

## Livello tre (Livello network)

### 5.1 Generalità:

Questo livello è incaricato di muovere i pacchetti dalla sorgente fino alla destinazione, attraversando tanti sistemi intermedi (router) della subnet di comunicazione.

I compiti principali del livello sono:

- conoscere la topologia della rete;
- scegliere di volta in volta il cammino migliore (*routing*);
- gestire il flusso di dati e le congestioni (*flow control e congestion control*);
- gestire i problemi legati alla comunicazione di reti eterogenee (*internetworking*);

#### 5.1.1. Servizi offerti:

In merito ai servizi offerti al livello superiore ci sono due scuole di pensiero, chi sostiene i servizi connection-oriented e chi quelli connection less .

Nel primo caso si sostiene che il livello network debba offrire un servizio sostanzialmente affidabile e orientato alla connessione, in quest'ottica quello che succede è :

- le peer entity stabiliscono una connessione negoziandone i parametri, a questa connessione viene associato un identificatore;
- questo identificatore viene inserito in ciascun pacchetto inviato;
- la comunicazione è bidirezionale;
- il controllo di flusso è fornito automaticamente durante la negoziazione iniziale

Per quanto riguarda le connessioni di tipo connection less, l'idea è quella di dover muovere soltanto i dati, senza preoccuparsi di altro, quindi i passi seguiti da questo tipo di connessioni saranno:

- la sotto rete è giudicata inaffidabile, quindi gli host debbono provvedere per conto proprio alla correzione, individuazione di errori e al controllo di flusso;
- il servizio offerto del livello network deve essere datagram;
- i pacchetti viaggiano indipendentemente gli uni dagli altri, e quindi tutti quanti devono contenere un identificatore della destinazione;

#### 5.1.2. Organizzazione interna della subnet:

Una subnet può essere organizzata secondo due modalità:

- basata su connessioni: in questo caso la subnet stabilisce dei circuiti virtuali sui quali verrà incanalato il traffico di un servizio connection oriented, tutti i router lungo questo cammino memorizzano in apposite strutture la parte del circuito di loro competenza, quando arrivano dei pacchetti che contengono l'id del circuito vengono instradati seguendo le regole memorizzate.
- connectionless: i router si limitano ad instradare ogni pacchetto che arriva sulla base del suo indirizzo di destinazione, decidendo di volta in volta come inoltrarlo, i router mantengono le *tabelle d'instradamento*, nelle quali vengono memorizzate, per ogni possibile destinazione, le linee d'uscita più idonee (questo tipo di tabelle esistono anche nelle subnet basate su connessione, vengono però utilizzate solamente nella fase di setup della connessione), il livello network, quando offre un servizio connection-oriented, fa credere al livello superiore che è in grado di offrire un servizio basato su connessioni, in realtà inoltra i pacchetti indipendentemente gli uni dagli altri.

Entrambe le scelte presentano dei vantaggi e degli svantaggi, in particolare, le subnet basate su connessioni, hanno una minore occupazione della banda trasmissiva, ma un maggior consumo di

risorse sui router, inoltre è presente un ritardo per il setup ma è assente quello per il routing, infine la possibilità di congestione è minore ma la vulnerabilità molto più alta. Per quanto riguarda invece le subnet connectionless, il loro consumo di banda è decisamente superiore (i pacchetti sono più grandi perché mantengono al loro interno l'indirizzo di destinazione), ma il carico sui router è molto più basso, il ritardo per il setup è totalmente assente mentre quello di routing è presente, infine la possibilità di congestione è elevata ma la vulnerabilità è molto bassa.

Una cosa molto importante da tener presente è che la realizzazione interna delle subnet è totalmente indipendente dai servizi offerti.

## 5.2 Algoritmi di routing:

Come abbiamo visto la funzione principale del livello network è quella di instradare i pacchetti della subnet facendogli fare molti *hop (salti)* tra un router ed un altro.

Un *algoritmo di routing* è un software che decide quale linea d'uscita utilizzare per un pacchetto appena ricevuto.

Nelle subnet di tipo datagram gli algoritmi di routing vengono applicati ex novo ad ogni pacchetto in arrivo, mentre nelle subnet di tipo virtual circuit, l'algoritmo viene utilizzato solamente nella fase di setup del circuito virtuale.

Tutti i tipi di algoritmi debbono soddisfare questi requisiti:

1. correttezza: il pacchetto deve essere inoltrato nella giusta direzione
2. semplicità
3. robustezza: l'algoritmo deve poter funzionare anche in caso di cadute di linee e/o di router o in funzione di riconfigurazioni della topologia
4. stabilità: deve convergere e in fretta
5. equità: non deve favorire nessuno
6. ottimalità: deve scegliere la soluzione globalmente migliore

Gli algoritmi di routing si dividono in due gruppi:

- *Algoritmi non adattivi (static routing)*: in questi algoritmi le decisioni sono prese in anticipo, all'avvio della rete e sono comunicate ai router che poi si atterranno scrupolosamente a quelle regole
- *Algoritmi adattivi (dynamic routing)*: in questi algoritmi le decisioni di routing sono riformulate molto spesso.

Un principio molto importante nel routing è il *principio di ottimalità*, il quale afferma che se il router  $j$  è nel cammino ottimo fra  $i$  e  $k$  allora anche il cammino ottimo tra  $j$  e  $k$  è sulla stessa strada, difatti se così non fosse ci sarebbe un altro cammino fra  $j$  e  $k$  migliore di quello che è parte del cammino tra  $i$  e  $k$ , ma allora esisterebbe anche un cammino tra  $i$  e  $k$  migliore di quello ottimo.

Una diretta conseguenza di questo principio è che l'insieme dei cammini ottimi da tutti i router a uno specifico router costituiscono un albero detto *sink tree*.

Analizziamo adesso il diversi tipi di algoritmi:

### 5.2.1. Algoritmi statici:

*5.2.1.1. Shortest path routing*: l'idea è semplice un host di gestione della rete mantiene un grafo che rappresenta la subnet, i nodi del grafo sono i router e gli archi sono le linee punto-punto. All'avvio della rete o in caso di modifiche, viene applicato un algoritmo per la ricerca dei cammini minimi ad esempio quello di Dijkstra, una volta individuato il cammino queste informazioni vengono inviate a tutti i router della subnet.

Per individuare il cammino minimo occorre anche trovare una grandezza da minimizzare, tipicamente la ricerca del minimo viene fatta su queste grandezze:

- 1) numero di hop, cioè di archi

- 2) lunghezza dei collegamenti
- 3) tempo medio di accodamento e trasmissione
- 4) una combinazione di lunghezza, banda trasmissiva , traffico medio ecc.

5.2.1.2. *Flooding*: questa tecnica consiste nell'inviare ogni pacchetto su tutte le linee eccetto quella da cui è arrivato. In linea teorica questa tecnica potrebbe essere anche utilizzata come algoritmo di routing, ma presenta un grosso inconveniente, difatti genera una quantità enorme di traffico spesso inutile.

Esistono tuttavia delle tecniche per limitare questo traffico, come l'inserimento di un contatore all'interno di ciascun pacchetto, tale contatore viene decrementato ad ogni hop e quando arriva a zero il pacchetto viene scartato, come valore iniziale è utile inserire il diametro della subnet.

Un altro sistema per limitare il traffico consiste nell'inserire la coppia (*source router ID*, *sequence number*) in ogni pacchetto, ogni router tiene traccia, memorizzando questi dati, dei pacchetti in transito se vede per la seconda volta lo stesso pacchetto lo scarta. Infine l'ultima tecnica utile, è il *selective flooding*, nella quale i pacchetti vengono duplicati solo sulle linee che vanno all'incirca nella giusta direzione.

Il flooding come algoritmo di routing è difficilmente utilizzabile, risulta però particolarmente utile in ambiti come quello militare, nel quale l'elevata ridondanza delle informazioni garantisce la massima affidabilità e robustezza, è inoltre utile quando si richiede l'aggiornamento contemporaneo di informazioni distribuite, ed infine è utile come strumento di paragone con altri algoritmi, visto che trova sempre, tra gli altri, il cammino minimo.

#### 5.2.2. *Algoritmi dinamici*:

Nelle moderne reti si usano algoritmi di tipo dinamico, che si adattano automaticamente ai cambiamenti della rete. Questi algoritmi non sono eseguiti solamente all'avvio della rete, ma rimangono in esecuzione sui router durante il loro normale funzionamento.

5.2.2.1. *Distance Vector*: ogni router mantiene una tabella che memorizza un elemento per ogni router raggiungibile, questo elemento della tabella contiene la distanza (numero di hop) che lo separa dal router in oggetto e la linea d'uscita da usare per arrivarci.

Ciascun router stima per i suoi vicini immediati, la distanza dei collegamenti corrispondenti, mandando degli speciali pacchetti ECHO e misurando quanto tempo ci mette la risposta a tornare. Ad intervalli regolari, ogni router manda la sua tabella a tutti i vicini, e riceve le loro. Quando un router riceve delle nuove informazioni, calcola una nuova tabella scegliendo tra tutte la concatenazione migliore: *se stesso -> vicino immediato -> router remoto di destinazione* per ogni destinazione.

Un grande problema di cui soffrono gli algoritmi di tipo distance vector è il *count to infinity*, quando un collegamento va giù, prima che l'informazione si propaghi tra tutti i router della subnet può trascorrere anche molto tempo, difatti quello che tipicamente succede è che:

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>router</i>
*-----	-----*-----	-----*-----	-----*-----	-----*	<b>collegamenti</b>
	1	2	3	4	<b>distanze da A</b>
Se cade la linea da a a b dopo uno scambio di informazioni					
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>router</i>
*	*-----	-----*-----	-----*-----	-----*	<b>collegamenti</b>
	3	2	3	4	<b>distanze da A</b>
questo avviene perché B non ricevendo risposta da A crede di poterci arrivare tramite C, col proseguire degli scambi si ha che					
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>router</i>
*	*-----	-----*-----	-----*-----	-----*	<b>collegamenti</b>
	3	4	3	4	<b>distanze da A (dopo 2 scambi)</b>
	5	4	5	4	<b>distanze da A (dopo 3 scambi)</b>
	5	6	5	6	<b>distanze da A (dopo 4 scambi)</b>

Una possibile soluzione al problema del count to infinity (se le distanze rappresentano il numero di hop) è limitare il numero massimo di hop al diametro della subnet.

Il distance vector era l'algoritmo di arpanet e tuttoggi è ancora usato con il nome di *RIP* (*Routing Internet Protocol*).

5.2.2.2. *Link State Routing*: A causa della lentezza di convergenza del distance vector si è cercato un approccio diverso, il *link state routing*.

L'idea di fondo è che ogni router tenga sotto controllo i collegamenti tra lui e i suoi vicini diretti, e distribuisca queste informazioni a tutti gli altri. Sulla base di tali informazioni ogni router ricostruisce localmente la topologia completa dell'intera rete e calcola il cammino minimo tra se e gli altri.

I passi seguiti da questo algoritmo sono:

- 1) scoprire i suoi vicini
- 2) misurare il costo delle relative linee
- 3) costruire un pacchetto con queste informazioni
- 4) inviare il pacchetto agli altri router
- 5) costruire la topologia della rete dopo aver ricevuto i pacchetti dai suoi vicini
- 6) calcolare il cammino minimo verso tutti gli altri router

Per determinare la configurazione dell'intera rete un router quando viene connesso invia un pacchetto *HELLO* su tutte le sue linee d'uscita, i suoi vicini a questo punto risponderanno inviandogli ciascuno il proprio indirizzo IP, inoltre inviando vari pacchetti *ECHO* e misurando il tempo di arrivo della risposta (diviso 2) e facendo la media tra tutti i pacchetti

invia, il router riesce anche a ricostruire il ritardo della linea.

Si costruisce quindi un pacchetto con l'identità del mittente, il numero di sequenza del pacchetto, l'età del pacchetto e la lista dei vicini con i relativi ritardi, questo tipo di pacchetto viene inviato ad intervalli regolari e comunque ogni qualvolta avvenga un evento significativo (es: caduta di linea).

La tecnica generalmente utilizzata per inviare questi pacchetti è il flooding, inserendo però nei pacchetti la coppia routerID e sequence number per eliminare i duplicati; tutti i pacchetti vengono confermati, e per evitare che un pacchetto vaghi all'infinito nella rete, tutti quanti riportano la loro età, che ad ogni passaggio di router viene decrementata fino allo 0 quando il pacchetto viene scartato.

Ogni router attraverso queste informazioni riesce così a ricostruire il grafo della subnet e ne calcola il cammino minimo.

I principali algoritmi di questo tipo sono *OSPF* (Open Shortest Path First), sicuramente il più diffuso, e *IS-IS* (Intermediate System- Intermediate System), inizialmente progettato per DECNet e poi adottato da OSI.

**5.2.2.3. Routing Gerarchico:** Quando la rete diventa molto grande diventa quasi impossibile mantenere l'intera topologia nella memoria di ogni router, per poter risolvere questo problema il processo di routing è stato suddiviso in modo gerarchico.

La rete viene quindi suddivisa in zone (*regioni*), all'interno di ogni zona vale quanto visto finora, ossia ogni router conosce i router di tutta la regione.

Quando però un router interno deve spedire dati ad un router di un'altra zona, sa soltanto che per farlo dovrà prima spedire i dati ad un router della sua regione incaricato di questo tipo di comunicazioni, questo tipo di router è detto *router di confine*.

Il router di confine si occuperà di trasmettere le informazioni ad un router di confine di un'altra regione.

Per concludere, i router interni mantengono nelle loro tabelle le informazioni per tutti i router interni alla loro regione e un'entrata per ogni altra regione raggiungibile, con il relativo router di confine.

### 5.3 Controllo della congestione:

Quando troppi pacchetti si trovano nella stessa parte di subnet, si verifica una congestione che degrada le prestazioni della subnet. Questo avviene in particolare quando i router non sono in grado di gestire la coda dei pacchetti in arrivo, in questo caso i pacchetti cominciano a essere scartati con la conseguenza che verranno ritrasmessi aggravando così la situazione di congestione.

Il controllo della congestione è un problema di tutta la rete ed è diverso dal controllo di flusso, gli approcci possibili sono:

- 1) open loop: senza contro reazione
- 2) closed loop: con contro reazione

Nel primo caso si cerca di evitare la congestione, ma se questa si verifica non viene fatto nulla per recuperarla, mentre nel secondo caso la rete viene controllata e in caso di congestione si cerca di porvi rimedio

5.3.1. *Traffic shaping*: Questo è un approccio di tipo open loop, l'idea è quella di forzare la trasmissione dei pacchetti ad un ritmo piuttosto regolare, esistono tre tecniche per implementare il traffic shaping:

- *leaky bucket*
- *token bucket*
- *flow specification*

L'algoritmo *Leaky Bucket* (secchio che perde) trova un'analogia in un secchio riempito d'acqua da un rubinetto regolabile, con un buco sul fondo dal quale riversa l'acqua che contiene a ritmo costante. In pratica sull'host si realizza un leaky bucket, che è autorizzato a riversare ad un determinato data rate i pacchetti che mantiene nel suo buffer, se l'host genera più pacchetti di quanti ne possa contenere nei buffer i pacchetti cominciano a perdersi, quindi finché l'host genera traffico anche bursty ma sotto il data rate di uscita non ci sono problemi.

L'algoritmo *Token Bucket* (secchio di gettoni) in questa tecnica si mantiene una certa irregolarità anche nel flusso di dati in uscita, praticamente quando un host non trasmette accumula un credito trasmissivo fino al massimo stabilito, quando poi deve trasmettere consuma tutto il suo credito alla massima velocità consentita dalla linea. Il secchio contiene gettoni che si accumulano ad intervalli di tempo prestabiliti (es 1 ogni millisecondo) fino ad un max di M, per poter trasmettere deve avere almeno un gettone, se nel secchio ci sono k token e l'host deve trasmettere h > k pacchetti, i primi k sono inviati gli altri rimarranno in attesa dei nuovi token. Una importante differenza con il leaky bucket è che in questa tecnica i pacchetti non vengono mai scartati, perché il secchio contiene gettoni, quindi se il secchio si riempie si informa il livello superiore di non inviare più pacchetti.

L'ultimo algoritmo è il *Flow Specification*, in questo caso tutti (sorgente subnet e destinazione) si accordano sul traffic shaping. Per ottenere tale accordo le varie parti debbono specificare che tipo di traffico andranno ad inviare (datarate, grado di burstiness) e la qualità del servizio (ritardo massimo, frazione di pacchetti che si può perdere), questo accordo si chiama appunto flow specification.

5.3.2. *Choke packet*: Questo è un approccio di tipo closed loop, è previsto un router che mantiene sotto controllo il grado di utilizzo delle linee d'uscita. Il router misura l'utilizzo istantaneo U di ciascuna linea, e memorizza entro una media esponenziale M gli eventi trascorsi:

$M_{nuovo} = aM_{vecchio} + (1-a)U$  dove il parametro a (compreso tra 0 e 1) è il peso dato alla storia passata, e (1-a) è il peso dato all'informazione più recente. Quando una linea d'uscita si avvicina ad una soglia critica prefissata, il router esamina i pacchetti in ingresso per vedere se sono destinati alla linea d'uscita che è in allarme.

In caso affermativo, invia all'host d'origine del pacchetto un *choke packet* per avvertirlo di diminuire il flusso.

Quando l'host riceve il choke packet diminuisce il flusso e ignora i successivi choke packet per un tempo prefissato.

Trascorso questo tempo, l'host si rimette in attesa di altri choke packet, se ne arrivano altri riduce ancora il flusso, altrimenti riprende le trasmissioni.

5.3.3. *Hop-by-hop choke packet*: Un problema della tecnica precedente, è la lentezza di reazione, perché l'host che manda i pacchetti ci mette un certo tempo a ricevere i choke packet e a diminuire il flusso. Un metodo più efficiente è quello di costringere ogni router a diminuire immediatamente il flusso non appena riceve un choke packet, in questo caso si parla di *hop-by-hop choke packet*.

5.4 Il livello network in Internet:

Internet è una collezione di AS connessi gli uni agli altri, le componenti principali sono:

- 1) backbone principale (linee ad alta velocità)
- 2) reti regionali (USA)
- 3) reti nazionali (Europa e resto del mondo)
- 4) reti locali

Il protocollo che tiene tutto insieme è il protocollo di livello network dell'architettura TCP/IP cioè IP.

IP è un protocollo di tipo datagram quindi non connesso e non affidabile, che opera come segue:

- a) riceve i dati di livello transport e li incapsula in pacchetti di dimensione massima a 64Kbyte (normalmente circa 1500)
- b) instrada i pacchetti sulla subnet eventualmente frammentandoli lungo il viaggio
- c) a destinazione:
  - a) riassembla i frammenti
  - b) estrae da questi i dati del livello transport
  - c) consegna al livello transport i dati nell'ordine in cui sono arrivati

5.4.1. Formato di un pacchetto IP:

Un pacchetto IP è costituito da un *header* e da una parte dati, l'header ha una parte fissa di 20 byte ed una parte opzionale di lunghezza variabile, i campi dell'header sono i seguenti:

32 bit

Version	IHL	Type of service	Total Length		
Identification			D	M	Fragment offset
			F	F	
Time to Live		Protocol	Header Checksum		
Source address					
Destination address					
Options					

Le funzioni di ciascun campo sono le seguenti:

Version	numero di versione del protocollo
IHL	lunghezza dell'header in parole di 32 bit (minimo 5 max 15)
Type of service	Ignorato dai router caratterizza affidabilità e velocità
Total length	lunghezza del pacchetto (inclusa la parte dati) max 65.535 byte
Identification	Identifica i frammenti di uno stesso pacchetto che avranno tutti lo stesso codice
DF	don't fragment (se uguale ad 1 non si deve per alcun motivo frammentare il pacchetto)
MF	more fragments (se uguale ad 1 il pacchetto non è ancora finito)
Fragment offset	indice del frammento di pacchetto

Time to live	contatore inizializzato a 255 che viene decrementato ad ogni hop (o ad ogni sec) quando arriva a zero il pacchetto viene scartato
Protocol	codice del protocollo di livello transport cui consegnare i dati
Header checksum	checksum di controllo del solo header: <ul style="list-style-type: none"> <li>• si sommano le parole a 16 bit (in complemento a 1) dell'header</li> <li>• si complementa a 1 il risultato</li> <li>• viene ricalcolato ad ogni hop</li> </ul>
Source e destination address	indirizzi di mittente e destinatario
Option	sono definite 5 opzioni possibili: <ol style="list-style-type: none"> <li>(1) security: quando è segreto il pacchetto</li> <li>(2) strict source routing: cammino da seguire</li> <li>(3) loose source routing: lista di router da non mancare</li> <li>(4) record route: ogni router appende il suo indirizzo</li> <li>(5) timestamp: ogni router appende il suo indirizzo più un timestamp</li> </ol>

5.4.2.. Indirizzi IP:

Un indirizzo IP è costituito da 32bit che codificano due informazioni: *network number* cioè il numero assegnato alla rete IP su cui si trova l'elaboratore e l'*host number* cioè il numero assegnato all'elaboratore.

La combinazione di questi due numeri è unica, quindi all'interno di una rete non possono esserci due elaboratori con lo stesso numero e non possono neanche esserci due reti con lo stesso numero.

Ogni host su una rete in realtà non possiede un suo indirizzo, difatti l'host number viene assegnato alla sua interfaccia di rete, quindi se un host avesse X interfacce, questo comporterebbe che quel determinato host sarebbe raggiungibile con X indirizzi ip.

Gli indirizzi IP sono assegnati da autorità nazionali i *NIC* Network Information Center che sono coordinate a livello mondiale.

Esistono 5 classi di indirizzi IP ciascuna in grado di coprire un numero sempre maggiore di elaboratori e sono:

	<i>8 bit</i>		<i>8 bit</i>		<i>8 bit</i>		<i>8 bit</i>	
<b>A</b>	0	Network		Host				
<b>B</b>	1	0	Network			Host		
<b>C</b>	1	1	0	Network			Host	
<b>D</b>	1	1	1	0	Multicast address (indirizzo di gruppo)			
<b>E</b>	1	1	1	1	0	Riservato per uso futuro		

Appartenenti a queste classi ci sono anche indirizzi di tipo particolare come l'indirizzo di loopback

utilizzato particolarmente per i test delle reti difatti, i pacchetti che utilizzano questo indirizzo non vengono inviati sulla rete ma vengono elaborati come fossero in entrata. Tutti gli indirizzi IP sono solitamente espressi nella forma dotted decimal notation ossia le varie parti dell'indirizzo vengono divise da un punto.

#### 5.4.3. Routing IP:

I collegamenti tra router avvengono direttamente, ma attraverso una network, di fatto quindi la loro connessione avviene tramite le due interfacce di rete e la linea di connessione che le collega.

Ogni router possiede al suo interno delle tabelle che contengono elementi del tipo:

- a) (this network number, host number) per ciascun host a cui il router è connesso;
  - b) (network number, 0) per ciascuna network lontana di cui il router conosce l'esistenza;
- Associate a questi elementi ci sono le informazioni sull'interfaccia di rete da utilizzare per inoltrare i pacchetti verso quella destinazione.

Viene inoltre mantenuto l'indirizzo del default router a cui inviare tutti i pacchetti destinati a network sconosciute.

#### 5.4.4. Subnetting:

Con la suddivisione in classi degli indirizzi IP si è arrivati presto a capire che con alcune di esse lo spreco di indirizzi era eccessivo per l'utilizzo che ne veniva fatto, si è quindi introdotto il sistema di subnetting, grazie al quale una network può essere suddivisa in molte subnet ciascuna contenente i suoi host.

Il meccanismo è molto semplice, viene introdotta una maschera detta subnet mask con la quale si specifica quanti bit dell'indirizzo originale sono destinati alla network e quanti alle subnet, in pratica quello che viene fatto è considerare gli indirizzi dei singoli host come una coppia di valori (subnet.host). A seconda dello spazio che viene riservato al subnet number si possono ottenere molte subnet con pochi host o poche subnet con molti host.

Nei router questa configurazione porta all'aggiunta di elementi del tipo:

- a) (this network number, this subnet number, host number) per ogni host connesso al router;
- b) (this network number, subnet number, 0) per ogni subnet conosciuta ma non direttamente connessa;

Con le relative informazioni circa l'interfaccia da utilizzare.

#### 5.4.5. CIDR (Classless Inter Domain Routing):

Nonostante l'introduzione del meccanismo di subnetting lo spreco di indirizzi che avviene con l'uso di classi del tipo A o B è veramente notevole, è quindi stato introdotto un ulteriore sistema per la riduzione del numero di indirizzi inutilizzati.

Questo nuovo sistema è il sistema CIDR con il quale le classi non sono più chiuse, e quindi il numero di bit che identificano il network number può variare da 1 bit a 31 (anche se valori estremi hanno poco senso), in questo modo è possibile utilizzare il numero di indirizzi strettamente necessario. Ovviamente potendo l'indirizzo variare di lunghezza occorre accompagnarlo con una maschera simile a quella delle subnet nella quale si specifica quanti bit compongono il network number, altrimenti l'informazione non sarebbe mai interpretabile.

Questo sistema se da un lato comporta uno spreco minimale degli indirizzi IP dall'altro lato però, obbliga i router a compiere un lavoro più gravoso, difatti, non solo debbono mantenere in memoria i network number con le relative informazioni circa la network mask, ma si trovano anche ad affrontare situazioni nelle quali ad un singolo network number corrispondano due o più entrate nelle tabelle del router.

Questa situazione è dovuta al fatto che come nelle subnet anche i network number possono essere suddivisi, quindi quando una compagnia richiede al NIC un network number gliene viene

assegnato uno ad esempio del tipo 184.13.152.0/22 nel quale i primi 22 bit sono destinati al network number e gli ultimi 10 agli host. A questo punto la compagnia può decidere di rivendere parte degli indirizzi che le competono suddividendoli in network number più piccoli ad esempio due da 512 host del tipo 184.13.152.0/23 e 184.13.154.0/23 in questa situazione l'indirizzo originale e l'indirizzo della prima sotto network sono uguali a meno del numero di bit che li identificano. In queste situazioni se consideriamo l'entrata delle tabelle di routing relativa al network number con lunghezza minore, ci stiamo riferendo alla rete più grande, altrimenti a quella più piccola.

I router di fronte a queste entrate multiple hanno due possibilità se entrambe le entrate hanno associata la stessa linea d'uscita allora il router le fonde e mantiene quella con il network number di lunghezza minore, altrimenti ogni volta deve decidere quale delle due ha il network number che è il più lungo prefisso dell'indirizzo da instradare, quindi in pratica ogni volta sceglie la strada che porta alla più piccola delle sotto reti che contenga l'host di destinazione.

#### 5.4.6. Protocolli di controllo:

Insieme ad IP esistono diversi altri protocolli che servono per il controllo del funzionamento della subnet, e sono:

a) ICMP: (Internet Control Message Protocol), questo protocollo è utilizzato dai router per scambiarsi le informazioni di controllo sull'operatività della subnet, i messaggi più comuni sono *destination unreachable*, *time exceeded*, *redirect* (utilizzato quando un router pensa che il pacchetto gli sia arrivato per errore perché ad esempio un host mobile si è spostato), *echo request*, *reply* (utile per sapere se una destinazione è viva e raggiungibile), *timestamp request*, *reply* (come prima solo che registra anche gli istanti di partenza e arrivo, quindi utile per misurare le prestazioni di rete).

b) ARP: (Address Resolution Protocol), è il protocollo che dall'indirizzo IP di destinazione trova l'indirizzo di livello data link per l'incapsulamento, opera poggiandosi direttamente sul livello data link e non su IP, i passi che segue sono i seguenti: invia in broadcast una richiesta del tipo chi ha indirizzo IP 124.133.122.1?, solo l'host che ha quell'indirizzo risponderà inviandogli il suo indirizzo data link, quando l'host che aveva fatto la richiesta riceve la risposta la mantiene in memoria per circa 15 min. Se l'indirizzo è esterno alla LAN allora ci sono due modi o si configura un indirizzo di default che sarà quello del router a cui mandar i pacchetti IP per le altre reti, o il pacchetto ARP viene inviato al router che risponderà con il proprio indirizzo data link e che farà da intermediario tra sorgente e destinazione.

c) RARP: (Reverse Address Resolution Protocol), questo è il protocollo inverso rispetto ad ARP difatti consente di trovare l'indirizzo IP associato ad un particolare indirizzo data link, particolarmente utile nel caso di host senza disco che caricano le informazioni di avvio dalla rete.

#### 5.4.7. Protocolli di routing:

Internet è una collezione di AS connessi tra loro da backbone ad alta velocità. Ciascun AS come abbiamo già detto è caratterizzato dal fatto di essere controllato da una singola autorità il routing complessivo è quindi organizzato in modo gerarchico, ossia all'interno di un AS si usa un solo Interior Gateway Protocol (IGP), mentre tra gli AS si usa un Exterior Gateway Protocol (EGP).

Tra i protocolli di tipo IGP troviamo *RIP (Routing Information Protocol)* che è di tipo distance vector ma ormai sostituito da un altro protocollo di questa famiglia *OSPF (Open Shortest Path First)* che è invece di tipo link state.

OSPF consente un routing gerarchico all'interno dell'AS che viene in questo modo suddiviso in diverse aree; i router di un AS possono essere:

*interni ad un area*: si occupano del routing interno

*sul confine dell'area*: si occupano del routing tra aree

*nell'area backbone*: si occupano del routing su backbone

*al confine dell'AS*: sono quelli che si occupano del routing tra AS utilizzando protocolli di tipo EGP.

Proprio di tipo EGP è il protocollo *BGP (Border Gateway Protocol)* che è di tipo distance vector, e le sue caratteristiche principali sono che da la possibilità di gestire manualmente le politiche di instradamento, mantiene e scambia con altri router non solo il costo per raggiungere le altre destinazioni ma anche il cammino completo. In questo modo si risolve il problema del count to infinity.

#### 5.4.8. IPv6:

Studiato per migliorare le prestazioni di Ipv4 le principali differenze sono:

- a) indirizzi di 16 byte quindi  $7 * 10^{23}$  indirizzi per metro quadrato del nostro pianeta;
- b) header semplificato 8 campi contro 23;
- c) funzioni di autenticazione e privacy basate su crittografia;
- d) gestione della qualità del servizio attraverso il campo flow label che consente di istituire connessioni con caratteristiche negoziate in anticipo