Politecnico di Bari

Facoltà di Ingegneria Elettronica

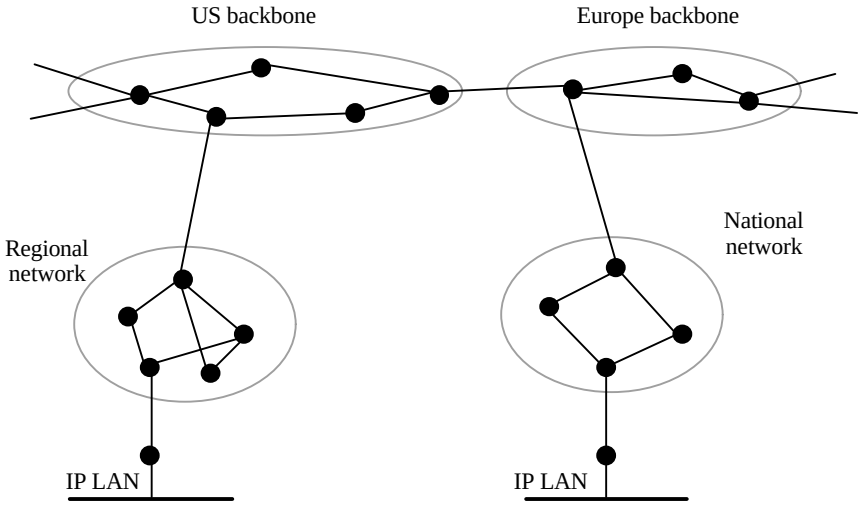
Corso di *Reti di Telecomunicazioni*

Internet Protocol versione 4

Generalità sulle reti IP

La più nota rete di calcolatori basata sull'architettura TCP/IP è la **rete Internet**. *Internet è una collezione di Autonomous System connessi gli uni con gli altri*; non esiste una struttura rigida, ma comunque si possono distinguere alcune componenti di base:

- **backbone principali** (linee ad alta velocità);
- **reti regionali** (USA);
- **reti nazionali** (Europa e resto del mondo);
- **reti locali**.



Schematizzazione della "struttura" di Internet

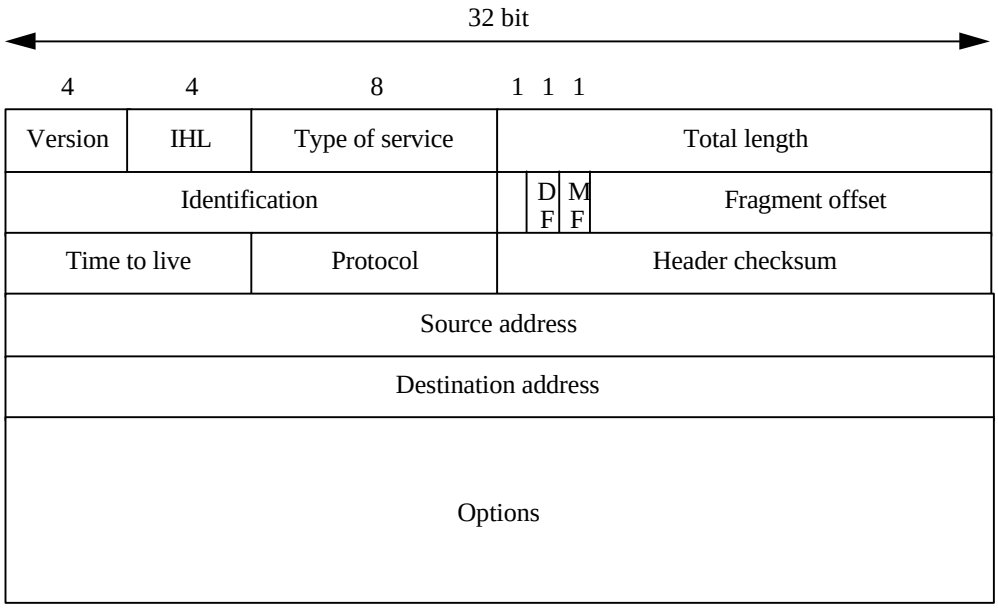
Ciò che tiene tutto insieme è il protocollo di livello network dell'architettura TCP/IP e cioè **IP** (**Internet Protocol**). Si tratta di un **protocollo datagram**, quindi non connesso e non affidabile, che opera come segue:

- in trasmissione:
  - riceve i dati dal livello transport e li incapsula in pacchetti di dimensione massima pari a **64 kbyte** (la dimensione media tradizionale è in realtà di circa 1500 byte);
  - instrada i pacchetti sulla subnet, eventualmente *frammentandoli* lungo il viaggio;

- in ricezione:
  - riassembla (se necessario) i *frammenti* in pacchetti;
  - estrae da questi i *dati del livello transport*;
  - consegna al livello transport i dati nell'ordine in cui sono arrivati (che non è necessariamente quello in cui sono partiti).

Lo header IP versione 4

Un **pacchetto IPv4** (definito nella RFC 791) è costituito da un **header** e da una **parte dati**. L' header ha una parte fissa di **20 byte** e una parte, opzionale, di lunghezza variabile. La figura seguente illustra la struttura dell' **header IP**:



Header IPv4

I pacchetti IP vengono trasmessi da sinistra a destra (e ovviamente dall'alto verso il basso), secondo il cosiddetto schema **big endian**; il primo bit è il bit più significativo del campo *Version*.

Andiamo a vedere con sufficiente dettaglio le funzioni dei vari campi dell' header:

- il campo ***Version*** descrive la versione del protocollo utilizzato; tramite questo campo è possibile fare in modo che il passaggio tra diverse versioni di IP duri mesi o addirittura anni, con alcuni router che usano ancora le vecchie versioni e altri che invece possiedono solo l'ultima versione rilasciata (**IPv6**);
- dato che la lunghezza dell' header non è costante, in esso è presente un campo (denominato **IHL**) che contiene la lunghezza dell'header stesso, espressa in termini di *parole da 32 bit*; il valore minimo è 5 (ossia l' header è lungo 5·32=160 bit = 20 byte) ed è quello che si ha quando non è presente alcuna *opzione* sul pacchetto; in presenza invece di una o più opzioni, la lunghezza cresce fino ad un massimo di 15 (dato che IHL è costituito da 4 bit): in questo caso, la lunghezza complessiva dell' header è 15·32=480 bit = **60 byte** e quella della parte destinata alle opzioni è **40 byte** (cioè i 60 totali meno i 20 fissi). C'è da sottolineare che, per determinate opzioni, la lunghezza di 40 byte è insufficiente: ad esempio, non è sufficiente per l'opzione che consentirebbe di memorizzare il cammino percorso da un pacchetto dalla sorgente alla destinazione. Di conseguenza, il campo delle opzioni diventa praticamente inutile nella pratica;
- il campo **Type of Service** consente agli host di comunicare alla rete il *tipo di servizio*; esistono in proposito una serie di combinazioni di *affidabilità* e *velocità*: ad esempio, per la voce digitalizzata si richiede la massima velocità di trasmissione, mentre invece per il trasferimento file si richiede soprattutto una trasmissione priva di errori. In teoria, questo campo (composto a sua volta da vari sottocampi) consentirebbe ai router di compiere delle scelte (ad esempio tra un canale satellitare ad alta capacità ma anche con molto ritardo ed una linea affittata a bassa capacità e a basso ritardo); in pratica, tuttavia, i router attuali ignorano completamente tale campo;
- il campo **Total Length** contiene la lunghezza l'intero pacchetto IP (header+dati); il valore massimo di 65535 byte. Questo limite è attualmente tollerabile, ma in futuro, con le nuove reti ad alta velocità, è presumibile sia necessario avere pacchetti IP più lunghi;

- il campo **Identification** consente all' host destinazione di determinare a quale pacchetto appartiene un dato frammento, in quanto tutti i frammenti di uno stesso pacchetto hanno lo stesso valore di questo campo ([1]);
- dopo il campo *Identification*, c'è un campo da 1 bit che rimane inutilizzato e poi due campi ciascuno da 1 bit:
  - il campo ***DF*** (*Don't Fragment*) ordina ai router, quando è =1, di non frammentare i pacchetti poiché la destinazione non è capace di rimettere insieme i pezzi; quindi, ponendo DF=1, il mittente sa che il pacchetto verrà consegnato tutto intero;
  - il campo **MF** (*More Fragments*) serve ad indicare se un pacchetto, diviso in più frammenti, è terminato (MF=0) oppure non lo è ancora (MF=1); è ovvio che tutti i frammenti tranne l'ultimo hanno MF=1;
- il campo successivo è **Fragment offset** (*scostamento del frammento*) e indica in quale posizione del pacchetto corrente si trova il frammento attuale: in pratica, questo campo consente di ricostruire il pacchetto originale a partire dai singoli frammenti in cui è stato scomposto, rispettando l'ordine corretto. Tutti i frammenti, tranne l'ultimo, devono essere multipli di **8 byte** (*unità elementare di frammentazione*). Dato che il campo Fragment offset è lungo 13 bit, è possibile avere un massimo di **8192** frammenti per pacchetto;
- il campo **Time to live** è un contatore (inizializzato al valore 255) che viene decrementato di uno a ogni *salto* (*hop*); in pratica, quindi, esso conta i "salti" compiuti dal pacchetto da un router all'alto; quando il contatore arriva a zero, il pacchetto viene scartato (in quanto si presume che sia una copia non necessaria) e contemporaneamente viene inviato alla sorgente un *pacchetto di avvertimento*. In questo modo, si evita che un pacchetto resti nella rete per sempre (ad esempio a causa di errori nelle tabelle di routing di uno o più router);

[1] Apriamo una breve parentesi sulla **segmentazione**: nel caso ideale, un intero datagramma IP viene incapsulato in un frame fisico, per rendere efficiente la trasmissione attraverso una rete fisica reale; tuttavia, ogni rete locale consente una dimensione massima (indicata con *Maximum Transfer Unit*, **MTU**) per i pacchetti che la attraversano ed è quindi ovvio che, siccome i datagrammi IP devono essere incapsulati in un frame, non possono prescindere da queste dimensioni per altro fortemente variabili a seconda del tipo di rete. Allora per tutelarsi dalla disomogeneità delle reti, invece di strutturare il datagramma in modo eccessivamente aderente ai vincoli fisici dei vari prodotti, si è deciso di scegliere una dimensione dei datagrammi IP conveniente ed escogitare poi un metodo (detto **segmentation and reassembly**) di dividerli in piccoli pezzi, detti **frammenti** , che possano essere accettati da una rete con qualsivoglia piccolo MTU e, ovviamente, riassemblati in uscita.

- quando il livello di rete ha assemblato un pacchetto IP completo, ha bisogno di sapere cosa deve farne: a tal fine, il campo **Protocol** indica a quale *processo* del livello di trasporto i dati devono essere consegnati. Le varie possibilità sono *TCP*, *UDP* e altro. La numerazione dei protocolli è globale in tutta Internet ed è definita nell' **RFC 1700**;
- il campo successivo è l' **Header checksum**, che serve a verificare solamente l'header, per prevenirsi da errori generati da problemi di memoria all'interno dei router. Da notare che il valore di questo campo va ricalcolato ad ogni salto, dato che cambia sempre almeno il campo *Time to live*. Il funzionamento dell'algoritmo è il seguente:
  - si sommano (in complemento ad uno) le parole a 16 bit dell' header, considerando il checksum a zero;
  - si complementa ad uno il risultato;
  - viene ricalcolato ad ogni hop (time to live cambia);
- i campi **Source address** e **Destination Address** contengono evidentemente gli indirizzi dell' *host sorgente* e dell' *host destinazione*; del formato di tali indirizzi parleremo tra poco;
- infine, l'ultimo campo è il campo **Options**, progettato per fornire informazioni non presenti nel progetto iniziale e per evitare di allocare bit dell'header per informazioni che sono raramente richieste. Le opzioni sono di lunghezza variabile; ognuna di esse inizia con un *codice* di 1 byte che la identifica e, opzionalmente, con un campo che indica la *lunghezza* dell'opzione stessa; poi c'è il campo che contiene l'opzione vera e propria. Il campo Options viene completato a multipli di 4 byte. Attualmente, sono definite solo 5 opzioni:
  - ***security***: descrive il grado di segretezza del pacchetto ma praticamente viene sempre ignorato dai router;
  - ***strict source routing***: fornisce l'intero percorso da seguire dalla sorgente alla destinazione, in forma di *sequenza di indirizzi IP*; esso viene usato spesso dagli amministratori di sistema per spedire *pacchetti di emergenza* quando le tabelle dei router sono corrotte o quando si vogliono effettuare misure di tempo;

- ***loose source routing***: indica una lista di router attraverso i quali si vuole che il pacchetto passi necessariamente, fermo restando che, durante il percorso, può incontrarne anche altri;
- ***record route***: impone ad ogni router di *appendere* il suo indirizzo IP, in modo da ricostruire in ricezione l'intero percorso seguito;
- ***timestamp***: ogni router appende il suo indirizzo più un timestamp (timbro temporale) di 32 bit; il contenuto di questo campo, insieme a quello del campo precedente, consente un efficace debug degli algoritmi di routing.

Concetti generali sugli indirizzi IPv4

Un **indirizzo IP versione 4** ([2]) corrisponde ad una *sequenza di 32 bit*, che per semplicità vengono raggruppati in 4 gruppi da 8 bit ciascuno, in modo che ciascun gruppo corrisponda ad un numero decimale compreso tra 0 e 255: ad esempio, l'indirizzo 192.168.150.10 corrisponde, in binario, a

11000000 10101000 10010110 00001010

Ogni indirizzo IP può essere visto come una coppia di due numeri: il **numero di rete** (*network number*) e il **numero di host** (*host number*) all'interno di quella rete.

Il numero di bit usato per il *numero di rete* non è fisso, ma dipende dalla cosiddetta **classe di indirizzo**. Esistono infatti cinque classi di indirizzi IP:

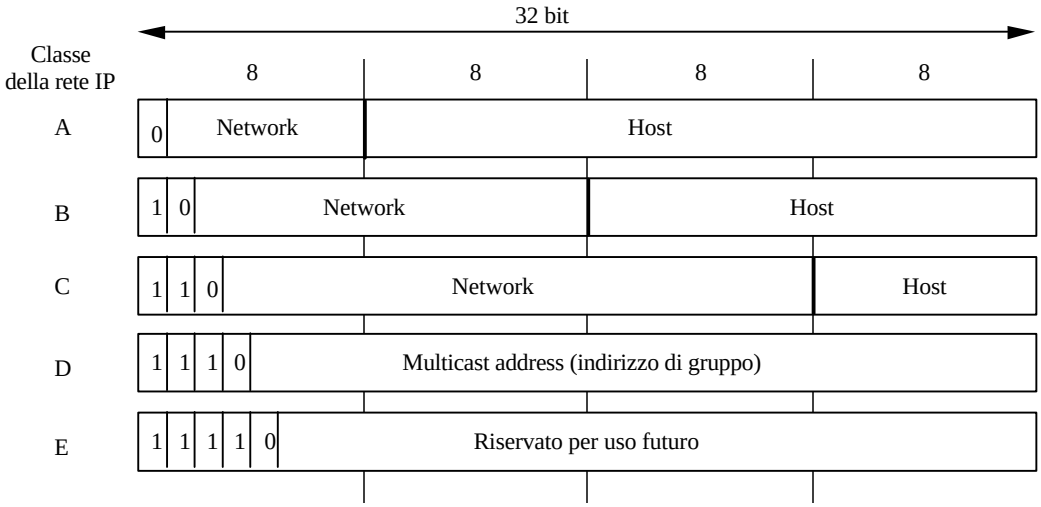
- **Classe A**: gli indirizzi di questa classe iniziano tutti con un bit a 0 (il che significa che il primo ottetto ha un valore decimale compreso tra 1 e 126); i successivi 7 bit corrispondono al numero di rete, mentre invece i rimanenti 24 bit corrispondono al numero di host. Allora, è possibile avere 126 reti di classe A con 16777213 host ciascuna ([3]);
- **Classe B**: gli indirizzi di questa classe hanno i primi due bit a 10 (per cui il primo ottetto ha un valore decimale compreso tra 128 e 191); i successivi

[2] Ormai da diverso tempo è stata standardizzata la versione 6 del protocollo IP e dei relativi indirizzi, dato che gli indirizzi IP versione 4 vanno ormai decisamente verso l'esaurimento. In questa sede si ritiene comunque utile parlare degli indirizzi della versione 4, essendo questi quelli maggiormente diffusi.

[3] Si tenga conto che 126 corrisponde a 2^7-2. Il motivo per cui, dalle 128 possibilità di numeri di rete, si tolgono 2 possibilità è che non sono ammessi i valori 0 e 255.

14 bit sono per il network number, mentre i rimanenti 16 per l'host number. Si hanno così 16382 reti di classe B, di 65534 host ciascuna;

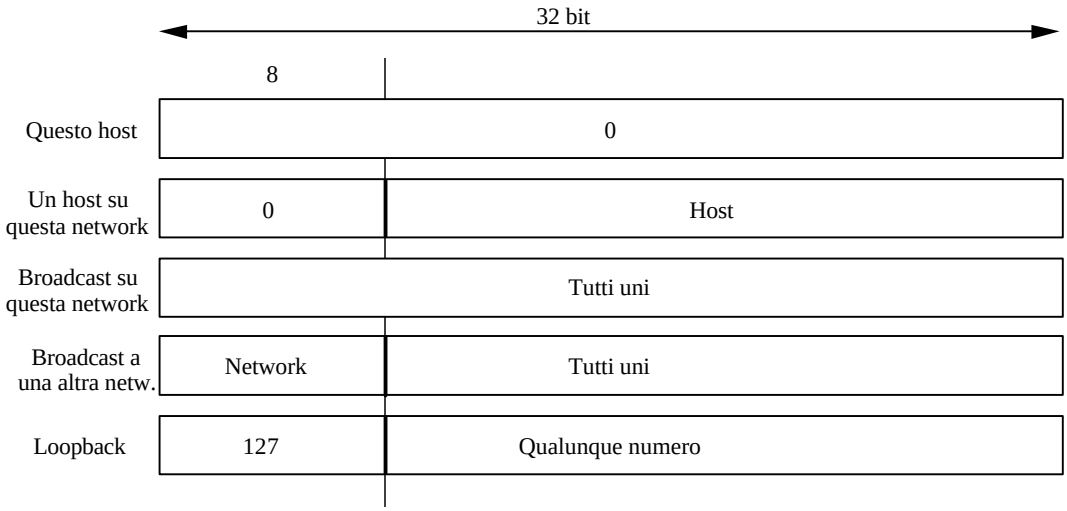
- **Classe C**: questi indirizzi hanno i primi 3 bit posti a 110 (per cui il primo ottetto varia, in decimale, tra 192 e 223), dopodiché i successivi 21 bit sono per il network number e solo gli ultimi 8 bit sono per l'host number. Si possono così avere 2097150 reti di classe C, di 254 host ciascuna;
- **Classe D**: questi indirizzi iniziano con quattro bit posti a 1110 (per cui il primo ottetto va da 224 a 239 in decimale) e sono riservati per il cosiddetto multicasting, ossia per fare in modo che un pacchetto IP inviato da una sorgente venga recapitato, anziché ad una sola destinazione (unicasting), a varie destinazioni (almeno 2);
- **Classe E**: quest'ultima classe di indirizzi prevede i primi quattro bit posti a 1111 (primo ottetto da 240 a 254 in decimale) ma non è ancora utilizzata nella pratica.



Configurazione binaria delle varie classi di indirizzi IP

Il numero di rete viene assegnato da un ente centrale, l'**InterNIC**; al contrario, il numero di host è deciso dal possessore di quel numero di rete.

Esistono altri indirizzi con un significato speciale, di seguito riassunti:



Indirizzi "speciali"

Per esempio, quando un pacchetto contiene un indirizzo IP di destinazione fatto da tutti 1, significa che il pacchetto deve essere recapitato a tutti i nodi della rete alla quale appartiene la sorgente del pacchetto stesso.

Inoltre, quando il numero di host è fatto solo da 0 mentre invece il numero di rete è non nullo, allora l'indirizzo nel suo complesso rappresenta evidentemente l'indirizzo di rete. Ad esempio, l'indirizzo 125.0.0.0 è quello di una rete di classe A. Al contrario, quando è il numero di rete ad essere composto da tutti 0, l'indirizzo complessivo identifica un preciso nodo della rete in questione.

Il concetto di "subnet"

Dato che la suddivisione per *classi* è piuttosto grezza, è stato creato il concetto di **subnet** (*sottorete*): esso permette di "sottrarre" qualche bit del numero dell'host al fine di ottenere una maggiore flessibilità di configurazione rete (ad esempio per separare il traffico in rete tramite un router), invisibile però fuori dalla rete.

In pratica, usando il concetto di subnet, il generico indirizzo IP risulta composto da 3 elementi: un **numero di rete** (secondo la classificazione esposta nel precedente paragrafo), un **numero di subnet** (che identifica appunto una sottorete all'interno della rete in questione) e un **numero di host** (che identifica un dato nodo all'interno della sottorete).

Dato l'indirizzo IP di un nodo di una rete che è stata a sua volta suddivisa in sottoreti, esiste un modo immediato per ricavare il numero di rete ed il numero della sottorete cui il nodo appartiene: basta fare un AND, bit a bit, tra l'indirizzo IP

e la cosiddetta **subnet mask**. Quest'ultima è una maschera (cioè una sequenza di bit, sempre 32) di bit, i cui valori sono impostati per default per le prime tre classi:

- *Classe A*: **255.0.0.0**, pari a 11111111.00000000.00000000.00000000
- *Classe B*: **255.255.0.0**, pari a 11111111.11111111.00000000.00000000
- *Classe C*: **255.255.255.0**, pari a 11111111.11111111.11111111.00000000

E' abbastanza evidente che queste maschere standard, applicate (tramite l'AND) ad un indirizzo IP, hanno il semplice effetto di restituire il numero di rete: ad esempio, per un indirizzo di classe B, sappiamo che i primi 16 bit danno il numero di rete ed è proprio tale numero che ricaviamo facendo l'AND tra un indirizzo di classe B e la maschera standard per tale classe.

Quindi, con le maschere standard non si ottiene alcuna suddivisione in sottoreti. Supponiamo, invece, data per esempio una rete di classe C, di volerla suddividere in 8 sottoreti: possiamo "rubare" tre bit al quarto ottetto (l'inizio del numero di host), ottenendo una Subnet mask fatta nel modo seguente:

11111111.11111111.11111111.11100000 (255.255.255.224)

Ad esempio, se l'indirizzo della rete 193.1.1.0 ([4]), gli host delle varie sottoreti avranno indirizzi che iniziano per:

11000001.00000001.00000001.00000001.00000000 (193.1.1.0)	1° subnet
11000001.00000001.00000001.00000001.00100000 (193.1.1.32)	2° subnet
11000001.00000001.00000001.00000001.01000000 (193.1.1.64)	3° subnet
11000001.00000001.00000001.00000001.01100000 (193.1.1.96)	4° subnet
11000001.00000001.00000001.00000001.10000000 (193.1.1.128)	5° subnet
11000001.00000001.00000001.00000001.10100000 (193.1.1.160)	6° subnet
11000001.00000001.00000001.00000001.11000000 (193.1.1.192)	7° subnet
11000001.00000001.00000001.00000001.11100000 (193.1.1.224)	8° subnet

[4] Si noti l'ultimo byte posto a 0

In realtà, è importante tener presente che le subnet 1 (=000) e 8 (=111) non possono essere usate, in quanto i numeri fatti di soli 0 o soli 1 hanno le funzioni particolari di cui parlavamo prima. Di conseguenza, nell'esempio appena descritto possiamo suddividere la nostra rete di classe C in 6 sottoreti, ciascuna da 32 host (data la disponibilità dei 5 bit finali di ciascun indirizzo).

Autore: **Sandro Petrizzelli**
e-mail: sandry@iol.it
sito personale: http://users.iol.it/sandry