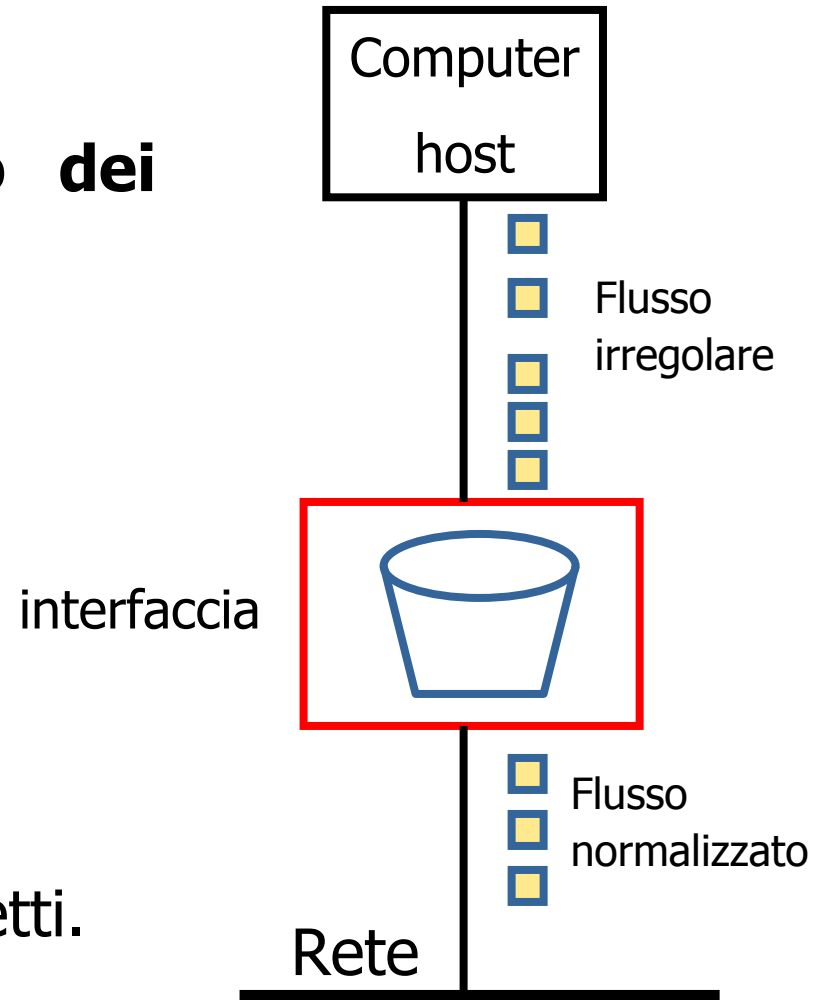
- modello del secchio bucato per regolare i pacchetti in rete con data rate fissato ( $k$  bps)
- mantiene nel buffer i pacchetti accodati per la trasmissione
- possibile perdita.

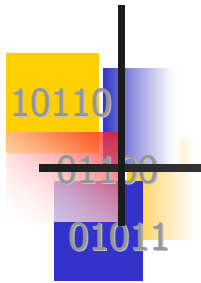


# Algoritmo del secchio bucato

## Regolazione del flusso dei pacchetti

Se data rate medio  $\leq k$  bps  
tutto OK,  
altrimenti si ha perdita pacchetti.

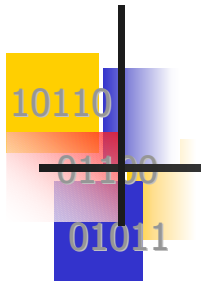




# Algoritmo token bucket

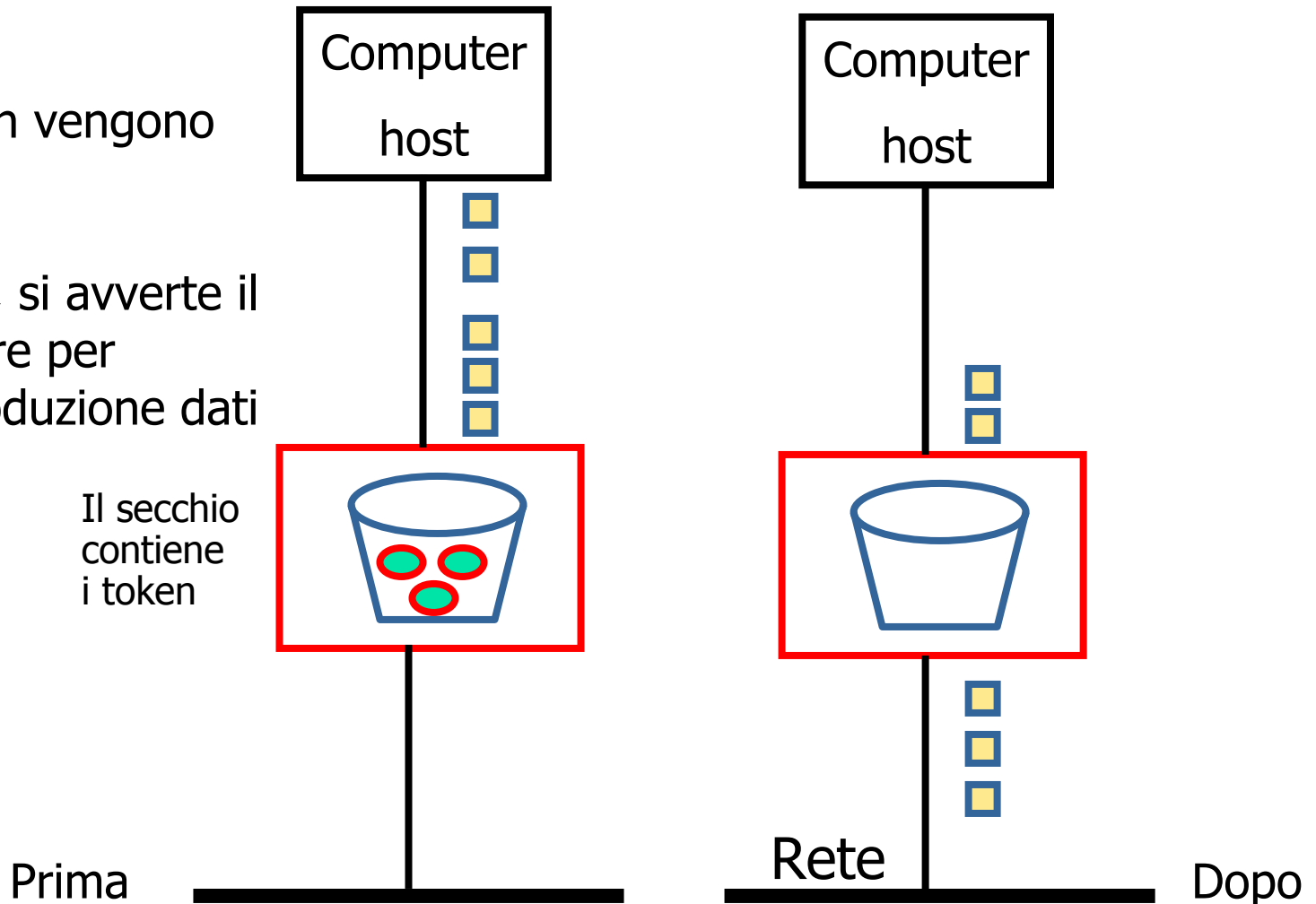
Consente irregolarità controllate nel flusso in uscita sulla rete

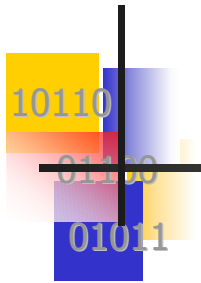
- si accumula un credito trasmissivo con un certo data rate quando non si trasmette nulla.
- quando si trasmette si sfrutta tutto il credito disponibile, alla massima velocità consentita dalla linea.
- secchio di **token**, creati con cadenza prefissata entro un valore max (M).
- per trasmettere un pacchetto deve essere disponibile un token.
- dati k token nel secchio e  $h > k$  pacchetti da trasmettere, i primi k sono trasmessi subito e gli altri aspettano.



# Algoritmo token bucket

- I pacchetti non vengono mai scartati
- Se necessario, si avverte il livello superiore per fermare la produzione dati



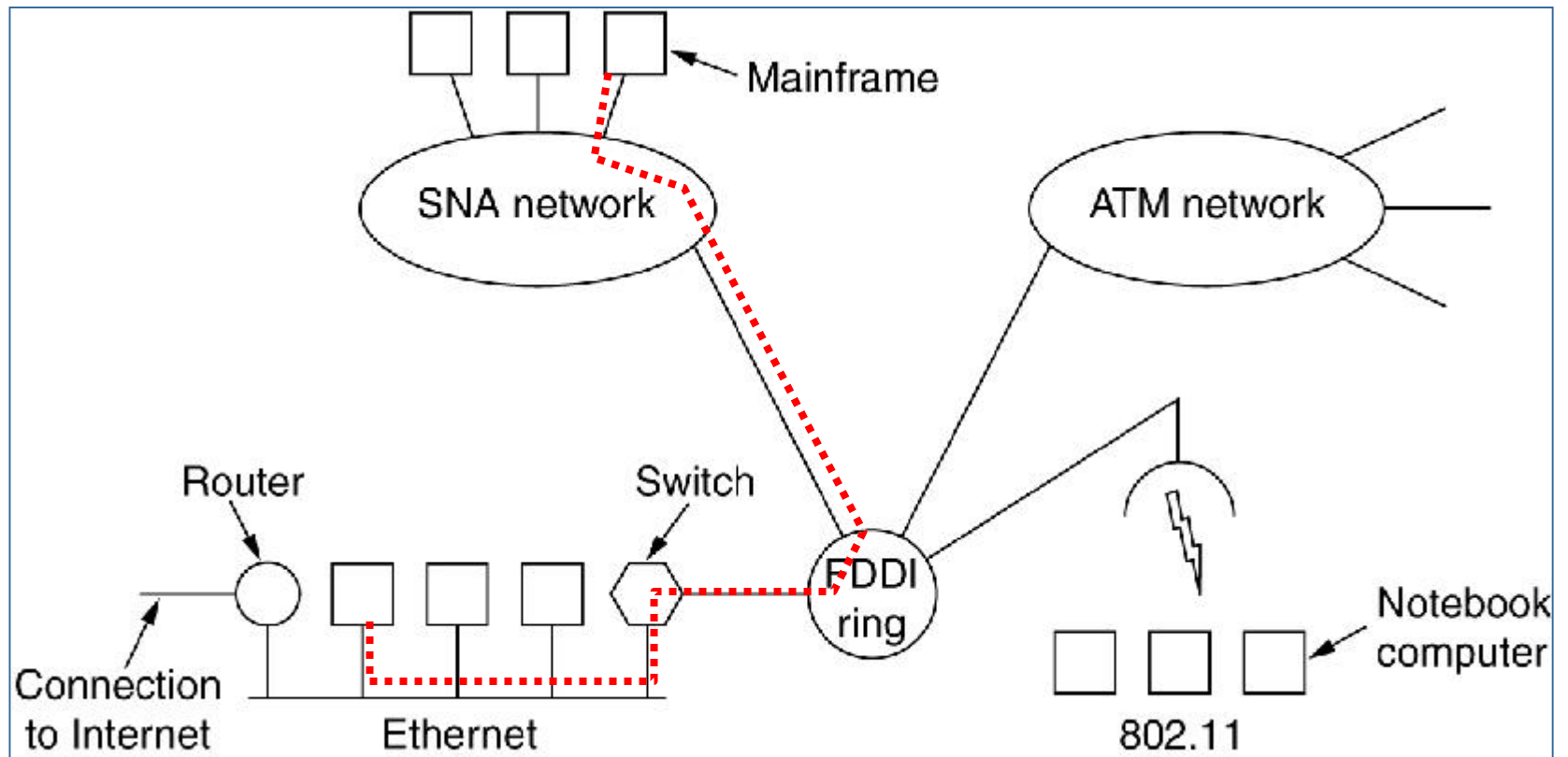


# Internetworking

- Connessione di reti eterogenee con diversi
  - formati di pacchetti e indirizzi
  - meccanismi di controllo dell'errore
  - meccanismi di controllo della congestione
  - max dimensione dei pacchetti.
  
- Diverse architetture di rete
  - servizi offerti dai vari livelli
  - modalità di indirizzamento
  - max dimensione dei pacchetti.

# Internetworking

## Interconnessione di reti diverse





# Tecniche e soluzioni per l'Internetworking

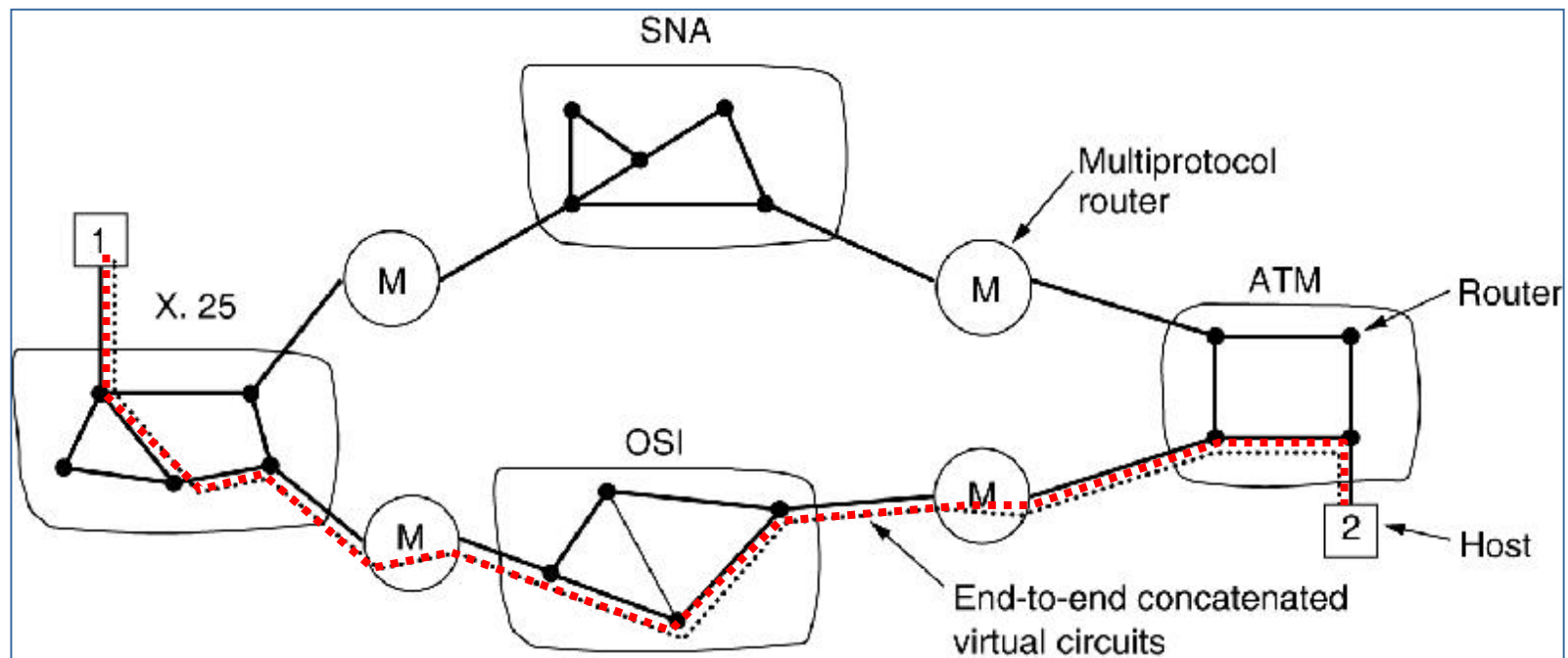
---

- Livello 1: **ripetitori** che copiano bit tra cavi.
- Livello 2: **bridge** memorizzano e inviano frame data link tra reti differenti.
- Livello 3: **router multiprotocollo** inviano pacchetti tra reti differenti.
- Livello 4: **gateway** di trasporto connettono flussi di dati.
- Livelli >4: **gateway** di applicazione con conversione di dati a livello delle applicazioni.



# Circuiti virtuali per l'Internetworking

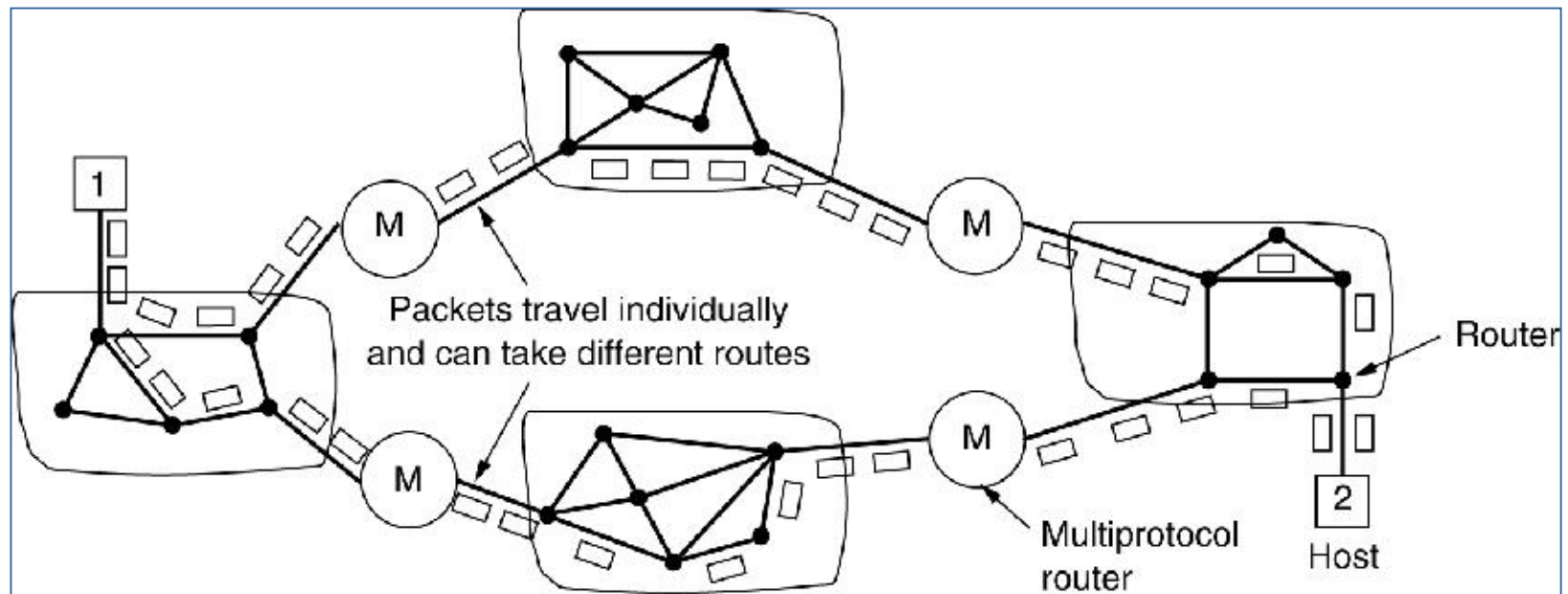
- Sequenza di circuiti virtuali tra più gateway.
- Conversione dei formati pacchetti e i numeri dei circuiti virtuali.





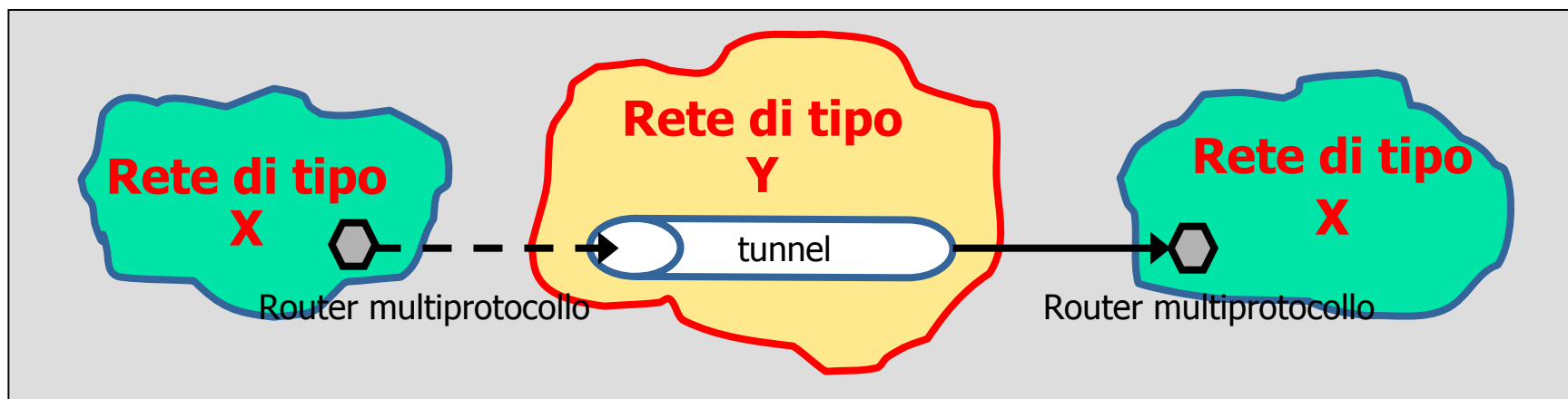
# Internetworking senza connessioni

- I pacchetti possono attraversare gateway differenti.
- Conversione necessaria ma maggiore adattività.



# Tunneling

- Quando l'host sorgente e l'host destinazione appartengono allo stesso tipo di rete ma bisogna attraversare reti differenti si usa la tecnica del **tunneling** (incapsulamento).



- La rete Y non ha router multiprotocollo.
- un router nelle due reti X è multiprotocollo e incapsula i pacchetti delle reti di tipo X dentro pacchetti di tipo Y, consegnandoli alla rete di tipo Y.

# Esempio di Tunneling

- L'invio di un pacchetto IP da una rete Ethernet ad un'altra rete Ethernet tramite una WAN.

