

Il routing

(Modulo 11, Approfondimento)

Maurizio Gabbrielli
Universita' di Bologna

Sommario:

- | Cosa e' il routing
- | Il routing nelle reti a commutazione di pacchetto
- | Un algoritmo di routing

Buon giorno, il mio nome è Maurizio Gabbrielli, sono dell'Università di Bologna ed in questa lezione parleremo del routing, quindi vedremo un approfondimento del modulo 11, nel contesto del quale sono state già viste altre caratteristiche delle reti di calcolatori e di Internet in particolare. I punti che affronteremo, come è esposto qui nel sommario sono: cosa si intende per routing, come avviene il routing nelle reti a commutazione di pacchetto, infine vedremo un algoritmo di routing specifico, che è quello di Dijkstra, usato in alcuni contesti.

Il routing nelle reti a commutazione di pacchetto 1

Il Routing nelle reti a commutazione di pacchetto

- 1 **Reti a commutazione di pacchetto:** messaggi suddivisi in pacchetti che sono spediti (a livello rete) da un mittente ad un destinatario
 - mittente e destinatario identificati in Internet da indirizzi IP.
- 1 **Router:** calcolatore specializzato con molti link la cui funzione è quella di instradare (o commutare) i pacchetti da link di ingresso a link di uscita
 - Sui router girano solo i protocolli dei livelli fisico, link e rete
- 1 **Routing:** consiste nello spostare i pacchetti attraverso i nodi intermedi (router) per farli transitare dalla sorgente alla destinazione.
- 1 **Due problemi principali:**
 - come si sceglie il percorso di un pacchetto
 - come si realizza la commutazione dal link di ingresso a quello di uscita
- soluzioni diverse per**
 - reti con circuiti virtuali
 - reti datagram

Iniziamo ricordando cosa si intende per reti a commutazione di pacchetto. In tale tipologia di rete i messaggi sono appunto suddivisi in pacchetti, tutti di una certa dimensione prefissata, che vengono spediti a livello di rete dal mittente al destinatario, identificati entrambi da un IP-address (indirizzo IP), cioè un indirizzo che consiste di 32 bit divisi in quattro gruppi da 8 bit ciascuno. Quindi, rispetto alle reti a commutazione di circuito, la differenza sostanziale è la mancanza di un uso esclusivo delle risorse offerte dalle reti, nel senso che le risorse di rete sono condivise fra più comunicazioni diverse, che possono quindi spedire i pacchetti usando gli stessi canali, gli stessi router, con un meccanismo di condivisione delle risorse non presente nelle reti a commutazione di circuito, dove invece una volta che sia stata stabilita la comunicazione, la risorsa per quel canale di quel circuito è assegnata in uso esclusivo ai due partner della comunicazione e viene mantenuta fino a che la comunicazione stessa non cessi. Nelle reti a commutazione di pacchetto tutto questo non avviene, se si spediscono vari pacchetti che devono transitare sulla rete, esistono i cosiddetti router, cioè dei calcolatori specializzati la cui funzione è quella di instradare, commutare i pacchetti.

Il routing nelle reti a commutazione di pacchetto 2

Il Routing nelle reti a commutazione di pacchetto

- 1 **Reti a commutazione di pacchetto:** messaggi suddivisi in pacchetti che sono spediti (a livello rete) da un mittente ad un destinatario
 - mittente e destinatario identificati in Internet da indirizzi IP.
- 1 **Router:** calcolatore specializzato con molti link la cui funzione è quella di instradare (o commutare) i pacchetti da link di ingresso a link di uscita
 - Sui router girano solo i protocolli dei livelli fisico, link e rete
- 1 **Routing:** consiste nello spostare i pacchetti attraverso i nodi intermedi (router) per farli transitare dalla sorgente alla destinazione.
- 1 **Due problemi principali:**
 - come si sceglie il percorso di un pacchetto
 - come si realizza la commutazione dal link di ingresso a quello di uscita
- soluzioni diverse per**
 - reti con circuiti virtuali
 - reti datagram

Questi router avranno appunto un certo numero di link in entrata e un certo numero di link in uscita e lo scopo del loro funzionamento è quello di prelevare un pacchetto da un link di ingresso e di instradarlo su un opportuno link di uscita. Va ricordato il fatto che sui router sono implementati soltanto i primi tre livelli a partire dal basso della gerarchia ISO-OSI per quanto riguarda le reti, cioè i protocolli del livello fisico, del livello link e del livello di rete, mentre non sono implementati i protocolli del livello di trasporto e di applicazione. Il routing consiste nell'attività specifica di un router, cioè nell'instradare pacchetti, in modo tale da farli arrivare dalla sorgente, cioè dal mittente, alla destinazione, facendoli transitare attraverso i nodi intermedi che costituiscono la rete. Nella realizzazione di questo meccanismo di instradamento ci sono due problemi principali che devono essere affrontati: il primo è la scelta del percorso del pacchetto, cioè come decidere dinamicamente il cammino che il pacchetto deve seguire per arrivare dalla sorgente alla destinazione; il secondo problema è come realizzare la commutazione dal link di ingresso a quello di uscita, cioè come far transitare un pacchetto in uno specifico router da un link di ingresso ad un link di uscita. Vedremo che ci sono soluzioni diverse a seconda del tipo di reti a commutazione di pacchetto considerato, cioè reti con circuiti virtuali oppure reti di tipo datagram.

Il routing nelle reti a commutazione di pacchetto con circuiti virtuali 1

Il Routing nelle reti a commutazione di pacchetto con circuiti virtuali

- | Il cammino dei pacchetti di un messaggio e' determinato al momento della **inizializzazione della chiamata** e rimane fisso durante tutta la call
- | ogni pacchetto porta una **etichetta** che identifica il circuito virtuale e che determina il prossimo salto
- | numeri diversi per i vari link del percorso: **tabelle di conversione** in ogni router convertono i numeri di circuito virtuale in numeri di link
- | routers mantengono lo **stato** di una una call
- | usati nelle reti **ATM** ma non in Internet

Nelle reti a commutazione di pacchetto con circuiti virtuali la situazione è la seguente: prima che venga iniziato il trasferimento effettivo dei dati, cioè la spedizione dei messaggi tramite pacchetti, c'è una fase di inizializzazione della chiamata, nella quale appunto i vari router che si trovano nel cammino tra la sorgente e il destinatario, si "mettono d'accordo" e stabiliscono una sorta di circuito virtuale, che sarà poi usato dal mittente e dal destinatario per spedire i pacchetti. Quindi in qualche modo queste reti ricordano un po' le reti a commutazione di circuito, la differenza sta nel fatto che, come è stato detto in precedenza, l'uso delle risorse non è esclusivo, cioè anche se viene stabilito un circuito virtuale tra mittente e destinatario, questo non vuol dire che essi abbiano in uso esclusivo le risorse che compongono questo circuito virtuale.

Il routing nelle reti a commutazione di pacchetto con circuiti virtuali 2

Il Routing nelle reti a commutazione di pacchetto con circuiti virtuali

- 1 Il cammino dei pacchetti di un messaggio e' determinato al momento della **inizializzazione della chiamata** e rimane fisso durante tutta la call
- 1 ogni pacchetto porta una **etichetta** che identifica il circuito virtuale e che determina il prossimo salto
- 1 numeri diversi per i vari link del percorso: **tabelle di conversione** in ogni router convertono i numeri di circuito virtuale in numeri di link
- 1 routers mantengono lo **stato** di una una call
- 1 usati nelle reti **ATM** ma non in Internet

Tornando alla rete a commutazione di pacchetto con circuiti virtuali dopo questa prima fase di inizializzazione della chiamata, viene quindi stabilito il circuito virtuale, cioè il percorso che devono seguire i pacchetti, il quale è identificato a livello di pacchetti tramite un'etichetta contenuta in ogni pacchetto. Nella realtà la situazione è un po' più complicata, nel senso che questi codici che identificano un circuito virtuale non sono gli stessi su tutti i pezzi del cammino che costituiscono l'intero circuito virtuale, cioè in punti diversi del circuito, esso viene identificato da numeri diversi e per tale motivo servono le tabelle di conversione di router, che associano al numero di circuito virtuale in arrivo il numero di circuito virtuale in uscita. A livello concettuale possiamo rimanere con l'idea che ogni circuito virtuale venga identificato da una certa etichetta inserita all'interno del pacchetto per indicare quale circuito virtuale deve essere seguito.

Il routing nelle reti a commutazione di pacchetto con circuiti virtuali 3

Il Routing nelle reti a commutazione di pacchetto con circuiti virtuali

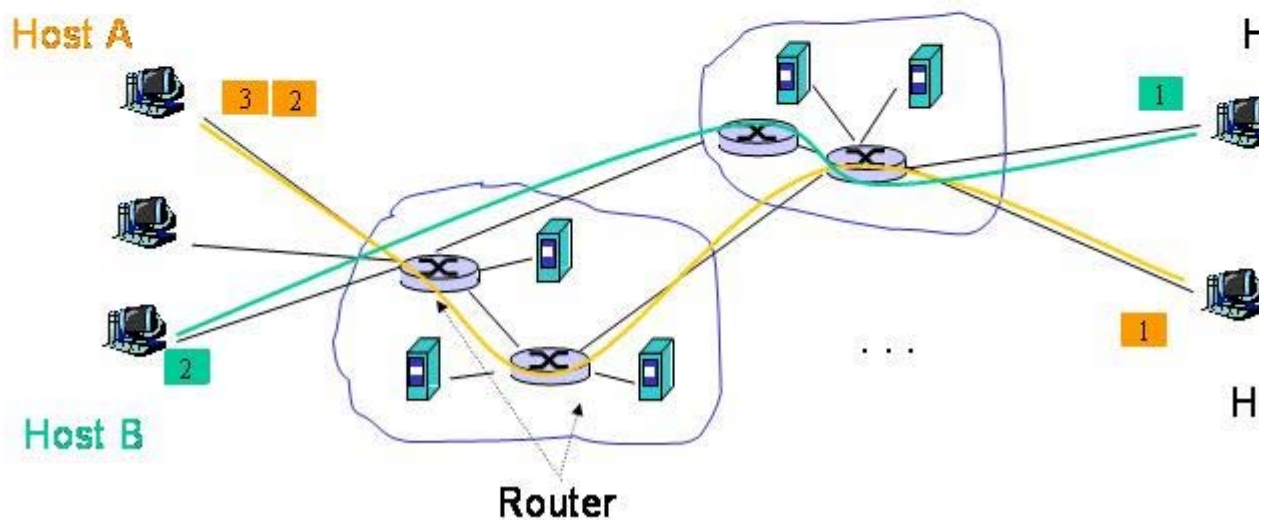
- | Il cammino dei pacchetti di un messaggio e' determinato al momento della **inizializzazione della chiamata** e rimane fisso durante tutta la call
- | ogni pacchetto porta una **etichetta** che identifica il circuito virtuale e che determina il prossimo salto
- | numeri diversi per i vari link del percorso: **tabelle di conversione** in ogni router convertono i numeri di circuito virtuale in numeri di link
- | routers mantengono lo **stato** di una una call
- | usati nelle reti **ATM** ma non in Internet

Da quanto detto appare abbastanza chiaro il fatto che i router mantengano lo stato di una chiamata, cioè sanno quanti e quali sono i circuiti virtuali che li attraversano, quindi ad ogni istante hanno un controllo sulla situazione della trasmissione dei dati per quanto riguarda i circuiti virtuali che attraversano quello specifico router. Le reti a commutazione di pacchetto con circuiti virtuali sono usate nelle reti ATM, che sono reti di origine telefonica, non molto usate attualmente per la trasmissione di dati; tali circuiti virtuali non sono invece usati in Internet, dove invece sono preferite le reti di tipo datagram come vedremo in seguito.

Reti a commutazione di pacchetto con circuiti virtuali 1

Reti a commutazione di pacchetto con circuiti virtuali

- | **Messaggi suddivisi in pacchetti**
- | **Pacchetti dello stesso messaggio seguono lo stesso percorso**
- | **Pacchetti di messaggi diversi possono condividere gli stessi link**

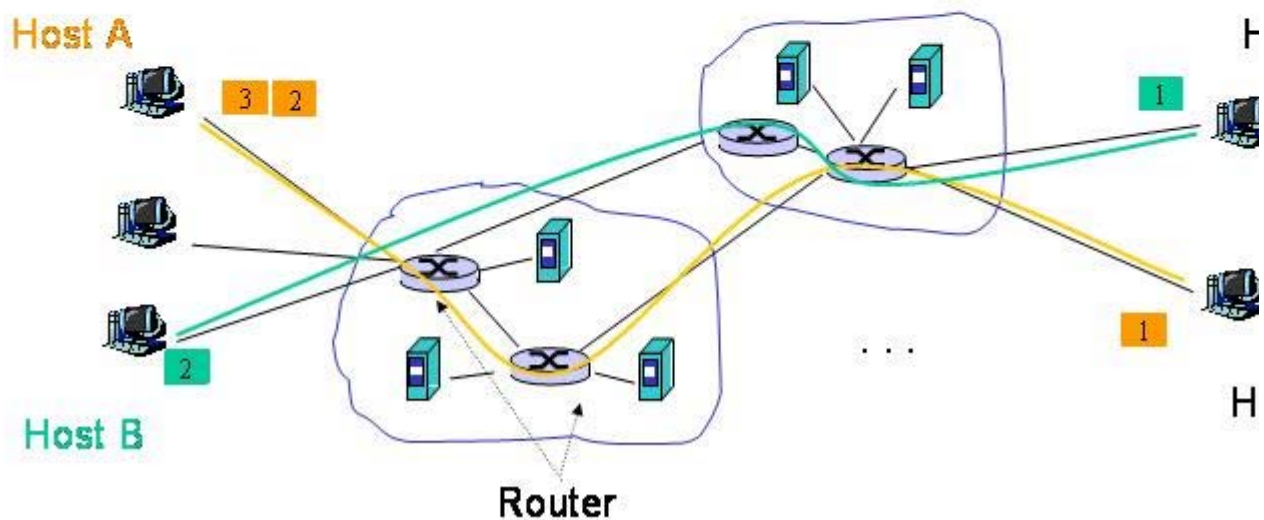


Per ricapitolare quanto detto sui circuiti virtuali, possiamo vedere questa visualizzazione di tipo grafico: abbiamo questa prima situazione con due host A e B che vogliono inviare dei pacchetti a degli altri host che si troveranno dall'altra parte della rete, anche a distanze geografiche notevoli. Tra i mittenti e i destinatari esiste tutta la struttura della rete costituita dai vari router. In questa rete a commutazione di pacchetto con circuiti virtuali se l'host A vuole inviare dei dati a questo altro host che si trova nella parte destra in basso, prima di tutto stabilisce un circuito virtuale, rappresentato in figura da una linea arancione, dopodiché i pacchetti vengono spediti lungo questo circuito virtuale, quindi i pacchetti 1, 2 e 3 dell'host A usano esclusivamente il cammino che è stato definito al momento dell'inizializzazione del circuito virtuale.

Reti a commutazione di pacchetto con circuiti virtuali 2

Reti a commutazione di pacchetto con circuiti virtuali

- | **Messaggi suddivisi in pacchetti**
- | **Pacchetti dello stesso messaggio seguono lo stesso percorso**
- | **Pacchetti di messaggi diversi possono condividere gli stessi link**



Analogamente l'host B che vorrà inviare dei pacchetti ad un altro host, nella figura lo troviamo a destra in alto, prima di inviare i pacchetti stabilirà un circuito virtuale, dopodiché i pacchetti transitano dal mittente al destinatario, verrà inviato prima il pacchetto 1 dell'host A, poi il pacchetto 1 dell'host B e così via, con un ordine casuale nell'alternarsi tra pacchetti dell'host A e dell'host B. È quindi importante ricordare che i messaggi sono suddivisi in pacchetti, se questi ultimi appartengono allo stesso messaggio, in queste reti a circuiti virtuali seguiranno lo stesso percorso e infine pacchetti di messaggi diversi possono condividere gli stessi collegamenti, dato che le risorse sono condivise tra le varie comunicazioni.

Il routing nelle reti datagram a commutazione di pacchetto 1

Il Routing nelle reti datagram a commutazione di pacchet

- 1 E' il routing usato in **Internet**
- 1 **Non c'e' inizializzazione** della chiamata
- 1 **Indirizzo di destinazione** (indirizzo IP in Internet) contenuto in ogni pacchetto determina il prossimo salto
- 1 Il percorso puo' cambiare durante la sessione: pacchetti diversi dello stesso messaggio possono seguire strade diverse
- 1 I router non mantengono alcuno stato delle trasmissioni: per ogni pacchetto controllano opportune **tabelle** nelle quali vengono memorizzate le corrispondenze:
 - indirizzo IP -- link di uscita**
- 1 Le tabelle sono create e aggiornate da opportuni algoritmi di routing

Per quanto riguarda invece il routing nelle reti a commutazione di pacchetto di tipo datagram, che è poi quello usato in Internet, diciamo innanzitutto che non esiste alcuna inizializzazione della chiamata, cioè non viene stabilito un circuito virtuale come abbiamo visto in precedenza. In questo caso infatti ogni pacchetto che viene spedito contiene l'indirizzo di destinazione, sottoforma di indirizzo IP. Quindi ogni destinatario spedisce un pacchetto al prossimo router che si trova sulla rete e il percorso che il pacchetto è fatto in passi successivi, nel senso che il pacchetto viene spedito dal mittente al router più vicino, questo ultimo analizza l'indirizzo di destinazione del pacchetto e in base a questo indirizzo decide a quale altro router successivo inviare il pacchetto. Nel router successivo si procede allo stesso modo, cioè viene analizzato nuovamente l'indirizzo IP di destinazione e deciso il nodo seguente e così via fino a che il pacchetto non arriva a destinazione. Parlando informalmente tale processo è analogo al tragitto compiuto da un'automobile per andare da una città ad un'altra e il conducente si ferma durante il viaggio a chiedere informazioni sulla strada da seguire.

Il routing nelle reti datagram a commutazione di pacchetto 2

Il Routing nelle reti datagram a commutazione di pacchet

- 1 E' il routing usato in **Internet**
- 1 **Non c'e' inizializzazione** della chiamata
- 1 **Indirizzo di destinazione** (indirizzo IP in Internet) contenuto in ogni pacchetto determina il prossimo salto
- 1 Il percorso puo' cambiare durante la sessione: pacchetti diversi dello stesso messaggio possono seguire strade diverse
- 1 I router non mantengono alcuno stato delle trasmissioni: per ogni pacchetto controllano opportune **tabelle** nelle quali vengono memorizzate le corrispondenze:
 - indirizzo IP -- link di uscita**
- 1 Le tabelle sono create e aggiornate da opportuni algoritmi di routing

Quindi da quanto detto appare chiaro che i pacchetti che sono spediti ad un destinatario e che costituiscono lo stesso messaggio in generale possono non seguire tutti la stessa strada, lo stesso cammino per arrivare a destinazione, cioè possono seguire cammini diversi a seconda dello stato della rete, quindi in generale il percorso può cambiare durante la sessione di trasmissione dei dati. Inoltre, sempre per quanto abbiamo detto sul meccanismo di trasmissione di pacchetti con datagram, appare evidente che i router non mantengono sulle trasmissioni in corso, nel senso che uno specifico router non sa quante e quali sono le trasmissioni che in quel momento lo stanno usando. Il router vede soltanto un insieme di pacchetti che gli arriva nei link di ingresso e che lui deve in qualche modo smistare, instradare nei link di uscita, senza sapere se questi pacchetti appartengono o meno alla stessa trasmissione. Il router, per poter instradare i pacchetti mantiene delle opportune tabelle nelle quali vengono memorizzate delle corrispondenze tra indirizzo IP, inteso come indirizzo di destinazione del pacchetto e link di uscita; quindi il lavoro del router consiste per ogni pacchetto che gli arriva in ingresso, controllare l'indirizzo IP di tale pacchetto, andare nella tabella che mantiene queste corrispondenze, vedere quale link di uscita corrisponde per quello specifico indirizzo IP e instradare il pacchetto su quello specifico link di uscita.

Il routing nelle reti datagram a commutazione di pacchetto 3

Il Routing nelle reti datagram a commutazione di pacchetto

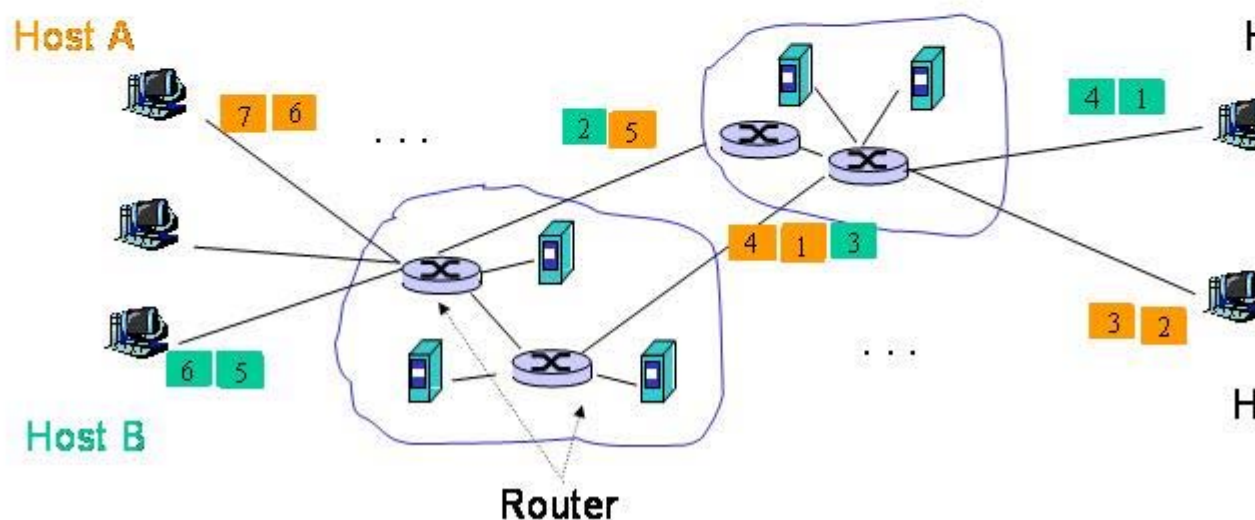
- 1 E' il routing usato in **Internet**
- 1 **Non c'e' inizializzazione** della chiamata
- 1 **Indirizzo di destinazione** (indirizzo IP in Internet) contenuto in ogni pacchetto determina il prossimo salto
- 1 Il percorso puo' cambiare durante la sessione: pacchetti diversi dello stesso messaggio possono seguire strade diverse
- 1 I router non mantengono alcuno stato delle trasmissioni: per ogni pacchetto controllano opportune **tabelle** nelle quali vengono memorizzate le corrispondenze:
 - indirizzo IP -- link di uscita**
- 1 Le tabelle sono create e aggiornate da opportuni algoritmi di routing

Quindi il problema essenziale che si pone alla realizzazione di router nelle reti a commutazione di pacchetto di tipo datagram è la gestione di queste tabelle, sia nel senso di come realizzare l'accesso alle tabelle in modo efficiente, sia soprattutto decidere come creare e aggiornare queste tabelle, cioè decidere come definire i cammini tra la sorgente e la destinazione e quindi instradare i pacchetti nella rete. Questo avviene usando degli opportuni algoritmi di routing dei quali ne vedremo in seguito uno in particolare che è quello di Dijkstra.

Reti datagram a commutazione di pacchetto

Reti datagram a commutazione di pacchetto

- | I messaggi sono suddivisi in pacchetti
- | Pacchetti dello stesso messaggio possono seguire percorsi diversi
- | Pacchetti di messaggi diversi possono condividere gli stessi link



Anche qui abbiamo una figura che riassume quello che abbiamo detto finora relativamente alle reti a commutazione di pacchetto di tipo datagram. Ciò che vedete è una situazione analoga a prima nella quale abbiamo due host A e B che vogliono spedire dei dati a due host che si trovano nella parte destra della figura. Come nel caso precedente i dati sono suddivisi in pacchetti, però ora i pacchetti che costituiscono un singolo messaggio possono seguire percorsi diversi. Quindi vedete ad esempio che i pacchetti 2 e 3 spediti dall'host A sono già arrivati, i pacchetti 1 e 4 seguono invece quest'altra strada e passano attraverso questo collegamento da questo router all'altro in alto, mentre il pacchetto 5 sempre dell'host A segue un percorso diverso. Vedete, come è esemplificato nella figura, i pacchetti 2 e 3 arrivano prima del pacchetto 1, questo vuol dire che in generale non c'è alcuna garanzia del fatto che venga mantenuto correttamente l'ordine di spedizione dei pacchetti, vale a dire che io spedisco i pacchetti nell'ordine 1, 2, 3, 4 quello che accade è che arrivano prima i pacchetti 2, 3 poi l'1 e infine il 4. Più in particolare, non c'è alcuna garanzia che i pacchetti arrivino effettivamente a destinazione, nel senso che i pacchetti si possono perdere nella rete. Se vogliamo quindi essere sicuri che i pacchetti arrivino effettivamente a destinazione nell'ordine corretto, dobbiamo implementare degli opportuni meccanismi e questo viene fatto in genere a livello superiore, a livello di trasporto, ad esempio dal protocollo TCP che garantisce la trasmissione affidabile dei dati. L'ultima cosa da notare in questa figura è il fatto che pacchetti originati da messaggi diversi, di colore arancione e verde in figura, possono condividere gli stessi collegamenti.

Protocollo di routing 1

Protocollo di routing

- | **Protocollo di routing:** serve per aggiornare le tabelle dei router e quindi determinare il cammino dei pacchetti nella rete

- | **Cammino = sequenza di router nella rete**

- | **Usiamo una rappresentazione astratta della rete in termini di un grafo:**
 - nodi del grafo = router
 - archi del grafo = link fisici
 - varie nozioni di costi associati ai link: ritardo, livello di congestione, costo della trasmissione ...

- | **Algoritmo di routing:** e' alla base di ogni protocollo di routing e serve per determinare il cammino migliore secondo una qualche nozione di costo

Parlando in generale del routing, abbiamo capito a questo punto che servono opportuni protocolli di routing, che i router usano per scambiarsi informazioni in modo tale da poter aggiornare le tabelle delle quali abbiamo parlato prima, in modo tale da poter determinare il cammino che i pacchetti devono seguire nella rete, dove per cammino intendiamo la sequenza di router che devono essere incontrati nel percorso dalla sorgente alla destinazione. Possiamo dal punto di vista concettuale identificare una rete di calcolatori con un grafo, dove un grafo è appunto un insieme di nodi alcuni dei quali sono collegati con degli archi, nel quale i nodi sono appunto i terminali ed i router mentre gli archi del grafo rappresentano i link, i collegamenti fisici, tra i nodi stessi.

Protocollo di routing 2

Protocollo di routing

- | **Protocollo di routing:** serve per aggiornare le tabelle dei router e quindi determinare il cammino dei pacchetti nella rete
- | **Cammino = sequenza di router nella rete**
- | **Usiamo una rappresentazione astratta della rete in termini di un grafo:**
 - nodi del grafo = router
 - archi del grafo = link fisici
 - varie nozioni di costi associati ai link: ritardo, livello di congestione, costo della trasmissione ...
- | **Algoritmo di routing:** e' alla base di ogni protocollo di routing e serve per determinare il cammino migliore secondo una qualche nozione di costo

Associati a questi archi possiamo avere varie nozioni di costo, nel senso che possiamo essere interessati a sapere qual è il ritardo associato ad un certo link, qual è il costo in termini economici della trasmissione su quello specifico link, possiamo pensare che il costo ci indichi anche qual è il livello di congestione di quel link e altre informazioni di questo genere. Quindi possiamo avere varie nozioni di costo ed avremo più in generale un algoritmo che decide sostanzialmente il cammino su questo grafo, sceglie un cammino all'interno del grafo e solitamente la scelta avviene tenendo conto dei costi associati a tale percorso. Ad esempio un criterio, che è poi quello seguito nella pratica da moltissimi algoritmi di routing, è quello di scegliere cammini di costo minimo in base alla funzione di costo scelta: se ad esempio siamo interessati a trasmettere pacchetti nel più breve tempo possibile, evidentemente sceglieremo la funzione di costo che ci indica il ritardo associato ai vari link e poi useremo un algoritmo opportuno per scegliere il cammino che ha il costo minimo.

Algoritmi di routing

Algoritmi di routing

- | **Tipi di algoritmi di routing:** si possono classificare gli algoritmi di routing come segue
- | **Centralizzati:**
 - ogni router conosce la topologia di tutta la rete e i costi di ogni link
 - algoritmi "link state"
- | **Decentralizzati**
 - un router conosce solo i router adiacenti, ovvero collegati da un link fisico ed i costi di tali link
 - determinazione cammino mediante processo iterativo e scambio di informazioni
 - algoritmi "distanza vector"
- | **Statici**
 - costi dei cammini cambiano lentamente nel tempo
 - variazioni delle tabelle dei router non frequenti
- | **Dinamici**
 - costi cambiano più velocemente
 - variazioni delle tabelle dei router frequenti

Esistono quindi vari algoritmi di routing che si usano per selezionare i cammini nelle reti a commutazione di pacchetto e quindi per definire le tabelle di instradamento e tra i vari algoritmi di routing possiamo distinguere sostanzialmente due tipi: gli algoritmi centralizzati e quelli decentralizzati. Gli algoritmi centralizzati sono quelli nei quali ogni nodo della rete, ogni router conosce la topologia di tutta la rete e i costi di tutti i link, quindi ogni nodo ha una visione completa di tutta la rete; questi algoritmi sono detti di solito link state. Negli algoritmi decentralizzati invece ogni router ha soltanto una visione locale della rete, ad esempio conosce soltanto i router adiacenti e i costi dei link che lo collegano ad essi. In tal caso quindi il singolo router non ha la visione di tutta la rete, per poter determinare quale sia il cammino per arrivare ad una certa estremità della rete, in una zona che non conosce direttamente, ma avrà bisogno di informazioni che gli arrivano dai router vicini. Quindi in questo tipo di algoritmi, un esempio è il famoso algoritmo chiamato distanza vector, il calcolo del cammino migliore avviene attraverso un processo iterativo di scambio di informazioni tra diversi router.

Algoritmo di Dijkstra 1

Algoritmo di Dijkstra

Algoritmo globale (link state)

- | ogni nodo conosce la topologia della rete e i costi di tutti i link
 - questo si ottiene mediante comunicazioni per broadcast dello stato de link
- | ogni nodo ha le stesse informazioni degli altri
- | calcola il costo minimo di tutti i cammini da un nodo (sorgente) a tutti gli altri nodi
 - questo definisce la tabella di routing per quel nodo

Notazione :

- | **S**: nodo sorgente
- | **costo (i,j)**: costo del link dal nodo i al nodo j (infinito se il link non esiste)
- | **dist(v)**: valore corrente della somma dei costi di tutti gli archi nel cammino dal S al nodo v
- | **pred(v)**: nodo che precede v nel cammino di costo minimo da S al nodo v
- | **Fin**: insieme di nodi per cui il cammino minimo da S e' gia' stato determinato

Altra distinzione che è possibile fare tra gli algoritmi di routing è quella tra algoritmi statici e dinamici, dove per statici si intendono quegli algoritmi che si applicano quando i costi dei cammini cambiano abbastanza lentamente nel tempo, cioè le variazioni all'interno alle tabelle di routing non sono molto frequenti, mentre gli algoritmi dinamici sono quelli che si usano quando queste informazioni cambiano abbastanza velocemente e quindi le tabelle devono essere aggiornate più frequentemente. Vediamo ora un esempio di un algoritmo centralizzato globale, cioè un algoritmo di tipo link state, nel quale quindi si assume che ogni nodo conosca esattamente la tipologia di tutta la rete; evidentemente questo tipo di algoritmo non può essere usato a livello geografico, nel senso che è molto difficile che un nodo possa conoscere la topologia di tutta la rete in ambito geografico, ma questo tipo di algoritmi si usa normalmente a livello di reti locali, dove l'estensione della rete, della quale siamo interessati a conoscerne i cammini minimi, non sia particolarmente estesa, cioè il numero di nodi non sia particolarmente elevato. Questo algoritmo è dovuto, come dice il nome stesso, a Dijkstra, un informatico olandese che ha dato notevoli contributi in vari settori dell'informatica, e sostanzialmente questo algoritmo calcola tutti i cammini minimi all'interno di un grafo a partire da un certo nodo che identifichiamo con nodo sorgente per arrivare poi a tutti i nodi che compaiono in questo grafo.

Algoritmo di Dijkstra 2

Algoritmo di Dijkstra

Algoritmo globale (link state)

- | ogni nodo conosce la topologia della rete e i costi di tutti i link
 - questo si ottiene mediante comunicazioni per broadcast dello stato de link
- | ogni nodo ha le stesse informazioni degli altri
- | calcola il costo minimo di tutti i cammini da un nodo (sorgente) a tutti gli altri nodi
 - questo definisce la tabella di routing per quel nodo

Notazione :

- | **S**: nodo sorgente
- | **costo (i,j)**: costo del link dal nodo i al nodo j (infinito se il link non esiste)
- | **dist(v)**: valore corrente della somma dei costi di tutti gli archi nel cammino dal S al nodo v
- | **pred(v)**: nodo che precede v nel cammino di costo minimo da S al nodo v
- | **Fin**: insieme di nodi per cui il cammino minimo da S e' gia' stato determinato

Per poter illustrare questo algoritmo introduciamo un po' di notazione. Come già detto identifichiamo il nodo sorgente, cioè il nodo dal quale vogliamo partire, con S; poi con costo(i,j) denotiamo il costo di un link dal nodo i al nodo j, dove, come abbiamo già detto, il costo può rappresentare informazioni di tipo diverso ad esempio il livello di congestione, il costo economico, eccetera, ma sarà in ogni caso un certo valore numerico e in particolare avrà valore infinito nel caso in cui il link non esista. Indichiamo con dist(v) il valore corrente della somma dei costi di tutti gli archi del cammino da S al nodo v, pred(v) è il nodo che precede v nel cammino di costo minimo dal nodo sorgente S al nodo v. Infine Fin è l'insieme dei nodi del grafo per i quali il cammino di costo minimo dal nodo sorgente è già stato determinato. Quindi lo scopo dell'algoritmo è partire dal nodo S e arrivare ad includere nell'insieme Fin tutti i nodi del grafo.

Algoritmo di Dijkstra 3

Algoritmo di Dijkstra

Inizializzazione

```

F = {S} ;
per ogni nodo v:
se v adiacente S allora
    allora dist(v) = costo(S,v)
    altrimenti dist(v) = infinito ;
  
```

Ripeti fino a quando F non contiene tutti i nodi :

```

{
    fra tutti i nodi non in F scegli quello w per cui dist (w) e' min
    aggiungi w a F;
    per ogni nodo v adiacente a w e non in F:
        dist(v) = minimo( dist(v), dist(w) + costo(w,v) )
        /* il costo del cammino per arrivare a v e' aggiornato
        perche' il cammino di costo minimo o rimane quello
        vecchio o passa attraverso w */
}
  
```

L'algoritmo di Dijkstra si compone di queste due parti che vedete qui in figura: abbiamo come al solito una fase di inizializzazione, dove viene inizializzata la variabile F come un insieme costituito dal solo nodo S, cioè il nodo sorgente, quindi per ogni nodo v che si trova nel grafo fornito in input dall'algoritmo eseguiamo la seguente istruzione: se il nodo v è adiacente ad S, cioè esiste un arco, un link fisico, tra v ed S, allora assegniamo a dist(v), cioè la distanza di v, il costo dell'arco da v ad S, altrimenti, cioè se questo arco non esiste, assegniamo a dist(v) il valore infinito. Notate che in questo esempio stiamo usando un pseudo-linguaggio di programmazione, cioè dei costrutti che ricordano in qualche modo un linguaggio di programmazione di tipo imperativo, senza per questo usare uno specifico linguaggio.

Algoritmo di Dijkstra 4

Algoritmo di Dijkstra

Inizializzazione

```

F = {S} ;
per ogni nodo v:
se v adiacente S allora
    allora  $\text{dist}(v) = \text{costo}(S,v)$ 
    altrimenti  $\text{dist}(v) = \text{infinito}$  ;

```

Ripeti fino a quando F non contiene tutti i nodi :

```

{
    fra tutti i nodi non in F scegli quello w per cui  $\text{dist}(w)$  e' min
    aggiungi w a F;
    per ogni nodo v adiacente a w e non in F:
         $\text{dist}(v) = \text{minimo}(\text{dist}(v), \text{dist}(w) + \text{costo}(w,v))$ 
        /* il costo del cammino per arrivare a v e' aggiornato
        perche' il cammino di costo minimo o rimane quello
        vecchio o passa attraverso w */
}

```

Terminata la fase di inizializzazione abbiamo un ciclo nel quale vengono ripetute un certo insieme di operazioni fino a quando l'insieme F non contenga tutti i nodi. Innanzitutto consideriamo tutti i nodi del grafo che ancora non sono in F (dove F è l'insieme di tutti i nodi del grafo per i quali il cammino di costo minimo è già stato determinato), fra questi nodi rimasti scegliamo il nodo per cui il valore di $\text{dist}(w)$ è minimo; in altri termini scegliamo il nodo per cui il costo del cammino dal nodo in questione al nodo sorgente S è minimo. Questo nodo lo aggiungiamo all'insieme F, avendone determinato il cammino minimo a partire da S. Dobbiamo ora aggiornare la situazione dei nodi adiacenti al nodo w che abbiamo appena aggiunto, cioè per tutti i nodi che sono adiacenti al nodo w che non siano già anch'essi appartenenti all'insieme F facciamo la seguente operazione.

Algoritmo di Dijkstra 5

Algoritmo di Dijkstra

Inizializzazione

```

F = {S} ;
per ogni nodo v:
se v adiacente S allora
    allora  $\text{dist}(v) = \text{costo}(S,v)$ 
    altrimenti  $\text{dist}(v) = \text{infinito}$  ;

```

Ripeti fino a quando F non contiene tutti i nodi :

```

{
    fra tutti i nodi non in F scegli quello w per cui  $\text{dist}(w)$  e' min
    aggiungi w a F;
    per ogni nodo v adiacente a w e non in F:
         $\text{dist}(v) = \text{minimo}(\text{dist}(v), \text{dist}(w) + \text{costo}(w,v))$ 
        /* il costo del cammino per arrivare a v e' aggiornato
        perche' il cammino di costo minimo o rimane quello
        vecchio o passa attraverso w */
}

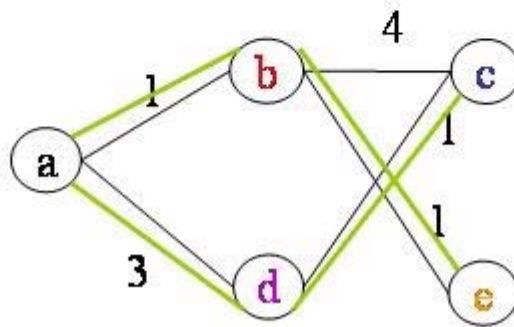
```

Confrontiamo la distanza del nodo dal nodo sorgente, indicata con $\text{dist}(v)$, con la somma $\text{dist}(w) + \text{costo}(w,v)$, cioè la distanza del nodo w a S sommata al costo dal nodo w al nodo v. Se il vecchio valore della distanza del nodo v da S è minore della somma calcolata non facciamo nulla, lasciamo cioè questo valore, altrimenti se è maggiore cambiamo il valore di $\text{dist}(v)$ assegnandogli il valore restituito dalla somma calcolata. In altri termini, noi sappiamo che per un certo nodo v adiacente ad un nodo w, che abbiamo appena aggiunto ad F, esiste un cammino che parte da S e arriva a questo nodo v facendo una certa strada con un relativo costo; con questa operazione controlliamo se esista un cammino che arrivi a v passando da w e se questo cammino abbia costo minore di quello che già conoscevamo. Se ciò è vero, essendo interessati a trovare cammini con costi minimi, evidentemente sceglieremo il cammino di costo minimo, cioè sceglieremo il cammino che passa attraverso w. Queste operazioni vengono appunto ripetute fino a quando l'insieme F non contenga tutti i nodi e, quando terminiamo il ciclo, abbiamo determinato tutti i cammini minimi dal nodo sorgente a tutti i nodi del grafo e, evidentemente, per ricordarsi quali sono questi cammini minimi, basta che ogni volta che noi aggiungiamo un nodo f ci ricordiamo da dove siamo arrivati, cioè ci ricordiamo qual è il predecessore del nodo appena aggiunto nel cammino che dalla sorgente S arriva a quel nodo.

Esempio di applicazione dell'algoritmo 1

Esempio di applicazione dell' algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				

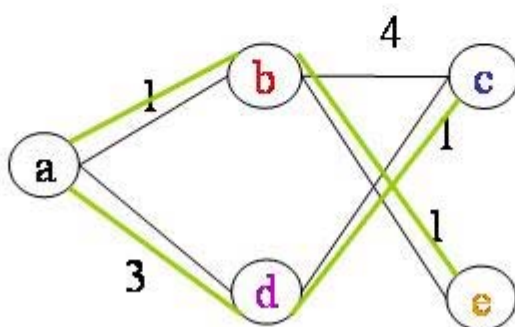


Per cercare di capire un po' meglio questo algoritmo vediamo un esempio di applicazione dell'algoritmo su un caso semplicissimo, quasi banale, che contiene 5 nodi, 5 archi, con i costi che vedete indicati nella figura. Siamo interessati a trovare i cammini minimi dal nodo A, che è quindi il nodo sorgente, a tutti i nodi del grafo. Vediamo che cosa avviene con l'applicazione dell'algoritmo: nel primo passo, indicato con passo 0, l'insieme F sarà costituito dal solo nodo A, dopodiché abbiamo che: la distanza di B dal sorgente A è 1, il predecessore di B è A (questo ci servirà per ricordarsi qual è la sequenza dei nodi del cammino minimo), la distanza di C dal nodo A è infinito, dato che il nodo C non è un nodo adiacente ad A, non esiste un arco che collega direttamente A con C, quindi non ci sarà nemmeno un predecessore di C; esisterà evidentemente un cammino, ma noi al momento non lo conosciamo, dato che nel primo passo controlliamo solamente i collegamenti diretti tra A e gli altri nodi.

Esempio di applicazione dell'algoritmo 2

Esempio di applicazione dell' algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				

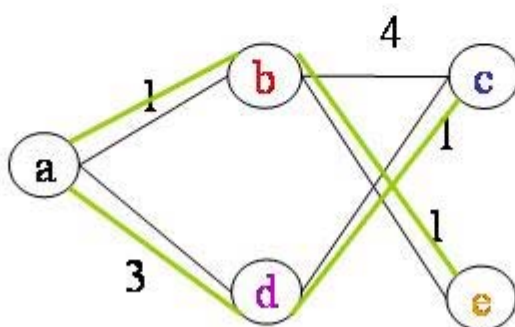


Analogamente possiamo vedere che per il nodo D abbiamo un cammino di costo 3, quindi la distanza da D ad A è 3, il predecessore di D è A. Infine per il nodo E la distanza è infinito, perché non esiste un arco che collega direttamente E ad A e quindi E non ha alcun predecessore. Veniamo ora al passo 1. Tra tutti i valori delle distanze, come abbiamo visto prima nella specifica dell'algoritmo, scegliamo il nodo che ha distanza minore; quindi i valori delle distanze sono rispettivamente 1, infinito, 3, infinito, la distanza minore è quella di B, scegliamo quindi il nodo B e inseriamolo nell'insieme F, questo vuol dire che abbiamo determinato un cammino minimo che va da A a B e questo è rappresentato dall'arco che abbiamo aggiunto indicato in figura dalla linea verde. A questo punto ripetiamo il ciclo, abbiamo 3 distanze: 5, 3, 2 e di nuovo tra queste distanze dobbiamo scegliere quella minore, cioè 2 relativa al nodo E (vi ricordo che le distanze sono calcolate dal nodo in questione al nodo sorgente A); quindi E viene aggiunto all'insieme F dei nodi per i quali i cammini sono stati determinati, il predecessore di E è B, ciò significa che il cammino di costo 2 da E ad A è quello che passa attraverso il nodo B; come già detto il predecessore ci serve appunto per ricordare il cammino fatto per andare da A ad E con tale costo. Dal punto di vista grafico ciò è indicato in figura ancora dall'arco verde che viene aggiunto al grafo.

Esempio di applicazione dell'algoritmo 3

Esempio di applicazione dell'algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				

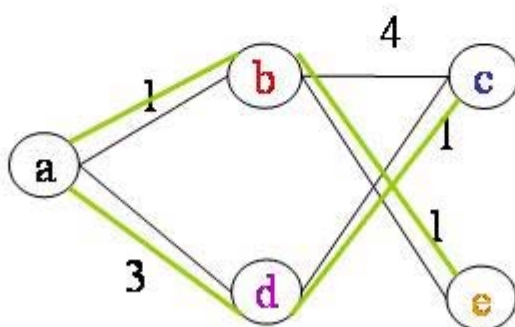


Quindi a questo punto abbiamo 3 nodi, si ripete questo ciclo perché non abbiamo esaurito tutti i nodi possibili, quindi abbiamo 2 nodi da scegliere: il nodo C e il nodo D, al solito scegliamo il nodo di costo minimo; scegliamo il costo minimo 3 relativo al nodo D che viene aggiunto all'insieme F. Aggiungendo tale nodo abbiamo un cammino che arriva a C passando da D, cammino che prima non conoscevamo, quindi mentre prima il collegamento, il cammino da A a C conosciuto era solo quello che passava da B e aveva costo 5, come vedete scritto lì nella terza riga, adesso aggiungendo il nodo D ci accorgiamo che esiste un altro cammino che permette di arrivare a C, passando da A a B e da B a C, che ha costo 4. Per cui scegliamo questo cammino e aggiorniamo la tabella, con il comando visto prima nell'algoritmo, dove sceglievamo il minimo tra i cammini possibili per raggiungere un certo nodo. Ricapitolando, abbiamo trovato un cammino che ha costo minore per arrivare al nodo C di quello visto in precedenza, aggiorniamo allora la tabella di C, dove il valore viene posto a 4 invece di 5 e il predecessore non è più B ma D. Questo lo facciamo soltanto adesso perché nei casi precedenti l'aggiunta di un altro nodo non modificava la situazione degli altri nodi; in generale però ad ogni passo dell'algoritmo, ogni volta che si aggiunge un nuovo nodo all'insieme F, viene controllato se l'aggiunta di questo nodo modifica i costi dei cammini che arrivano ai nodi che ancora non sono nell'insieme F.

Esempio di applicazione dell'algoritmo 4

Esempio di applicazione dell'algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				



Per completare l'algoritmo, a questo punto ovviamente basterà aggiungere il nodo C. Ora abbiamo quindi tutti i nodi A, B, C, D, E del grafo nell'insieme F, l'algoritmo termina e sappiamo tutti i cammini di costo minimo, rappresentati in figura dagli archi verdi, per arrivare a tutti i nodi del grafo con i relativi costi. Questo appena descritto è appunto l'algoritmo di Dijkstra, si può dimostrare che tale algoritmo ha complessità quadratica, è abbastanza buono e, dal punto di vista pratico non può sicuramente essere usato per determinare il cammino minimo di routing all'interno di reti geografiche particolarmente vaste, ma può essere usato all'interno di sistemi autonomi, cioè sistemi abbastanza limitati sia dal punto di vista geografico che dal punto di vista del numero di nodi che lo compongono.

OSPF (Open Shortest Path First) RFC 2178

OSPF (Open Shortest Path First) RFC 2178

- | **L'algoritmo di Dijkstra e' usato principalmente all'interno di "sistemi autonomi". Un protocollo di routing che usa tale algoritmo e' OSPF (Open Shortest Path First):**
 - protocollo del livello link per inviare le informazioni ai nodi della rete (flooding)
 - +
 - algoritmo dei cammini minimi

- | **Caratteristiche principali di OSPF**
 - **Open**, cioè protocollo non proprietario
 - **Mappa** della topologia di rete ad ogni nodo
 - **Informazioni di instradamento** inviate periodicamente da un router a **tutti** i router della rete
 - distanza dal router ai router adiacenti contenute nelle informazioni di instradamento
 - **Algoritmo di Dijkstra** per calcolo dei cammini minimi

Un protocollo di routing reale che usa l'algoritmo di Dijkstra è l'OSPF (Open Shortest Path First), che è un protocollo di tipo Open, cioè non proprietà di una particolare azienda, è un protocollo dove ogni nodo ha una mappa della topologia della rete, essendo infatti applicabile l'algoritmo di Dijkstra ogni nodo ha bisogno di conoscere la situazione di tutta la rete, cioè i costi di tutti gli archi; per avere queste informazioni ogni router teoricamente invia delle informazioni di instradamento a tutti gli altri router della rete. Come già detto questo protocollo usa l'algoritmo di Dijkstra.

Caratteristiche avanzate di OSPF 1

Caratteristiche avanzate di OSPF

- | **Sicurezza**: tutti i messaggi OSPF sono autenticati per prevenire intrusioni. Sono usate connessioni TCP.
- | Permessi **percorsi multipli** con lo stesso costo.
- | **Costi metrici diversi** per lo stesso link per Tipi di Servizio diversi
- | **Supporto per instradamento multicast**
- | **OSPF gerarchico** : un sistema autonomo può essere strutturato gerarchicamente e l'indirizzamento OSPF tiene conto di tale gerarchia.

Come già detto questo protocollo usa l'algoritmo di Dijkstra con alcune caratteristiche aggiuntive, tra queste ce ne sono alcune che permettono un maggior controllo sulla sicurezza del protocollo stesso e altre che invece permettono di migliorare l'instradamento, ad esempio tenendo conto della struttura gerarchica che può esistere in un certo sistema, dove per struttura gerarchica si intende il fatto che un sistema può essere organizzato a livelli, non visto solo come un grafo ma anche come un albero e di questi livelli dell'albero si deve tener conto per migliorare l'instradamento. Nell'OSPF si tiene appunto conto di queste caratteristiche e anche del fatto che si possono usare costi metrici diversi per lo stesso link a seconda del tipo di servizio del quale si ha bisogno; sono permessi anche percorsi multipli con lo stesso costo e esiste anche il supporto per l'instradamento di tipo multicast.

Caratteristiche avanzate di OSPF 2

Caratteristiche avanzate di OSPF

- | **Sicurezza**: tutti i messaggi OSPF sono autenticati per prevenire intrusioni. Sono usate connessioni TCP.
- | Permessi **percorsi multipli** con lo stesso costo.
- | **Costi metrici diversi** per lo stesso link per Tipi di Servizio diversi
- | **Supporto per instradamento multicast**
- | **OSPF gerarchico** : un sistema autonomo puo' essere strutturato gerarchicamente e l'indirizzamento OSPF tiene conto di tale gerarchia.

Per quanto riguarda la sicurezza, come vi accennavo prima, per trasmettere informazioni vengono usate semplicemente connessioni TCP-IP in modo tale da essere sicuri che i dati inviati arrivino effettivamente a destinazione e tutti i messaggi OSPF sono autenticati per prevenire intrusioni, perché chiaramente quello che vogliamo evitare è che informazioni necessarie al calcolo del cammino minimo, e quindi al calcolo dell'instradamento dei pacchetti, cioè le informazioni necessarie per valutare il traffico nella rete, possano essere influenzate da intrusioni esterne, potrebbero esserci degli intrusi, interessati per vari motivi a modificare lo stato di instradamento di una rete e ad arrivare a provocare la congestione della rete stessa. Per evitare questo tipo di situazioni vengono usati i meccanismi di autenticazione. Questo termina ciò che avevo da dirvi sull'aspetto del routing sulle reti, ovviamente abbiamo visto soltanto una panoramica generale riguardo a cosa sia il routing in reti a commutazione di pacchetto, un esempio di algoritmo di routing, ne esistono altri altrettanto importanti, in particolare l'algoritmo di distante vector, che permette il calcolo di cammini minimi in modo distribuito e che è abbastanza usato dal punto di vista pratico. Con questo avrei concluso.