

Livello di rete

# Livello di rete

Lo scopo del livello di rete è quello di controllare l'intero movimento dei pacchetti dalla sorgente alla destinazione.

A differenza di quanto avviene a livello data-link, in cui le informazioni vengono trasmesse da un estremo all'altro di un canale fisico, il livello tre prevede l'attraversamento di tanti sistemi intermedi (router) della sottorete di comunicazione quanti è necessario.

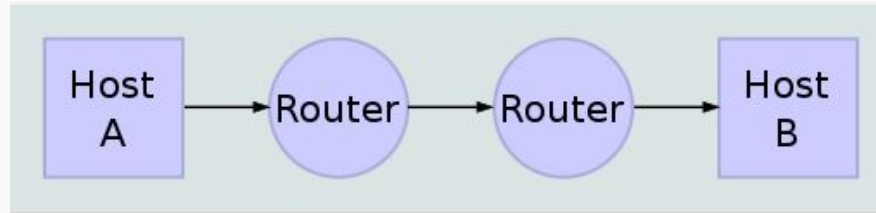
# Livello di rete

I compiti che svolti dal livello sono:

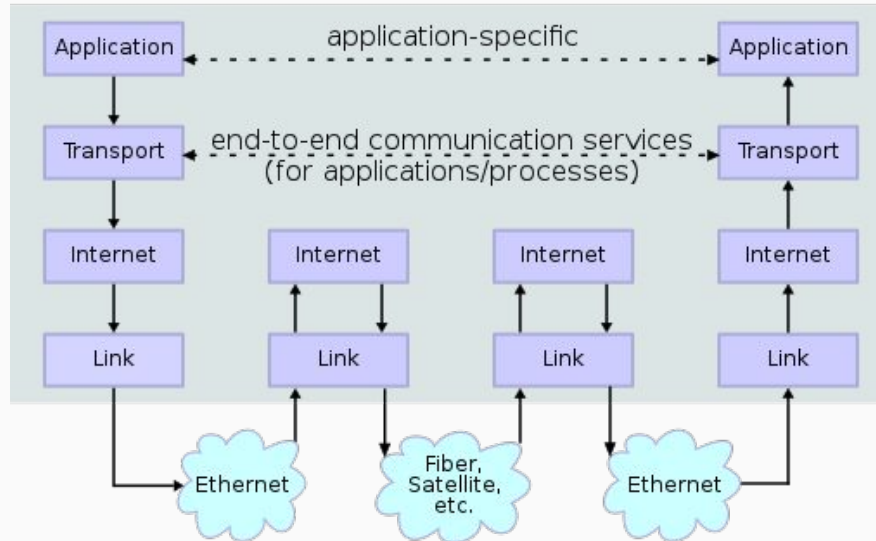
- instradare i pacchetti lungo la sotto rete di comunicazione (routing/forwarding dei pacchetti);
- controllare la congestione del traffico;
- interconnettere reti diverse (internetworking).

# Router

## Network Topology



## Data Flow



# Router

Ogni router ha alcune linee di ingresso e di uscita.

Il router deve ricevere i pacchetti in arrivo dalle varie linee, deve esaminare ciascun pacchetto e deve decidere su quale linea inoltrarlo.

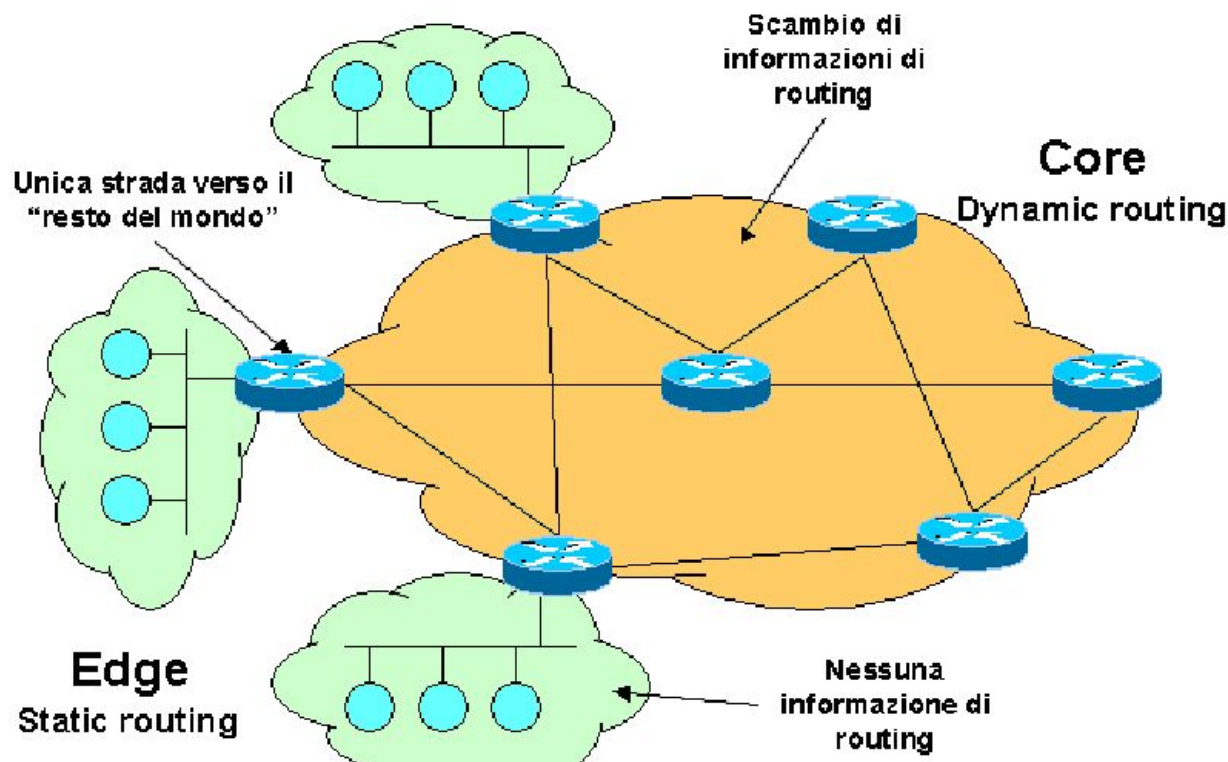
Il principale “strumento di lavoro” di un router è la sua **tabella di routing**, cioè una tabella che indica, per ogni possibile destinazione, quale linea in uscita utilizzare.

# Routing statico/dinamico

Il routing può essere:

- **statico**: le tabelle sono definite “off-line” e caricate nel router all'accensione;
- **dinamico** (o adattivo): sono i router stessi a calcolare periodicamente le proprie tabelle, in modo da tener conto di eventuali variazioni nella struttura della rete.

# Routing statico e dinamico



In Internet è utilizzato in mix di routing dinamico (nell'infrastruttura che collega i provider) e statico (collegamento di singole organizzazioni alla rete).

# Operatività del router

Normalmente a ogni linea del router è associata una coda dove vengono depositati i pacchetti in arrivo e una coda dove vengono depositati i pacchetti che devono uscire.

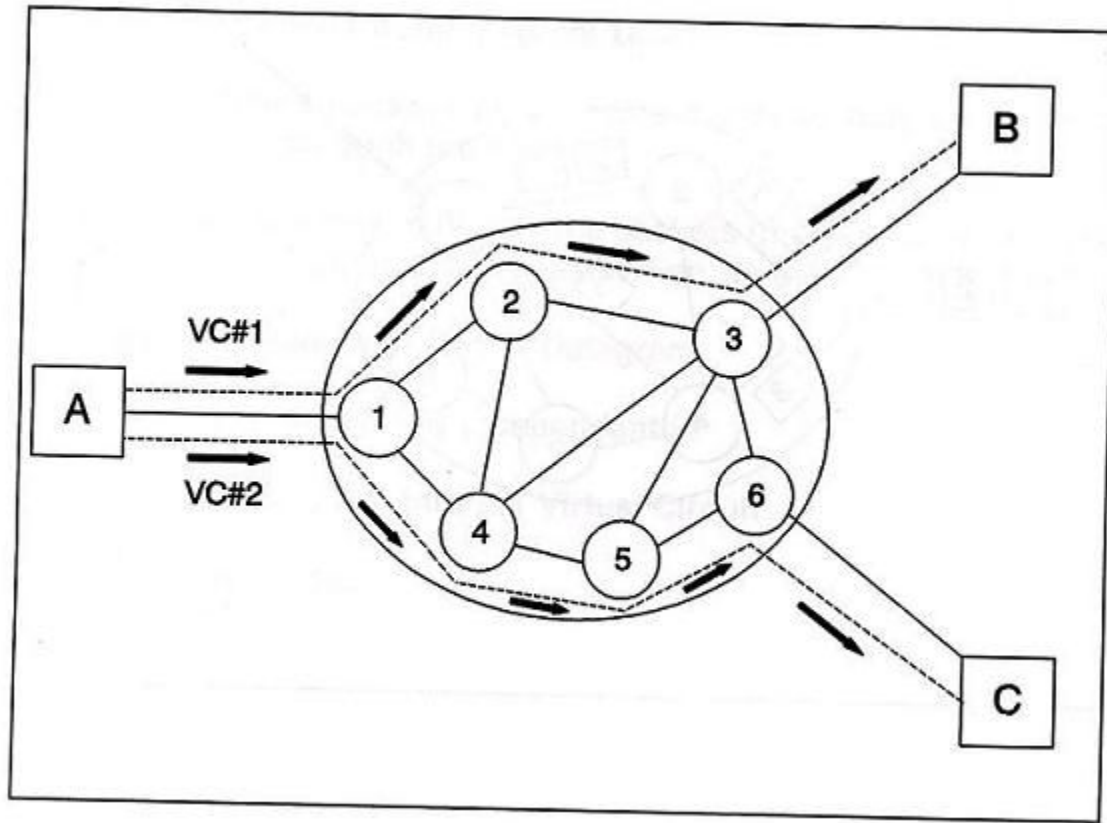
Nello svolgere il proprio compito quindi il router prende un pacchetto da un buffer di ingresso, lo esamina per verificare dove è diretto, consulta la propria tabella e deposita il pacchetto nel buffer di uscita corrispondente.



# Tipi di servizio

Il tipo di servizio offerto dal livello di rete al livello di trasporto può essere:

- ❖ servizio con connessione e con riscontro (circuiti virtuali);
- ❖ servizio senza connessione e senza riscontro (servizio datagram), che è quello offerto dal protocollo IP.



# Circuito virtuale

Tutti i pacchetti sono instradati attraverso un circuito virtuale, cioè un percorso definito al momento della connessione, che rimane valido per tutta la durata della connessione stessa.

Per definire il percorso si utilizzano le tabelle di routing.

Tutti i router lungo il circuito ricordano, in una apposita struttura dati (tabella dei circuiti virtuali), la parte di loro competenza del percorso (e cioè quale linea in entrata e quale in uscita sono assegnate al circuito).

# Circuito virtuale

L'inoltro viene fatto basandosi sulle tabelle dei circuiti virtuali, quando la connessione viene chiusa il circuito scompare definitivamente.

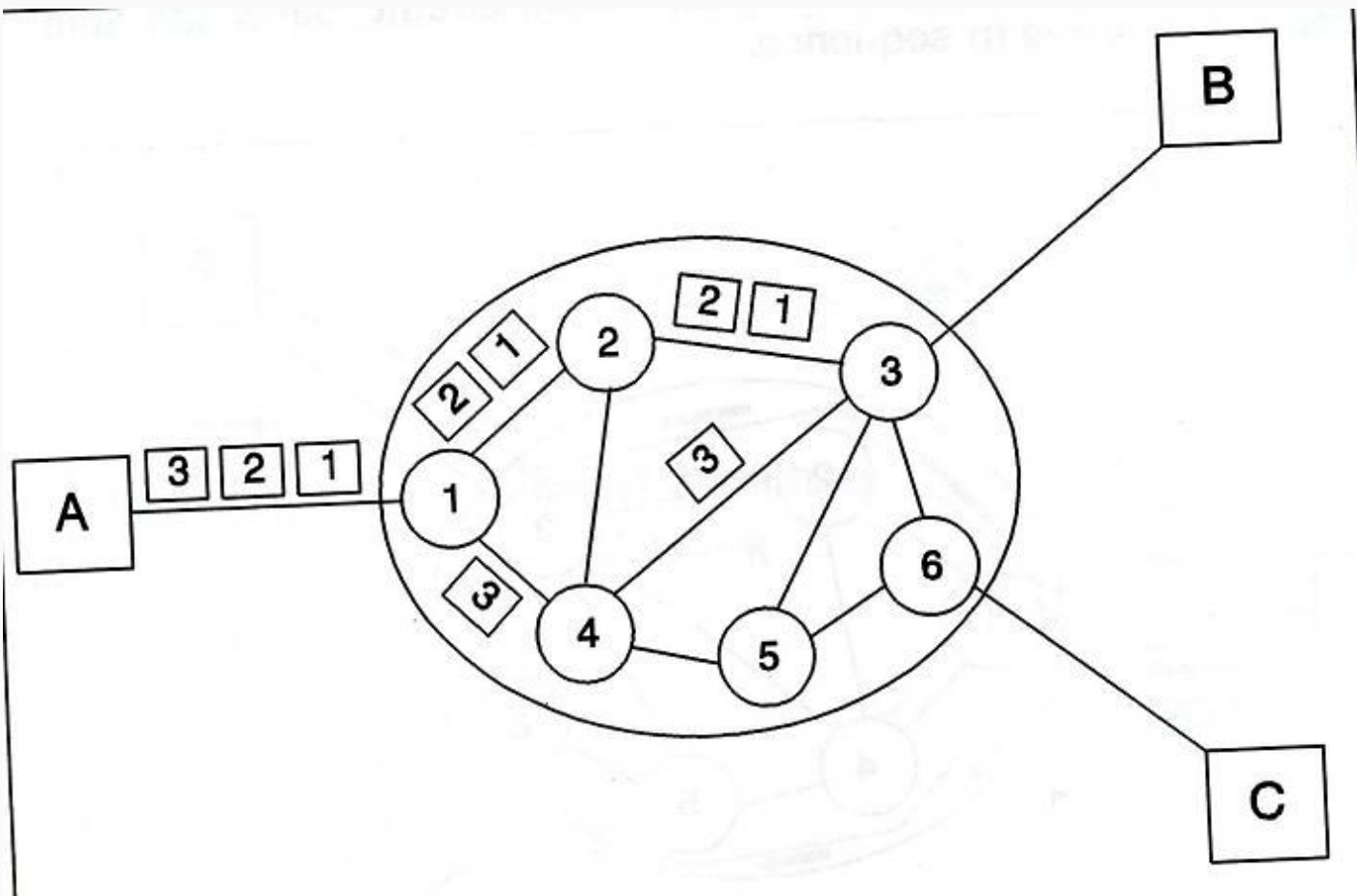
Una rete a circuito virtuale consegna a destinazione un flusso ordinato e affidabile (il trasmettitore riceve l'ack di ogni pacchetto inviato) di pacchetti.

Se un nodo cade, cade tutta la comunicazione e si possono perdere dei pacchetti.

# Circuito virtuale

Al momento della connessione si ha una fase di negoziazione in cui si stabiliscono alcune caratteristiche della connessione stessa, ad esempio la velocità massima di trasferimento dei dati.

Questo rende possibile implementare facilmente delle politiche di controllo della congestione.

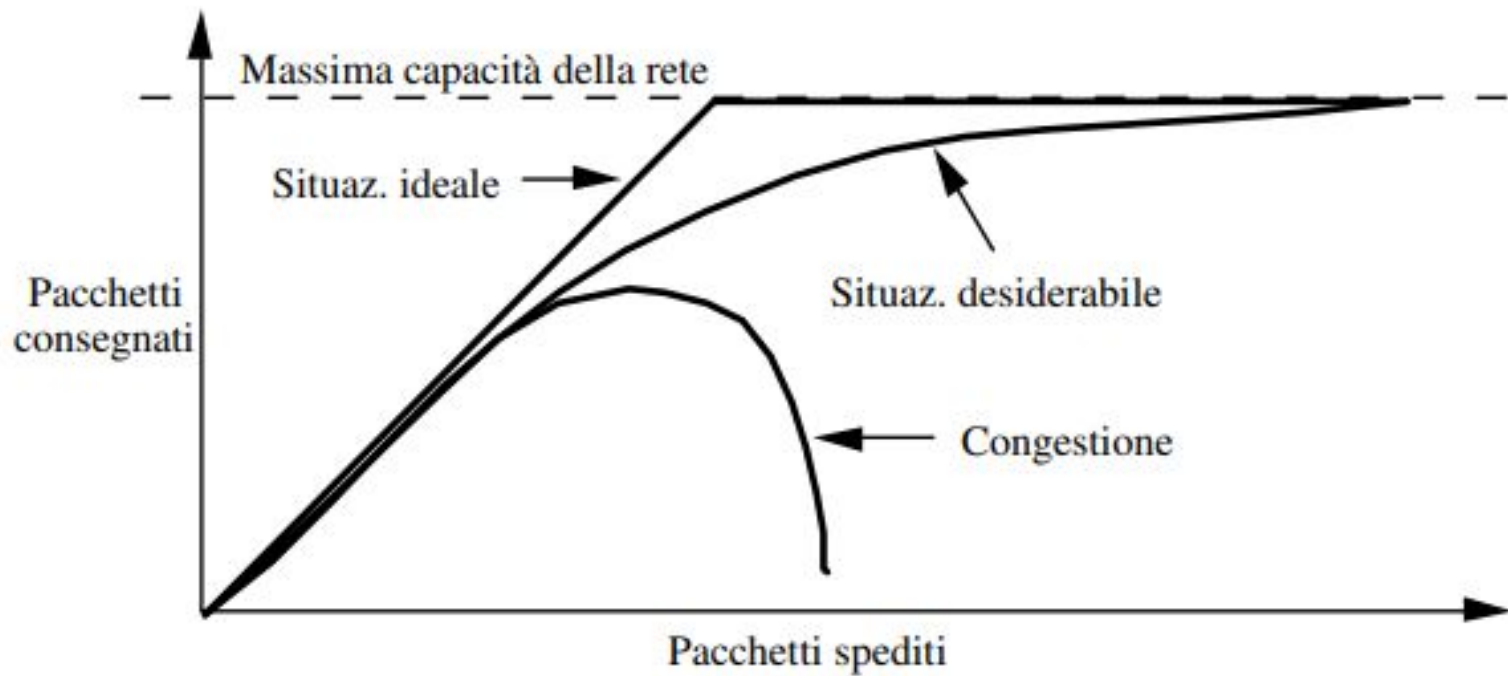


# Servizio datagram

I pacchetti, detti datagram, seguono percorsi diversi non stabiliti a priori. Ogni datagram deve contenere l'indirizzo completo del destinatario.

I router attraverso i quali passa il pacchetto determinano di volta in volta come inoltrarlo basandosi sulle tabelle di routing.

È possibile che i pacchetti non arrivino nell'ordine giusto, ma se un router intermedio cade, il pacchetto può seguire un percorso alternativo.





# Congestione di rete

Ogni pacchetto rimane all'interno router per un certo intervallo di tempo, "ospitato" all'interno di un buffer.

Se una linea di uscita di un router è troppo lenta per smaltire il traffico in quella direzione, il buffer di uscita non si svuota abbastanza rapidamente e la relativa coda si riempie.

# Congestione di rete

Se la situazione dura a lungo, il router si trova nell'impossibilità di inoltrare pacchetti verso quella linea di uscita.

I pacchetti devono rimanere memorizzati nelle code di ingresso, che si riempiono.

Quando le code di ingresso sono piene, il router inizia a scartare i pacchetti in arrivo, che vanno persi.

# Congestione di rete

I pacchetti persi vengono ritrasmessi (a livello 3 o 4, a seconda del tipo di servizio) e questo contribuisce ad aumentare il traffico di rete, peggiorando la situazione.

Questo fenomeno prende il nome di congestione, e provoca rapidamente il collasso delle prestazioni della rete.

# Algoritmi di routing

La funzione principale del livello rete è l'instradamento dei pacchetti, tipicamente facendo fare loro molti salti (hop) da un router ad un altro.

Recentemente si è iniziato a separare i due concetti di routing e forwarding.

Il routing (instradamento) si occupa di capire quali sono le destinazioni raggiungibili e lungo quali percorsi.

# Algoritmi di routing

Un algoritmo di routing, oltre ad essere in grado di recapitare i dati a destinazione, deve determinare il percorso migliore rispetto a una metrica di riferimento.

Possibili metriche sono:

- minimizzazione del tempo medio di recapito di un pacchetto;
- percorso più breve in termini di router attraversati;
- percorso a banda maggiore;
- ....

# Algoritmi di routing

Il risultato finale del processo di routing è la creazione di una serie di informazioni locali (ad esempio la tabella di routing in tecnologia IP) in ognuno dei nodi della rete che specificano la direzione per ogni possibile destinazione.

Il forwarding (inoltro) avviene in modo indipendente su ciascun nodo utilizzando le informazioni prodotte dal processo di routing (es. le tabelle di routing IP) e consiste appunto nell'inoltro dei pacchetti provenienti da ciascuna linea di ingresso verso la linea di uscita corretta.

# Algoritmi di routing

Una caratteristica importante del processo di forwarding è la mancanza di conoscenza del percorso globale.

Le informazioni memorizzate in ogni nodo, infatti, specificano banalmente la direzione del prossimo passo ma non il percorso globale.

Ambedue i processi (routing e forwarding) sono necessari per l'operatività di una rete.

# Caratteristiche degli algoritmi di routing

Semplicità:

- implementazione “snella”, con minore possibilità di errori;
- utilizzo limitato di tempo e risorse di elaborazione (CPU, memoria).



# Caratteristiche degli algoritmi di routing

## Robustezza ed Adattabilità:

- l'algoritmo deve funzionare su una topologia di rete generica e senza porre alcun vincolo sulla stessa;
- l'algoritmo deve supportare cambiamenti di topologia senza interrompere il funzionamento della rete:
  - Fault Detection: l'algoritmo deve essere in grado di accorgersi della presenza di guasti isolandoli automaticamente;
  - Autostabilizzazione: la rete deve essere in grado di convergere ai nuovi percorsi di instradamento (se questi esistono) in un tempo finito senza alcun intervento esterno.

# Caratteristiche degli algoritmi di routing

Ottimalità nella scelta dei cammini, rispetto alle metriche prescelte.

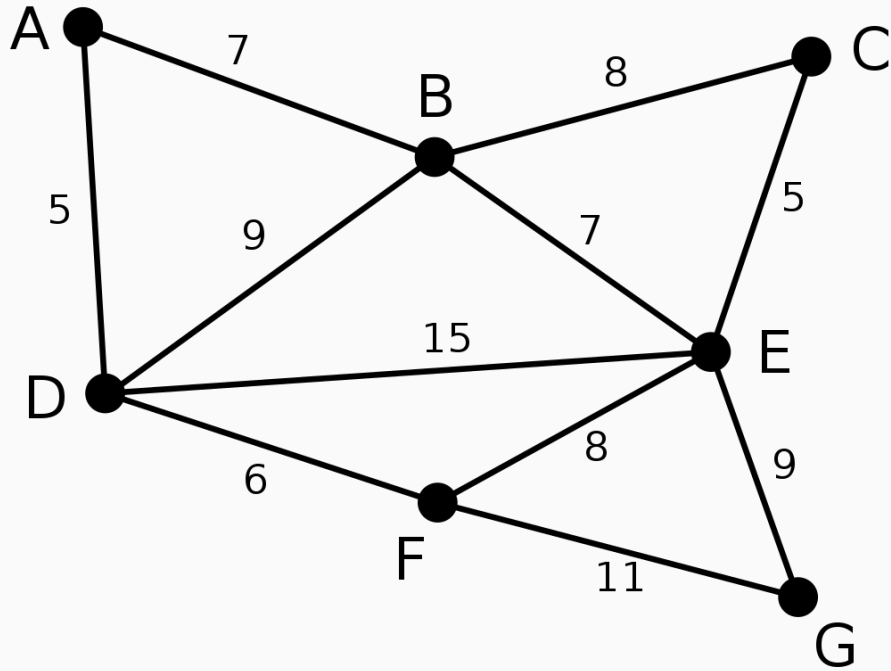
Stabilità: la rete deve raggiungere, a regime, uno stato stabile nel quale il routing non deve più cambiare a meno che si verifichino delle variazioni dei costi o della topologia.

Equità: non devono esistere nodi “trascurati” o “danneggiati” dalla scelta dei percorsi di instradamento.

# Routing basato sul cammino minimo: algoritmo di Dijkstra

Inizialmente tutti i nodi tranne quello di partenza sono possibili tentativi e vengono “etichettati” con costo infinito. A ogni iterazione:

- si sceglie il nodo K a costo minore tra quelli non ancora resi definitivi,
- si vede a quale costo X possono essere raggiunti a partire da K i vicini di K che non sono ancora definitivi; se X risulta inferiore al valore indicato nell’etichetta del nodo, il nodo viene marcato come raggiungibile da K a costo X
- K viene marcato come definitivo.



# Flooding

Ogni pacchetto viene inoltrato su tutte le linee eccetto quella da cui è arrivato.

In questo modo ogni pacchetto arriva a tutti i router, ma si genera un numero enorme (teoricamente infinito!) di pacchetti.

Non richiede di conoscere la topologia della rete, è utilizzato da altri algoritmi di routing per lo scambio di informazioni tra i router.

# Flooding

Ci sono delle tecniche per limitare il traffico generato:

- si inserisce in ogni pacchetto un contatore che viene decrementato ad ogni hop, quando il contatore arriva a zero, il pacchetto viene scartato;
- si inserisce la coppia (source router ID, sequence number) in ogni pacchetto, quando un router riceve per la seconda volta lo stesso pacchetto lo scarta.

# Distance vector routing

È un algoritmo di routing basato sul cammino minimo.

Il vettore delle distanze di un router è un vettore in cui il router registra la distanza (misurata o stimata) tra se stesso e tutte le possibili destinazioni.

L'algoritmo è molto semplice da implementare, purtroppo presenta un problema di convergenza molto lenta in caso di cadute di nodi/collegamenti (conteggio all'infinito).

Il primo protocollo di routing utilizzato in Internet (RIP) era basato su questo algoritmo.

# Distance vector routing

L'algoritmo opera in un ambiente distribuito:

- i router non devono conoscere la topologia della rete, conoscono solo il costo per raggiungere i vicini immediati;
- i router scambiano messaggi solo con i vicini immediati;
- tutti i router eseguono lo stesso algoritmo di parallelo, avendo tutti lo stesso ruolo;
- sia i router che i collegamenti possono cadere, i messaggi possono andare persi.



# Distance vector routing

- Ogni nodo  $i$  inizializza il proprio vettore delle distanze impostando a 0 la distanza da se stesso e a  $\infty$  la distanza dagli altri;
- periodicamente invia il vettore ai vicini immediati e aggiorna il proprio vettore di distanza in base ai vettori ricevuti dai vicini, scegliendo per ogni destinazione  $x$  la distanza  $\min(c(j)+d(j,x))$  essendo  $c(j)$  il costo (misurato) per andare da  $i$  a  $j$  e  $d(j,x)$  la distanza (misurata o stimata) che  $j$  dichiara da  $x$ ; per l'inoltro si sceglie il "miglior" vicino.

# Problema del conteggio all'infinito

Infatti, consideriamo questo esempio:

A   B   C   D   E   <- Router

\*-----\*-----\*-----\*-----\*   <- Collegamenti (topologia lineare)

1   2   3   4   <- Distanze da A

# Problema del conteggio all'infinito

Se ora cade la linea fra A e B, dopo uno scambio succede questo:

A   B   C   D   E <- Router

\*   \*\_\_\_\_\*\_\_\_\_\*\_\_\_\_\* <- Collegamenti

3   2   3   4 <- Distanze da A (dopo uno scambio)

Ciò perché B, non ricevendo risposta da A, crede di poterci arrivare via C, che ha distanza due da A.

# Problema del conteggio all'infinito

Col proseguire degli scambi, si ha la seguente evoluzione:

A   B   C   D   E   <- Router

\*   \*\_\_\_\_\*\_\_\_\_\*\_\_\_\_\*   <- Collegamenti

3   4   3   4   <- Distanze da A (dopo due scambi)

5   4   5   4   <- Distanze da A (dopo tre scambi)

5   6   5   6   <- Distanze da A (dopo quattro scambi)

# Problema del conteggio all'infinito

A lungo andare, tutti i router vedono lentamente aumentare sempre più la distanza per arrivare ad A. Questo problema viene chiamato “conteggio all'infinito”.

In questo lungo transitorio (che impedisce di trovare eventuali percorsi validi) i pacchetti vengono instradati in modo non corretto.

Anche se ci sono delle soluzioni per limitare l'effetto del conteggio all'infinito, questo protocollo non viene più usato.

# Link state routing

È un algoritmo più recente, che supera il problema della lentezza di convergenza del distance vector routing.

Ogni router tiene sott'occhio lo stato dei collegamenti fra sé e i suoi vicini immediati (misurando il ritardo di ogni linea) e distribuisce tali informazioni a tutti gli altri.

Sulla base delle informazioni ricevute, ogni router ricostruisce localmente la topologia completa dell'intera rete e calcola il cammino minimo fra sé e tutti gli altri.

# Link state routing

I passi da seguire sono:

- scoprire i vicini e identificarli;
- misurare il costo (ritardo o altro) delle relative linee;
- costruire un pacchetto con tali informazioni;
- mandare il pacchetto a tutti gli altri router;
- previa ricezione degli analoghi pacchetti che arrivano dagli altri router, costruire la topologia dell'intera rete;
- calcolare il cammino più breve a tutti gli altri router.

# Link state routing

Quando il router si avvia, invia un pacchetto HELLO su tutte le linee in uscita. In risposta riceve dai vicini i loro indirizzi (univoci in tutta la rete).

Periodicamente i router inviano vari pacchetti ECHO ai vicini immediati, misurando il tempo di arrivo della risposta (diviso 2) e mediando su vari pacchetti si deriva il ritardo della linea.

Si costruisce un pacchetto (chiamato pacchetto di link state) con: identità del mittente, numero di sequenza del pacchetto, età del pacchetto, lista dei vicini con i relativi ritardi.



# Link state routing

La distribuzione dei pacchetti è la parte più delicata, perché errori in questa fase possono portare qualche router ad avere idee sbagliate sulla topologia, con conseguenti malfunzionamenti.

Di base si usa il flooding, inserendo nei pacchetti le coppie (source router ID, sequence number) per eliminare i duplicati. Tutti i pacchetti sono confermati. Inoltre, per evitare che pacchetti vaganti (per qualche errore) girino per sempre, l'età del pacchetto viene decrementata ogni secondo, e quando arriva a zero il pacchetto viene scartato.

# Link state routing

Combinando tutte le informazioni arrivate, ogni router costruisce il grafo della subnet e calcola il cammino minimo a tutti gli altri router.

Il link state routing è usato in Internet: il protocollo standard OSPF (Open Shortest Path First) è basato su tale principio e si avvia ad essere l'algoritmo più utilizzato.

# Routing gerarchico

Quando la rete cresce fino contenere decine di migliaia di nodi, la tabelle di instradamento diventavano di complessità insostenibile.

Il routing va quindi impostato in modo gerarchico. La rete viene divisa in zone (spesso dette regioni) e ci sono due livelli di routing:

- routing all'interno di ogni regione;
- routing fra tutti i router di confine, che sono quelli che collegano le regioni tra loro.

# Routing gerarchico

I router interni di una regione sanno come arrivare a tutti gli altri router della regione. Quando un router interno deve spedire qualcosa a un router di un'altra regione sa soltanto che deve farlo pervenire al router di confine della propria regione.

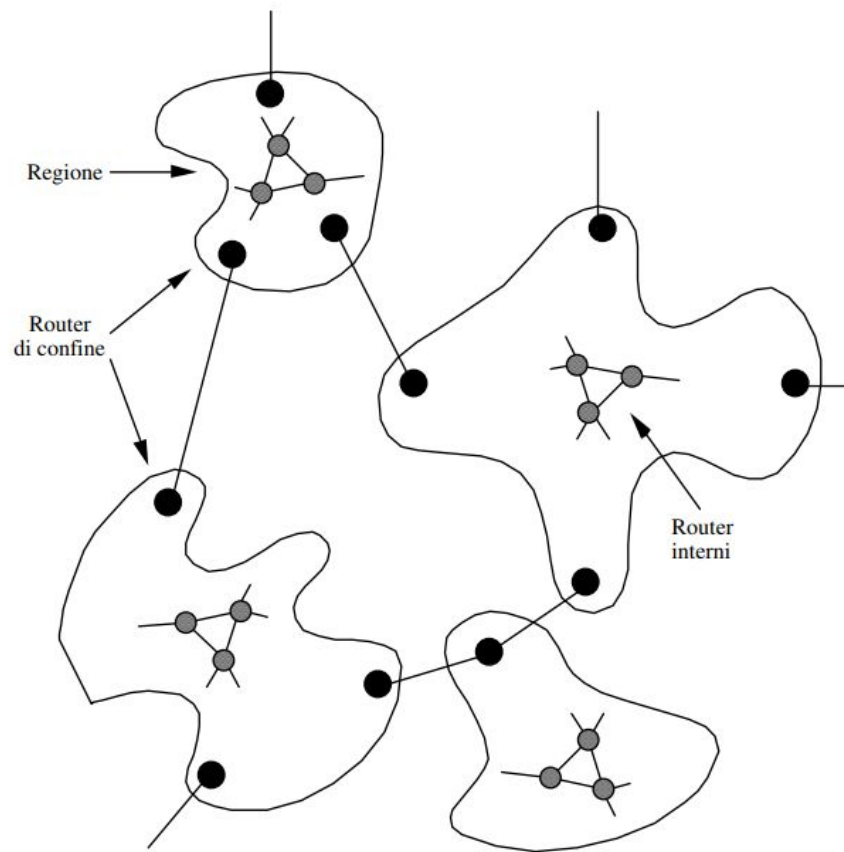
I router interni quindi devono avere nelle loro tabelle di routing una riga per ogni altro router interno (con la relativa linea da usare per arrivarci) e una riga per ogni altra regione (con l'indicazione della linea di uscita da utilizzare per arrivare al relativo router di confine).

# Routing gerarchico

I router di confine sanno a quale altro router di confine deve inviare i dati perché arrivino alla regione di destinazione.

I router di confine quindi mantengono nelle proprie tabelle, oltre alle informazioni necessarie all'instradamento interno, anche una riga ogni altra regione, con l'indicazione del prossimo router di confine da contattare e della linea da usare per arrivarci.

Se due livelli di routing non sono sufficienti, è possibile aggiungerne un terzo.

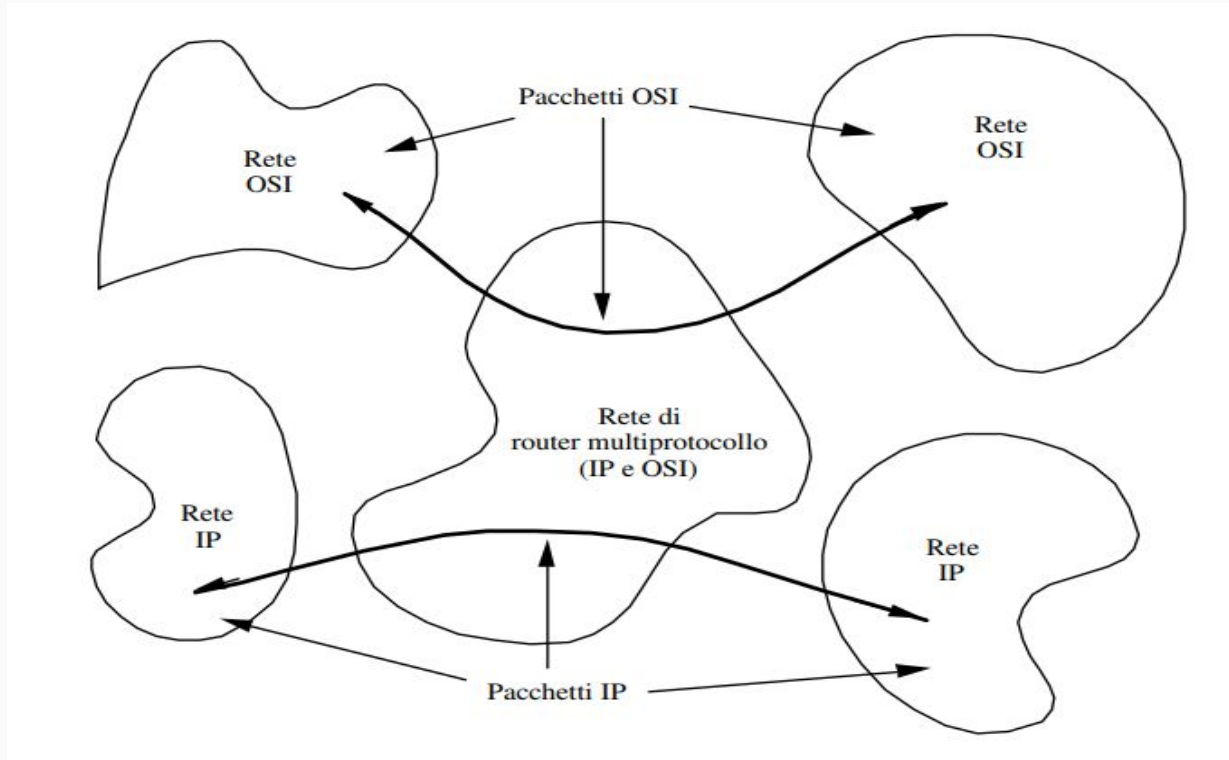


# Internetworking

Per connettere fra loro reti eterogenee si devono superare problemi non banali, tra i quali:

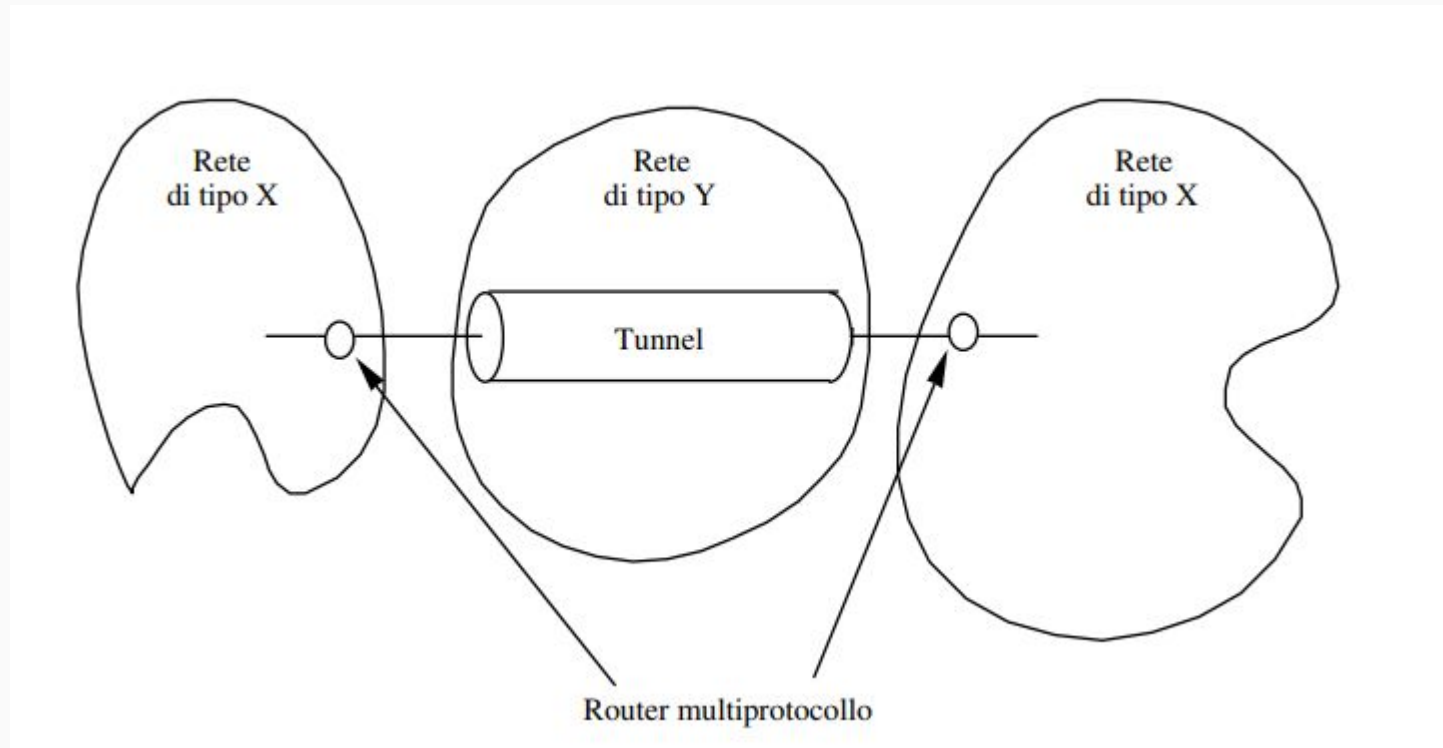
- difformità nei servizi offerti (circuiti virtuale o datagram);
- difformità nei formati dei pacchetti e degli indirizzi;
- difformità dei meccanismi di controllo dell'errore e della congestione;
- difformità nella dimensione massima dei pacchetti;
- .....

# Reti multiprotocollo

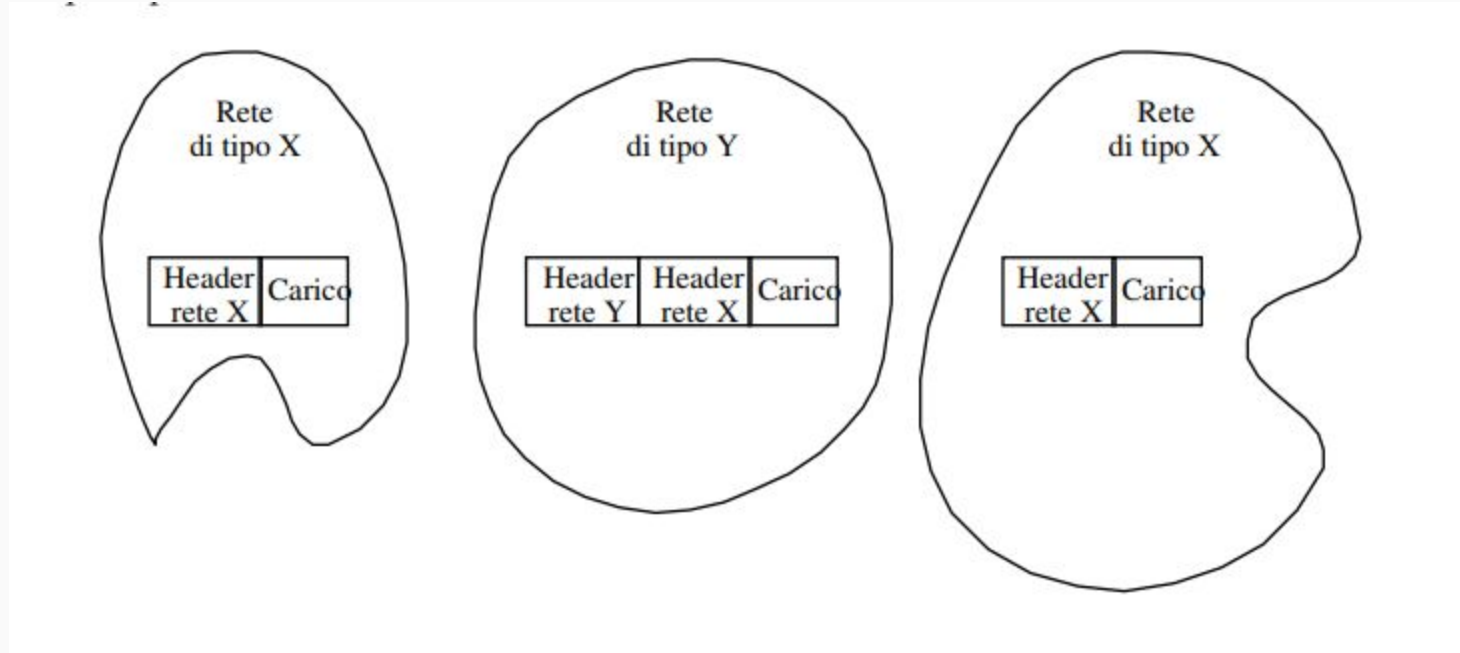




# Tunneling



# Tunneling



# Tunneling

Il tunneling è una tecnica utilizzata per connettere due reti di tipo X attraversando una rete di tipo Y (esempio tipico, interconnettere due reti IPv6 utilizzando una rete IPv4 o viceversa).

Alle due estremità del tunnel ci sono due router multiprotocollo.

I pacchetti di tipo X in ingresso al tunnel vengono incapsulati in pacchetti di tipo Y, raggiungono l'estremità del tunnel, l'intestazione di tipo Y viene rimossa e i pacchetti immessi nella rete di destinazione.