CAPITOLO 4: Livello di rete

**4.1 Introduzione**

Il compito del livello di rete è quello di trasportare dati all’interno di pacchetti da un host sorgente ad un host destinatario. A differenza del livello di trasporto dove ho un’entità di trasporto nel nodo mittente, un’entità di trasporto nel nodo ricevente e i due si scambiano segmenti che trasportano messaggi dell’applicazioni attraverso la rete. Come è fatta la rete? È composta da entità di rete che sono presenti in tutti i nodi e hanno il compito di implementare il servizio di trasferimento di segmenti tra nodo sorgente e nodo destinatario. Il livello 3 deve quindi svolgere una serie di compiti:

1. Calcolo del percorso (Routing) ossia come le varie entità della rete calcolano il percorso per far arrivare i dati da un nodo sorgente ad un nodo destinatario.
2. Forwarding ossia instradare questo traffico. Devo capire come ricevo un pacchetto, come lo elaboro e come lo inoltro verso un nodo di uscita.
3. Call setup ossia instaurare delle connessioni logiche nella rete per inviare traffico.

Un algoritmo di instradamento popola le tabelle di instradamento usate dai router quando devono decidere cosa fare con ogni pacchetto ricevuto ottenendo così, tramite il campo di intestazione, a quale interfaccia di collegamento il pacchetto deve essere diretto. Il campo su cui viene determinato l’instradamento è l’indirizzo di destinazione ossia un insieme di 32 bit e per semplicità gli indirizzi vengono suddivisi in insiemi (dal bit x al bit y l’interfaccia è 1, dal bit y al bit z l’interfaccia è 2, etc).

Riguardo la call setup, le connessioni logiche vengono stabilite tra due host mentre nel livello di trasporto le connessioni logiche erano tra processi.

**Servizi offerti**

I servizi offerti agli utenti di livello superiore a singoli datagrammi sono:

-garanzia della consegna

-garanzia della consegna entro un certo tempo

I servizi offerti agli utenti di livello superiore a d un flusso di datagrammi sono:

-la sequenza di arrivo è ordinata

-garanzia di una banda minima garantita

-restrizioni sulle variazioni dei ritardi di pacchetti consecutivi

Le caratteristiche dei servizi offerti da una rete internet sono:

-best-effort

-non c’è una garanzia di una banda minima

-non garantisce nulla sui pacchetti persi

-non da garanzia sull’ordine

-non da garanzia sul timing

-non da informazioni sul feedback in caso di congestione

Altre architetture di rete hanno definito modelli di servizi che vanno oltre al servizio best-effort di Internet, come per esempio quella ATM. Erano reti integrate che offrivano diversi modelli di servizi alla rete:

1. CBR: supportava servizi con banda costante (chiamata), esigenze stringenti sulla perdita, sull’ordine, sul timing e non c’è congestione.
2. VBR: supportava garanzia con banda variabile (videochiamata), esigenze stringenti sulla perdita, sull’ordine, sul timing e non c’è congestione.
3. ABR: supportava servizi con banda minima garantita (servizi dati), nessuna garanzia sulle perdite, garanzie sull’ordine, nessuna garanzia sul timing e c’era un meccanismo di controllo di congestione.
4. UBR: non supportava nessuna garanzia tranne che sull’ordine (circuito virtuale).

**4.2 Reti Datagram (Internet) e Circuito Virtuale (ATM)**

Sia nella rete internet che ATM c’era la possibilità di avere una rete con connessione o senza connessione. Le reti a datagramma sono reti senza connessioni (reti IP) mentre le reti a circuito virtuale sono reti a pacchetto con connessioni.

1. Internet: è nata per realizzare una rete che permettesse lo scambio di dati tramite calcolatori. Serviva per realizzare della comunicazione tra apparati intelligenti che si sapevano adattare, lasciando la complessità all’interno dei calcolatori.
2. ATM: offriva un servizio di rete complessa, nata come un’evoluzione della rete telefonica che supportava anche servizi dati. Nel tempo scomparve per la troppa complessità e oggi la rete telefonica viaggia su internet

**4.3 Struttura router**

È un apparato con due funzioni principali:

-deve eseguire gli algoritmi di instradamento

-si occupa di instradare i datagrammi da una porta di ingresso ad una porta di uscita.

È formato da:

1. Porte di ingresso, svolgono le funzioni a livello fisico di terminazione di un collegamento in ingresso al router, svolgono le funzioni a livello di collegamento, svolgono la funzione di ricerca in modo che il pacchetto inoltrato esca sulla porta di uscita corretta. In generale non è detto che la matrice di commutazione riesce a commutare i pacchetti che provengono dalle porte di ingresso alla loro stessa velocità e questo può causare un
2. Porte in uscita, in cui il pacchetto viene bufferizzato per poter essere trasmesso secondo la disciplina di servizi ossia quale pacchetto trasmetto (per esempio FIFO).
3. Switching fabric (matrice di commutazione) ossia una struttura che permette di scambiare pacchetti tra porte di ingresso e uscita. Può essere una memoria, un bus o una crossbar.
4. Processore di routing che interagisce con la matrice di commutazione per scambiarsi dati su cui viene fatto l’instradamento.

**Switching**

La switching fabric è il cuore del router, attraverso questa fabbrica i pacchetti sono commutati da una porta di input ad una porta di output. Può essere realizzata nei seguenti modi:

-Nello switching memory la commutazione viene fatta attraverso la CPU. Il pacchetto viene ricevuto, copiato nella memoria del processore, gli viene estratto l’indirizzo di destinazione, viene copiato verso la porta di uscita trovata tramite la tabella di forward e poi inviato. In questo modo non si possono inoltrare due pacchetti allo stesso tempo anche se questi si trovano in porte differenti poiché una singola memoria può scrivere o leggere allo stesso tempo;

-Nello switching bus evito di passare per la memoria e i dati vengono trasferiti alle porte di uscita attraversando una sola volta un bus condiviso in cui, tramite un’etichetta di commutazione, viene indicata la porta di destinazione di ciascun pacchetto. È più efficiente rispetto alla memory. Se arrivano molti pacchetti al router nello stesso tempo, ognuna su porte differenti, tutti eccetto uno aspetta poiché solo un pacchetto alla volta può attraversare il bus. Dunque, dato che ogni pacchetto deve attraversare il bus, la larghezza di banda della commutazione è limitata da quella del bus.

-Nella rete di interconnessione (crossbar) viene eliminato il problema della limitazione di bando di un singolo bus, infatti, viene creata una rete di 2n bus che collegano n porte di ingresso con n porte di uscita. Ogni bus verticale interseca ogni bus orizzontale. Quando arriva un pacchetto dalla porta A e ha bisogno di essere inoltrata alla porta Y, il controller dello switch chiude il punto di incrocio nell'intersezione A-Y, e la porta A allora invia il pacchetto nel suo bus il quale viene “preso” esclusivamente da Y. Questo consente di non bloccare un pacchetto quando arriva al router anche in presenza di altri pacchetti. L'unico caso in cui si dovrà aspettare è quando le porte in ingresso sono diverse, ma la porta di destinazione è uguale.

**Accodamento nei Router**

Nei router si possono formare code di pacchetti sia presso le porte di ingresso sia quelle di uscita. Il dove si verifica e la dimensione della coda dipende dal carico del traffico di rete, dalla velocità relativa della switching fabric e dalla velocità di linea. Porte di uscita: Supponiamo che il rate dello switch è N volte più veloce del rate di linea e che i pacchetti in arrivo ad ogni N porta di inpute sono destinati alla stessa porta di output. In questo caso, nel tempo necessario ad inviare un singolo pacchetto sul link di uscita, arriveranno N pacchetti alla porta di uscita. Dato che la porta di uscita può trasmettere solo un singolo pacchetto per unità di tempo, gli N pacchetti in arrivo dovranno accordarsi per la trasmissione sul link di uscita. Ci sono i buffer che bufferizzano i pacchetti in attesa di trasmissione, i quali hanno, però, dimensione finita e in particolare la dimensione suggerita è di:



con C=capacità link e N=valore stimato del numero di flussi nel buffer ma se i pacchetti eccedono questa dimensione vengono scartati.

Porte di ingresso, in modo analogo, se non c’è un trasferimento di pacchetti rapido, può verificarsi accodamento. In particolare, può verificarsi il fenomeno del Blocco in testa alla coda, ossia un pacchetto nella coda di ingresso deve attendere il trasferimento anche se la sua porta di destinazione è libera, in quanto è bloccato da un altro pacchetto che lo precede, il quale è in attesa di trasferimento a causa di un conflitto della stessa porta di uscita con un altro pacchetto. Per questo per progettare router efficienti bisogna guardare oltre al primo pacchetto in coda così da evitare i ritardi di accodamento.

**4.4 Protocollo Internet (IP)**

Era un protocollo che stava sopra i livelli 3 delle altre reti differenti tra loro che mi permetteva di farle comunicare tra loro. Attualmente non ci sono più reti di livello 3 diverse ma ci sono reti che sono tecnologie di livello 1 e 2 per esempio rete fissa, rete 4G, reti in fibra e tutte parlano tra loro usando router IP perché sopra questi livelli 2 c’è un unico livello 3 ossia il livello IP (protocollo Internet). Quindi ora abbiamo un unico protocollo di rete IP, sopra abbiamo 2 protocolli di trasporto TCP, UDP e tutte le applicazioni, sotto IP abbiamo tanti livelli 1 e 2 come ethernet, ATM, 4G: “everything over IP” ossia chi sviluppa applicazioni è interessato solo a fare applicazioni che girano su una rete IP e non a come è fatta la rete a livello fisico; “IP over everything” ossia per tutte le tecnologie è importante che ci giri sopra IP.

**Formato dei datagrammi IP**

Un datagramma IP come TCP è formato da 20 byte più dei byte opzionali per funzioni particolari:

1. La parte obbligatoria è l’indirizzo IP del nodo sorgente (32 bit) e del nodo destinatario (32 bit).
2. La prima parola di memoria è formata da:

-campo di 16 bit che indicano la lunghezza del datagramma IP che è di 65000 byte.

-campo di 3 bit della versione di IP (versione corrente IPv4).

-campo di 5 bit della lunghezza dell’intestazione, necessari a determinare dove il payload all'interno del datagramma inizia.

-campo di 8 bit per indicare pacchetti di tipo diverso

1. La seconda parola di memoria viene usata per gestire la frammentazione ossia se il pacchetto supera il massimo di dimensione supportato, viene frammentato e poi ricostruito.
2. La terza parola di memoria è formata da:

-campo di 16 bit di header checksum ossia checksum dell’intestazione, controllando se ci sono errori nell’intestazione

-campo di 8 bit di upper layer ossia quando un pacchetto arriva ad un nodo, devo sapere quale protocollo di livello 4 deve gestire il campo dati di quel pacchetto.

- campo di 8 bit di time to live ossia il tempo di vita massimo del pacchetto. Questo campo viene decrementato di un’unità ogni volta che il datagramma è elaborato da un router e quando raggiunge 0 il datagramma viene scartato.

**Frammentazione**

Definisco la MTU (maximum transfer size) ossia la massima quantità di dati che un frame a livello di collegamento può trasportare. Le tratte del percorso tra mittente e destinatario possono usare diversi protocolli e dunque avere diversi MTU e se ricevo un datagramma IP che ha lunghezza maggiore del MTU devo frammentare i dati del datagramma IP in due o più datagrammi IP più piccolo detti frammenti e poi trasferirli sul collegamento di uscita. I frammenti dovranno poi essere riassemblati prima di raggiungere il livello di trasporto di destinazione tramite un identificatore, un flag e un campo offset che specifica l’ordine esatto originario dei frammenti, per capire se qualche frammento è andato perduto.

**Indirizzamento IPv4**

Un host ha un solo collegamento con la rete e quando l’implementazione di IP dell’host vuole inviare un datagramma, lo fa su tale collegamento. Il confine tra host e collegamento fisico è detto interfaccia. Poiché il compito di un router è ricevere datagrammi da un collegamento e inoltrarli su un altro, questo deve essere connesso ad almeno altri due collegamenti e anche il confine tra un router e i suoi collegamenti è detto interfaccia. Quindi il router ha più interfacce ognuna con il proprio indirizzo IP. Essi sono lunghi 32 bit e tali indirizzi sono scritti nella notazione decimale puntata, in cui ciascun byte dell’indirizzo viene indicato in forma decimale ed è separato con un punto dagli altri byte. Ogni indirizzo è diviso in due parti: NetID che identificano la rete e HostID che identificano l’host all’interno della rete.

Inoltre, chiamiamo sottoreti o reti IP quelle porzioni della rete globale che hanno i nodi che comunicano tra loro senza far passare il loro traffico attraverso un router. Le diverse sottoreti hanno una parte dell’indirizzo uguale che le identificano. I calcolatori che si trovano in sottoreti diverse possono comunicare tra loro grazie ad un router. Per determinare le sottoreti (subnets), scollega ogni interfaccia dai suoi host/router, creando isole di reti isolate. Ognuna di queste reti isolate vengono dette subnet. L'indirizzamento IP assegna un indirizzo alle sue subnet per esempio 233.1.1.0/24, in cui /24 è detta subnet mask. Esso indica che i primi 24 bit dei 32 definiscono l'indirizzo della subnet, i restanti 8 identificano l’host. Originariamente quando sono stati assegnati gli indirizzi, si usava uno schema a classi (classe A, B, C, D), ciascuna con caratteristiche diverse in base al numero di bit che identificavano la rete e l’host. Era però uno schema troppo rigido.

Un altro tipo di indirizzamento era dato dagli indirizzi IP broadcast, ossia quando un host invia un datagramma con indirizzo broadcast, il messaggio viene consegnato a tutti gli host sulla stessa subnet. I router opzionalmente inoltrano il messaggio all'interno delle sottoreti vicine.

L'assegnamento degli indirizzi IP viene effettuata con la strategia detta Classless InterDomain Routing **CIDR**, ossia dato un indirizzo IP non esiste uno schema rigido che per separare i bit della rete e dell’host ma di volta in volta lo devo identificare. CIDR, dunque, generalizza la nozione di indirizzamento di sottorete: l’indirizzo IP a 32bit viene diviso in due parti in notazione decimale puntata da a.b.c.d/x in cui x indica il numero di bit nella prima parte dell'indirizzo, cioè la porzione di rete dell'indirizzo IP.

**Come ottenere l’indirizzo di un host**

Si usa un meccanismo automatico per farsi assegnare gli indirizzi e per far ciò si usa un protocollo DHCP, il quale si fa assegnare l’indirizzo di rete dinamicamente da un server.

Per gli host nuovi arrivati, il protocollo DHCP è composto da un processo a quattro passi:

• DHCP server discovery: il primo compito di un host appena arrivato è quello di trovare un server DHCP con il quale interagire. Questo viene fatto utilizzando un DHCP discover message inviato all'interno di un pacchetto UDP sulla porta 67. Il pacchetto UDP viene incapsulato in un datagramma IP. L'host non conosce l'indirizzo IP della rate al quasi si connette e neanche l'indirizzo del server DHCP per questa rete. Quindi, il client DHCP crea un datagramma IP contenente il suo meggaggio discover DHCP e a questo punto il client DHCP passa il datagramma IP al livello di link che infine invia in broadcast il frame a tutti i noti attaccati alla sottorete;

• DHCP server offer(s): un server DHCP ricevente un messaggio di scoperta DHCP risponde al client con un DHCP offer message che viene inviato in broadcast a tutti i nodi della sottorete attraverso un indirizzo IP broadcast. Dato che possono essere presenti molteplici server DHCP, il client si trova nella scelta fra tante offerte, ciascuna delle quali contiene l'ID della transazione del messaggio di identificazione ricevuto, l'indirizzo IP proposto per il client, la maschera di rete e il tempo di scadenza dell'indirizzo;

• DHCP request: il client sceglierà tra uno o più offer del server e risponderà a quella selezione con un DHCP request message, ritornando i parametri di configurazione;

• DHCP ACK: il server risponde al messaggio di richiesta DHCP con un DHCP ACK message confermando i parametri richiesti.

Una volta che il client riceve DHCP ACK, l'interazione è completata e il client può usare l'indirizzo IP allocato da DHCP per la durata della scadenza. Dato che un client potrebbe utilizzare il suo indirizzo oltre la scadenza, DCHP fornisce un meccanismo che permette al client di rinnovare la scadenza dell'indirizzo. Inoltre, questo tempo di scadenza mi permette di utilizzare un numero finito di indirizzi per un numero molto grande di utenti.

**Come ottenere un blocco di indirizzi**

Per ottenere un blocco di indirizzi IP da usare in una sottorete, un amministratore di rete deve contattare il proprio ISP, che potrebbe fornire degli indirizzi attingendo da un blocco più grande. Se è stato allocato per esempio un blocco di indirizzi 200.23.16.0/20, potrebbe dividerlo in otto blocchi uguali di indirizzi e fornirne a ciascuna delle 8 organizzazione che supporta. Ogni organizzazione, quindi, ha un insieme di indirizzi assegnati con un prefisso assegnato. Rivolgersi ad un ISP non è l’unico modo, considerando che esso a sua volta deve richiedere un blocco di indirizzi. Infatti, esiste un’autorità globale che si occupa di gestire lo spazio di indirizzamento IP e di allocare blocchi di indirizzi ossia la ICANN, la quale non solo alloca indirizzi ma gestisce i root DNS server e gestisce i nomi di dominio e la sua risoluzione di dispute.

**NAT**

Per risolvere il problema del non avere più indirizzi, sono state introdotte le NAT secondo le quali si può riusare lo stesso intervallo di indirizzi e dunque posso avere più nodi che hanno lo stesso indirizzo. In particolare, se un router è di tipo NAT fa un’operazione particolare: quando riceve per un nodo interno un pacchetto da trasmettere verso l’esterno, modifica il campo sorgente di tutti i pacchetti, sostituendo al campo sorgente l’indirizzo IP della propria interfaccia e assegna un nuovo numero di porte.

Il motivo del NAT è che in questo modo la rete locale usa un solo indirizzo IP poiché gli altri sono indirizzi all’interno della rete e dunque privati e possono essere modificati. Inoltre, posso cambiare ISP senza cambiare lo schema di indirizzamento interno. Inoltre, poiché i nodi interni hanno indirizzi non visibili, i dispositivi interni alla rete non sono indirizzabili in modo esplicito e questo avrà un impatto sulla sicurezza. Infine, i nodi all’interno della rete possono comunicare tra loro senza problemi.

**Funzionamento router NAT**

Suppongo che un nodo voglia mandare un pacchetto verso l’esterno, lo inoltra al router NAT il quale associa nella propria tabella all’indirizzo IP interno il numero di porta e all’indirizzo IP esterno un altro numero di porta e modifica la parte sorgente dell’indirizzo e del numero di porta. Il pacchetto viene inviato all’esterno e il server ricevente invia la risposta al router (poiché non sa quale nodo glielo ha inviato), il quale dalla tabella sostituisce i valori di indirizzo e porta del nodo interno e lo inoltra alla destinazione corretta. In tutto ciò, il client e il server non si accorgono che c’è un router che fa da NAT. Quindi riassumendo:

-Per tutti i datagrammi in uscita viene sostituito l’IP e il numero di porta con l’indirizzo IP NAT e un nuovo numero di porta. Client e server rispondono come se il NAT fosse la sorgente

-Nella tabella NAT vengono inserite le informazioni precedenti

-Nei pacchetti in ingresso cerco la corrispondenza e reinserisco il numero di porta originario e IP originario.

Osservo che se dall’esterno arriva un pacchetto che non trovo nella tabella NAT, quel pacchetto viene scartato per una questione di sicurezza poiché non ho trovato nessuna corrispondenza nella tabella. Inoltre, poiché devo assegnare un numero di porta diverso, ciò limita il numero di connessioni. Infine, il router in generale opera fino al livello 3 ma un router NAT deve modificare gli indirizzi, quindi, deve per forza operare fino al livello 4.

**Problemi NAT**

Se un nodo nella rete genera pacchetti verso l’esterno tutto funziona, ma se nella mia rete ho un server 10.0.0.1, il quale dovrebbe avere un IP pubblico ed essere raggiungibile, esso ha un indirizzo IP privato e quindi dall’esterno nessuno può raggiungerlo. Questo è detto il problema dell’”attraversamento della NAT”. La soluzione è per esempio configurare in modo statico all’interno della tabella la corrispondenza indirizzo esterno/porta 80 e indirizzo interno/porta 80 così tutto il traffico diretto a questo indirizzo IP/porta 80 viene instradato al server corretto. Questa soluzione funziona ma per un solo server all’interno della rete. Una seconda soluzione consiste nel protocollo IGD che permette di configurare in maniera automatica le corrispondenze all’interno del router NAT.

Come fanno i client a comunicare tra loro?

Prendiamo in esempio il funzionamento di skype:

c’è un client che vuole contattare un altro client che sta dietro un router NAT. Si usa un meccanismo per il quale ogni client, sia interno che esterno alla rete, deve stabilire una connessione con un nodo di relay (server esterno skype) così lo scambio di traffico P2P tra i due client avviene tramite questo server esterno. Quindi un’applicazione P2P a causa delle NAT si ritrasforma in un’applicazione client-server.

**4.5 Algoritmi di instradamento**

Il problema di instradare un pacchetto tra sorgente e destinazione si riconduce al problema di instradare il pacchetto tra questi due router. Dato un insieme di router interconnessi, lo scopo di un algoritmo di instradamento è quello di trovare un buon percorso da quello sorgente a quello di destinazione, ossia il percorso con costo minore. Per formulare i problemi di instradamento si usa un grafo G=(N,E) ossia un insieme di N nodi e un insieme di E archi dove ciascun arci collega una coppia di nodi. I nodi rappresentano i router mentre gli archi rappresentano i collegamenti fisici tra i router. Gli archi sono etichettati ossia esiste un costo per attraversare quel link e il costo del percorso è dato dalla somma dei costi degli archi. Gli algoritmi di instradamento sono classificabili come globali o decentralizzati ossia che tipo di informazione ho bisogno per calcolare il cammino minimo:

1. Globali: viene calcolato il percorso a costo minimo tra sorgente e destinazione sapendo che tutti i router hanno completa informazione sulla tipologia della rete e sui costi dei link. Questo tipo di algoritmo viene detto di tipo link state poiché è a conoscenza del costo di ciascun collegamento della rete.
2. Decentralizzati: viene calcolato il percorso a costo minimo tra sorgente e destinazione ma non si hanno informazioni complete sulla tipologia della rete o sui costi dei link ma inizialmente i nodi conoscono solo i costi dei collegamenti a loro adiacenti. Successivamente attraverso un processo iterativo e lo scambio di informazioni con i nodi adiacenti, un nodo calcola il percorso a costo minimo verso una destinazione. Questo tipo di algoritmo viene detto di tipo distance vector poiché ogni nodo elabora un vettore di stima dei costi verso gli altri nodi della rete.

Un altro modo per classificare gli algoritmi di instradamento riguardo il fatto di essere statici o dinamici:

1. Statici: i percorsi cambiano lentamente nel tempo.
2. Dinamici: i percorsi cambiano velocemente nel tempo. Può essere eseguito sia periodicamente o come conseguenza diretta di un cambiamento nella tipologia o nel costo di un collegamento.

**Algoritmo di instradamento link state: Dijkstra**

Assume che sia nota l’insieme dei nodi, l’insieme degli archi e i relativi costi. Ogni nodo periodicamente invia agli altri nodi lo stato di tutti i link a cui è collegato in modo che dopo un intervallo di tempo tutti i nodi hanno queste informazioni. Una volta ottenute le informazioni, ciascun nodo si calcola il percorso di costo minore da ogni nodo ad ogni altro nodo. Così si può costruire la tabella di instradamento. L’algoritmo è iterativo. La notazione utilizzata è:

-c(x,y) è il costo del link dal nodo x al nodo y, assumendo che sia infinito se i nodi non sono vicini.

-D(v) è il valore corrente del costo di un percorso verso una destinazione v.

-p(v) è il predecessore lungo un percorso che va fino a v.

-N’ è l’insieme dei nodi per cui ad un certo passo k ho già calcolato il modo esatto per cui l’insieme dei nodi è di costo minimo.

L’algoritmo consiste in un passo di inizializzazione e un ciclo che viene eseguito una volta per ogni nodo del grafo:

Inizializzazione

Il nodo u esegue l’algoritmo. Mi chiedo quale è il percorso di costo minimo verso tutte le possibili destinazioni. N’=u mentre per tutti gli altri nodi se v è adiacente a u allora inizializzo il costo verso v con il costo dell’arco stesso D(v)=c(u,v) altrimenti imposto il costo a infinito.

Loop

Prende il nodo w che non è in N’ tale che D(w) è minimo;

Aggiunge w a N’;

Aggiorna D(v) per tutti i nodi adiacenti a w che non sono in N’ in modo che D(v)=min(D(v),D(w)+c(w,v));

Itero fino a che tutti i nodi sono in N’.

L’algoritmo di dijsktra permette di calcolare in n passi il costo e il percorso di costo minimo verso tutte le destinazioni partendo da un nodo. L’algoritmo è dell’ordine di O(n^2) anche se si può ottimizzare fino ad avere O(nlogn) ed in generale è un algoritmo facile da implementare. Nel caso in cui il costo dei link è rappresentato dal carico attraverso di essi per rappresentare la congestione del link si verificano delle oscillazioni del cammino poiché nell'esecuzione dell'algoritmo verrano trovati path differenti in base al traffico di rete. Per prevenire queste oscillazioni una soluzione potrebbe essere quella di evitare di assegnare il costo del link sull'ammontare di traffico oppure è quella di assicurarci che non tutti i router eseguono l'algoritmo LS nello stesso tempo.

**Algoritmo di instradamento Distance Vector:**

L’algoritmo DV è distribuito, in quanto ciascun nodo riceve parte dell’informazione da uno o più dei suoi vicini direttamente connessi a cui successivamente restituisce i risultati del calcolo, iterativo, in quanto questo processo si ripete fino a che non c’è più scambio di informazioni tra vicini e asincrono, in quanto non richiede che tutti i nodi operino al passo con gli altri.

Questi algoritmi si basano sull’equazione di programmazione dinamica di Bellman-Ford. Sia dx(y) il costo del percorso a costo minimo dal nodo x al nodo y, sia minv il minimo tra tutti i vicini di x, allora i costi minimi si basano su questa equazione:

Ossia il costo del percorso di costo minimo è pari al minimo preso tra tutti i vicini del costo con cui la x va ai vicini più il costo del percorso di costo minimo tra il generico vicino e la destinazione.

L’idea è usare l’equazione in modo iterativo. Definisco Dx(y) la stima del costo del percorso di costo minimo da x a y, Dx= [Dx(y):y in N] l’insieme della stima presso x del costo verso tutte le destinazione y in N. Ogni nodo mantiene i seguenti dati:

- il costo c(x,v);

-il vettore delle istanze Dx;

-i vettori delle istanze di ciascuno dei suoi vicini Dv.

Lavorando in modo asincrono, quando un nodo riceve un vettore delle istanze da un vicino, aggiorno la stima delle istanze usando l’equazione di BF ma al posto di “d” c’è “D”.

L’algoritmo è fatto in questo modo:

-Ogni nodo è in attesa di avere informazioni dai vicini o che cambi qualcosa su un link locale;

-In entrambi i casi ricalcolo le mie stime (c(x,v) oppure Dv(y));

-se il vettore dei costi cambia allora lo comunica ai miei vicini, altrimenti mi metto in attesa;

**Problema conteggio all’infinito e avvelenamento percorso**

In una rete può succedere che vengono cambiati i pesi. Questo può accadere o quando qualcuno cambia un link oppure quando qualcosa si rompe. Se un link cambia per esempio il costo, vengono cambiate le stime e vengono informati i vicini. In particolare, se il costo diminuisce tutto migliora ossia l’algoritmo converge velocemente, ma se il costo aumenta l’algoritmo ci mette tantissimo a convergere e la rete entra in loop. Questo perché gli algoritmi distance vector conosco il costo di un percorso ma non conosco il percorso per dove passa quindi questi algoritmi si comportano malissimo quando c’è un aumento del costo di un link. Per evitare che questo avvenga avviene quello che viene chiamato “avvelenamento del percorso inverso”: se un nodo z fa passare il traffico attraverso y per raggiungere x, quando z invia il vettore delle distanze a y gli dice che verso la destinazione x, che raggiunge passando per y, non ci sa arrivare (infinito). Questo fa si che si riduce il problema ma non si risolve del tutto.

**Confronto LS e DV**

Nel calcolare l’instradamento i due algoritmi sono differenti. Nel LS, ciascun nodo comunica con tutti gli altri nodi via broadcast ma li informa solo dei costi dei link direttamente connessi. Nel DV ciascun nodo dialoga solo con i vicini direttamente connessi, dando informazioni sulle stime a costo minimo da sé stesso a tutti i nodi nella rete. Altre differenze sostanziali sono:

Complessità messaggi:

LS: ogni nodo conosce il costo di ogni collegamento nella rete e necessito quindi di O(nE) messaggi inviati.

DV: ogni nodo comunica solo con i vicini.

Velocità convergenza:

LS: O(n^2).

DV: il tempo di convergenza può variare. Posso avere dei loop e c’è il problema del conteggio all’infinito.

Robustezza (in caso di malfunzionamenti):

LS: ogni nodo può comunicare un’informazione scorretta tramite un link e ogni nodo si calcola la propria tabella. Gli errori sono quindi limitati all’area dove c’è il nodo difettoso e ciò fornisce un certo grado di robustezza.

DV: manda messaggi relativi a percorsi sull’intera rete e quindi ci sono errori che si propagano su tutta la rete.

**Instradamento gerarchico**

Il routing su internet è un discorso molto più ampio in cui LS e DV non possono rientrare in quanto sono algoritmi che convergerebbero dopo troppo tempo a causa della vastità del problema. Il problema dell’instradamento viene suddiviso in due livelli ossia procedendo in modo gerarchico. Infatti, fino ad ora si è parlato di router tutti uguali e la rete considerata come flat ossia i nodi sono più o meno dello stesso valore. In realtà non è così sia perché ci sono 200 milioni di rete quindi le tabelle non posso essere memorizzate tutte e i nodi si scambierebbero troppe informazioni, sia perché la rete è organizzata come un insieme di reti sotto responsabilità amministrativa di entità diverse (problemi di scalabilità e autorità amministrativa).

Si introduce il concetto di sistema autonomo, ossia un sistema composto da regioni (gruppi di router) sotto lo stesso controllo amministrativo che usano lo stesso protocollo di instradamento, come DV e LS, detto protocollo intra-sistema autonomo. Perciò diversi sistemi autonomi posso usare diversi protocolli intra-sistemi autonomi. Per far funzionare l’intera rete come una cosa sola, vengono introdotti dei router speciali “gateway routers” che stanno al confine tra sistemi autonomi, i quali eseguono il protocollo intra-sistema autonomo nella rete ed inoltre eseguono protocolli diversi che hanno il compito di permettere l’instradamento tra sistemi autonomi (protocolli inter-sistemi autonomi). Quindi le tabelle di routing dei nodi sono popolate con le informazioni sia interne che esterne al sistema autonomo.

Riguardo il funzionamento di un protocollo inter-sistema autonomo, immagino che un router in una rete abbia un pacchetto da inviare all’esterno della rete. Se il sistema autonomo presenta un solo gateway connesso ad un altro sistema autonomo, l’instradamento del pacchetto è un problema facile. Infatti, l’algoritmo di instradamento interno ha determinato il percorso a costo minimo da ogni router interno e questi sanno come inoltrare un pacchetto, il quale viene ricevuto dal gateway che a sua volta lo inoltra all’esterno. Se il sistema autonomo presenta diversi gateway, si usano i protocolli inter-sistemi autonomi che permettono di far conoscere tutte le destinazioni raggiungibili e riportarle a tutti i router interni al sistema. A questo punto, una tecnica utilizzata per l’instradamento è quella dell’”hot potato routing”, che consiste nello scegliere, tra diverse uscite, di inviare il pacchetto all’uscita più vicina in modo da liberarmi al più presto del pacchetto, ossia viene scelto il percorso di costo minimo.

Riepilogando, i problemi di scalabilità e autorità amministrativa vengono risolti con il concetto di sistema autonomo, all’interno del quale tutti i router usano lo stesso protocollo di instradamento interno. La scalabilità viene risolta dato che un router intra-sistema autonomo necessita solo delle informazioni sui router del proprio sistema. Il problema dell’indipendenza amministrativa viene risolto poiché ogni organizzazione può usare qualsiasi protocollo di instradamento intra-sistema autonomo. Infine, per coppie di sistemi connessi bisogna usare lo stesso protocollo di instradamento inter-sistema autonomo quando si scambiano informazioni di raggiungibilità.

**Routing in Internet**

La rete è composta da un insieme di sistemi autonomi tra loro interconnessi. Essi possono essere di diverso tipo in base al ruolo del sistema autonomo all’interno della rete:

-Stub AS: è un AS collegato ad altri sistemi autonomi tramite un altro sistema autonomo che fa da intermediario.

-Multihomed AS: è un AS collegato a più sistemi autonomi.

-Transit AS: fornisce connessione ad altri sistemi autonomi ma non genera traffico né ne è il destinatario.

Ci sono due livelli di routing:

-Intra-sistema autonomo: il cui responsabile è il sistema autonomo stesso, in particolare su quale tipo di algoritmo usare.

-Inter-sistema autonomo: permette di far comunicare tra loro tutti i sistemi autonomi e quello utilizzato è il BGP.

I protocolli intra-sistemi autonomi sono anche detti Interior Gateway Protocols (IGP) e i più usati sono: RIP, OSPF e IGRP.

**RIP**

È un algoritmo Distance Vector, la cui metrica è il numero di hops (numero di sottoreti attraversate lungo il percorso minimo dal router sorgente al router destinazione) ossia tutti i collegamenti hanno costo unitario. In particolare, i costi sono calcolati tra i router sorgente e la sottorete destinazione. Il costo massimo di un percorso è limitato a 15 per cui l’uso di RIP è limitato a sistemi autonomi con diametro^2 inferiore a 15 hop. In RIP, a differenza dei protocolli DV dove i router adiacenti si scambiavano i vettori delle distanze, i router adiacenti si scambiano gli aggiornamenti di routing ogni 30 secondi circa usando un messaggio RIP advertisements, il quale contiene un elenco che comprende la distanza del mittente rispetto ad un massimo di 25 sottoreti.

Un esempio che illustra il funzionamento del RIP:

ho un sistema autonomo con dei router con delle sottoreti, ciascun router mantiene una tabella di instradamento che contiene il vettore delle distanze e la tabella di inoltro. La prima colonna contiene le sottoreti di destinazione, la seconda il router successivo lungo il percorso più beve verso la destinazione e la terza indica il numero di hop per raggiungere la sottorete di destinazione lungo il percorso più breve. Ogni 30 secondi vengono aggiornate le tabelle, perciò, potranno cambiare i valori in base agli advertisements ricevuti. Può succedere che si perdono i messaggi oppure che un router non funzioni più. Per capire se qualcosa non funziona, si aspettano 3 minuti e se non si ricevono advertisements dal vicino vuol dire che quel router o quel link non funziona più, quindi non considero più validi i percorsi per quel nodo e si aggiornano le tabelle.

I messaggi dei RIP hanno un formato standard con un’intestazione e una sequenza di 20 byte per ogni possibile destinazione. 8 bit servono per indicare il tipo di comando, 8 bit per la versione. Ogni destinazione contiene la specifica il tipo di famiglia di indirizzi, l’indirizzo IP della rete e la metrica. La lunghezza massima di un datagramma è di 512 bytes.

**OSPF**

È un protocollo link state che utilizza il flooding di informazioni riguardo lo stato dei collegamenti e l’algoritmo di Dijkstra per trovare il percorso a costo minimo. Un router costruisce un grafo del sistema autonomo e manda in esecuzione dijkstra per trovare un albero di percorsi minimi verso tutte le sottoreti. I costi dei collegamenti vengono impostati a 1 oppure in modo inversamente proporzionale alla capacità del collegamento. Ogni volta che c’è un cambiamento nello stato di un collegamento il router manda informazioni di instradamento via broadcast a tutti i router del sistema autonomo e periodicamente invia lo stato dei collegamenti anche se non ci sono variazioni. I vantaggi di OSPF sono:

-Sicurezza: gli scambi tra router sono autentificati quindi solo router fidati possono far parte di questo protocollo.

-TOS: pacchetti diversi vengono gestiti in modo diverso.

-Percorsi con stesso costo: se più percorsi vanno verso una destinazione con lo stesso costo, OSPF consente di usarli senza doverne scegliere uno.

-Supporto per routing unicast e multicast

-può essere organizzata come una rete gerarchica, ossia in due livelli: livello di area e livello di backbone e gli advertisement sono limitati ad ogni area ossia i router conoscono la topologia solo della loro area. I router di bordo area, ossia quelli che fanno parte sia dell’area sia del livello backbone, hanno il ruolo di riassumere le distanze e annunciarle agli altri router di bordo. L’area di backbone ha il ruolo di instradare il traffico tra le altre aree del sistema.

L’insieme dei messaggi (advertisements) è detto link state database e ogni router ha lo stesso link state database. Ogni volta che invio un messaggio devo ricevere un ACK. A differenza di RIP dove i messaggi viaggiano in UDP, in OSPF i messaggi viaggiano in pacchetti IP e hanno un’intestazione e un corpo con dati specifichi di quel messaggio e advertisement.

**BGP**

È un protocollo di tipo path-vector basato su prefissi e policy. È l’attuale standard de facto dei protocolli di instradamento tra sistemi autonomi, il quale permette di ottenere informazioni di raggiungibilità dei sistemi autonomi vicini,

propagare le informazioni di raggiungibilità a tutti i router interni di un sistema autonomo, determinare delle “buone” rotte verso le sottoreti ed infine di comunicare ad ogni sottorete la propria esistenza.

In BGP, coppie di router si scambiano informazioni di instradamento tramite connessioni TCP semipermanenti, che possono essere instaurate tra due router di diversi sistemi autonomi o dello stesso sistema autonomo. I router ai capi di una connessione TCP sono detti BGP peer e la connessione TCP è detta sessione BGP, la quale se avviene tra router dello stesso sistema autonomo è chiamata sessione BGP interna, mentre se avviene tra due sistemi autonomi è chiamata sessione BGP esterna. BGP permette di conoscere le destinazioni raggiungibili che sono dei prefissi che rappresentano una o più sottoreti.

Le operazioni di BGP sono:

-si stabilisce una connessione tra router sulla porta 179

-si scambiano tutte le rotte che conoscono

-periodicamente si scambiano degli aggiornamenti

Ci sono diversi tipi di messaggi:

-OPEN: viene inizializzata una connessione TCP

-UPDATE: annunciare degli aggiornamenti

-KEEP ALIVE: aggiornare il fatto che una rete è presente

-NOTIFICATION: messaggi di errori

Il protocollo è di tipo incrementale e nel tempo imparerà l’esistenza di certi percorsi verso certe destinazioni. Memorizza i percorsi in una tabella, applica delle policy per selezionarne una, valuta se comunicare le rotte imparate ai vicini. I messaggi scambiati sono messaggi di annuncio (seleziono una nuova rotta e la comunico) o di withdrawal (dico che una rotta non c’è più).

Una rotta è l’insieme di un prefisso destinazione e un insieme di attributi della rotta che includono diverse informazioni tra cui:

-AS path: la sequenza dei numeri di sistemi autonomi che sono coinvolti nel percorso

-Next-Hop: viene usato dai router per configurare le loro tabelle di inoltro in modo corretto. In esso è riportata l’interfaccia del router che inizia l’AS-path. Per scegliere tra diverse rotte di un un sistema autonomo, BGP sceglie l’AS path più breve ossia il percorso che attraversa meno sistemi autonomi. BGP non si limita solo a questo routing ma si basa anche su altre policy. Infatti, quando un router riceve un annuncio di rotta, usa le proprie policy per decidere se accettare o filtrare la rotta e in particolare queste policy possono cancellare una rotta perché magari conosce già una via migliore verso lo stesso prefisso. Quindi se ci sono più rotte, per decidere quale scegliere si applicano regole di eliminazione:

1-guardo la local preference ossia quella che preferisco di più in base a delle policy.

2-a parità di preferenza, scelgo quella migliore ossia quella più corta.

3-a parità di lunghezza, scelgo quella con il next hop più vicino ossia quello con il percorso con costo minore (hot potato routing).

4-altrimenti si usano altri criteri addizionali.

Riguardo le policy, una volta che importo una rotta posso decidere di scartarla o modificarne gli attributi mentre quando ne esporto una posso decidere di scartarla se non la voglio comunicare ai miei vicini o posso modificarne gli attributi.