APPUNTI
DI
RETI DI CALCOLATORI

IL MODELLO ISO / OSI

L' Open Systems Interconnection (OSI) è uno standard per reti di calcolatori stabilito nel 1978 dall'International Organization for Standardization (ISO), il principale ente di standardizzazione internazionale, che stabilisce per l'architettura logica di rete una struttura a strati composta da una pila di protocolli di comunicazione suddivisa in 7 livelli, i quali insieme espletano in maniera logico-gerarchica tutte le funzionalità della rete.

Il modello ISO/OSI, concepito per reti di telecomunicazioni a commutazione di pacchetto, è costituito da una pila (o stack) di protocolli attraverso i quali viene ridotta la complessità implementativa di un sistema di comunicazione per il networking. In particolare ISO/OSI è costituito da 7 strati (o livelli), i cosiddetti layer, che definiscono e racchiudono in sé a livello logico uno o più aspetti fra loro correlati della comunicazione fra due nodi di una rete.
ISO/OSI realizza una comunicazione per livelli, cioè:

- Il livello n del nodo A può comunicare solo con il livello n del nodo B, ma non con gli altri.
- Il livello n di un nodo offre servizi al livelli n + 1 utilizzando i servizi offerti dal livello n – 1.

Sicché ISO/OSI incapsula i messaggi di livello n in messaggi del livello n-1. Così se A deve inviare, ad esempio, una e-mail a B, l'applicazione (liv. 7) di A propagherà il messaggio usando il layer sottostante (liv. 6) che a sua volta userà il SAP del layer inferiore, fino ad arrivare alla comunicazione ovvero alla trasmissione sul canale o mezzo fisico trasmissivo.
Tutto ciò conferisce modularità al sistema con maggiore semplicità di progettazione e gestione della rete. Sara quindi possibile ad esempio, modificare, sostituire i protocolli di un livello senza modificare le funzioni degli altri livelli.

Elenco e funzioni dei livelli

Livello 1: fisico (Physical Layer)
L'obiettivo di questo livello è di trasmettere un flusso di bit (quindi dati non strutturati) attraverso un mezzo fisico, definendo :

- Le tensioni scelte per rappresentare i valori logici dei bit tramessi;
- Bit Time (durata in microsecondi del segnale che identifica un bit);
- La modulazione e la codifica utilizzata;
- L'eventuale trasmissione simultanea in due direzioni (duplex);
- La forma e la meccanica dei connettori usati per collegare l'hardware al mezzo trasmissivo.

Livello 2: collegamento (Datalink Layer)
Consente il trasferimento affidabile di dati attraverso il livello fisico. Invia *frame* di dati

- sincronizzandoli
- effettuando un controllo degli errori
- controllando le perdite di segnale
- controllando il flusso dei dati

Tutto ciò fa apparire al livello superiore come se il mezzo trasmissivo fosse esente da errori.
I pacchetti provenienti dal terzo livello vengono *incapsulati*, cioè gli vengono aggiunti un nuovo header (intestazione) e un tail (coda), a formare quindi un nuovo pacchetto da inviare poi al livello fisico.
Per ogni frame ricevuto il destinatario deve inviare una conferma (ACK), se il frame non arriva o arriva con errori, la trasmissione deve essere ripetuta. I pacchetti ACK possono anche essere raggruppati e mandati in blocchi.
Il controllo del flusso dei dati consiste nell'adeguare la velocità delle macchine nel caso una sia più veloce dell'altra, minimizzando così le perdite dovute a sovraccarico sul destinatario.

Livello 3: rete (Network Layer)
L'obiettivo del Network Layer è rendere i livelli superiori indipendenti dai meccanismi e dalle tecnologie di trasmissione usate per la connessione e prendersi carico della consegna a destinazione dei pacchetti.
Le sue funzioni sono:

- routing: scelta ottimale del percorso di rete da utilizzare per garantire la consegna delle informazioni dal mittente al destinatario, scelta svolta dal router attraverso dei particolari algoritmi di Routing e tabelle di routing.
- conversione dei dati nel passaggio fra due reti con diverse caratteristiche. Deve, quindi:
  - ❖ tradurre gli indirizzi di rete;
  - ❖ frammentare, se necessario, i pacchetti dati se la nuova rete ha una diversa Maximum Transmission Unit (MTU);
  - ❖ valutare la necessità di gestire diversi protocolli attraverso l'impiego di gateway.

Livello 4: trasporto (Transport Layer)
Questo livello permette un trasferimento di dati trasparente e affidabile tra due host.
Si occupa di:

- aprire, mantenere e chiudere le connessioni
- frammentare e riassemblare i messaggi (segmenti)
- rilevare e correggere gli errori
- controllare il flusso e le congestioni (troppi pacchetti allo stesso router)
- gestire connessioni multiple all'interno dello stesso elaboratore

A differenza dei livelli precedenti, che si occupano di connessioni tra nodi contigui di una rete, il Trasporto (a livello logico) si occupa solo del punto di partenza e di quello finale.

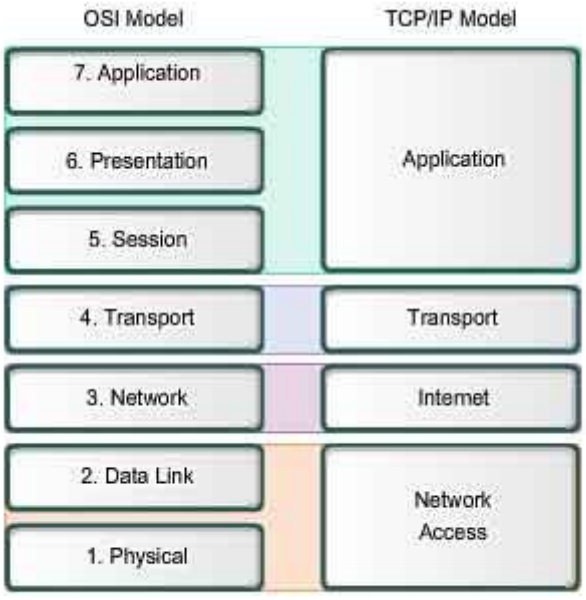
Livello 5: sessione (Session Layer)
Controlla la comunicazione tra applicazioni. Instaurare, mantenere ed abbattere connessioni tra applicazioni cooperanti. Si occupa anche della sincronia di invio/ricezione messaggi.
Esso consente di aggiungere, ai servizi forniti dal livello di trasporto, servizi più avanzati, quali la gestione del dialogo (mono o bidirezionale), la gestione del token (per effettuare mutua esclusione) o la sincronizzazione (inserendo dei checkpoint in modo da ridurre la quantità di dati da ritrasmettere in caso di gravi malfunzionamenti).

Si occupa anche di inserire dei punti di controllo nel flusso dati: in caso di errori nell'invio dei pacchetti, la comunicazione riprende dall'ultimo punto di controllo andato a buon fine.

Livello 6: presentazione (Presentation Layer)
Trasforma i dati forniti dalle applicazioni in un formato standardizzato e offrire servizi di comunicazione comuni, come la *crittografia*, la *compressione* del testo e la *riformattazione*.

Livello 7: applicazione (Application Layer)
Obiettivo di questo livello è interfacciare utente e macchina.
Fornisce un insieme di protocolli che operano a stretto contatto con le applicazioni. È errato identificare un'applicazione utente come parte del livello applicazione. I protocolli delle applicazioni tipiche di questo livello realizzano operazioni come ad esempio:
Trasferimento file, Terminale virtuale, Posta elettronica

Confronto con il TCP / IP
ISO/OSI è stato progettato per permettere la comunicazione in reti a "commutazione di pacchetto", del tutto simili al paradigma TCP-UDP/IP usato in Unix e nella rete ARPAnet, poi divenuta Internet. Il modello ISO-OSI non è rigido: costituisce, piuttosto, un punto di riferimento per le architetture di rete a pacchetto, che possono distanziarsi più o meno da esso. La differenza sostanziale fra TCP/IP e ISO/OSI consiste nel fatto che nel TCP/IP i livelli di presentazione e di sessione sono esterni alla pila di protocolli. I livelli sono dunque solo quattro:
Applicazione,
Trasporto,
Livello di rete (internetworking),
Livello di collegamento (data link).
I livelli Sessione e Presentazione sono assenti perché implementati (eventualmente) altrove, cioè nell'applicazione stand-alone esterna.



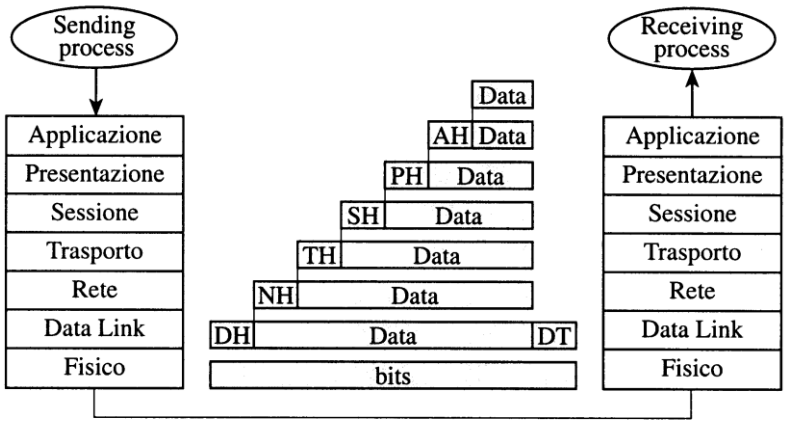
The key parallels are in the Transport and Network layers.

Incapsulamento dei Pacchetti

La trasmissione dei dati avviene :
- attraverso una serie di passaggi da livelli superiori a livelli inferiori nel sistema che trasmette,

- poi attraverso il mezzo fisico di comunicazione,
- infine attraverso un'altra serie di passaggi, questa volta dai livelli inferiori a livelli superiori del ricevente.

Notare come a livello 2, sia necessario aggiungere in coda un campo che identifica la fine del pacchetto prima di passare lostesso al livello che utilizza il mezzo trasmissivo.

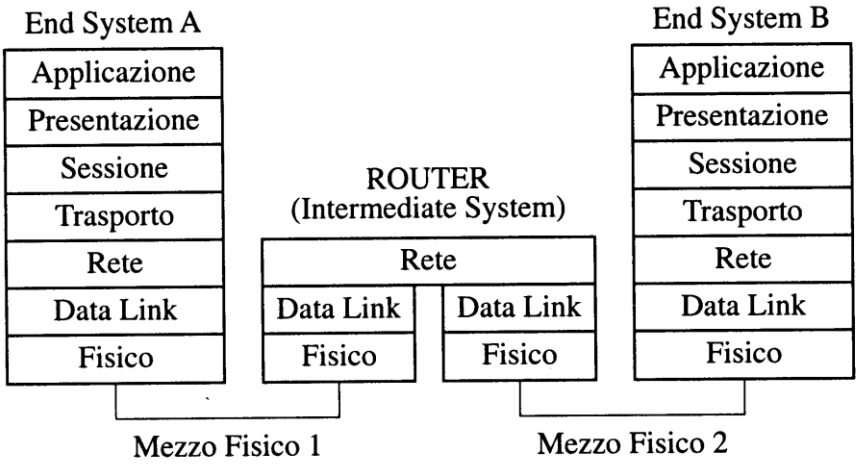


Sistemi Intermedi

Non sempre lo scambio delle informazioni avviene direttamente tra i due sistemi finali che contengono le applicazioni (ES: End Systems). Può anche implicare l'attraversamento di alcuni sistemi intermedi (IS: Intermediate Systems). In questi Intermediate Systems esistono delle entità che assumono la funzione di Router (in senso esteso), ovvero entità che instradano le informazioni. Tali entità possono essere collocate a diversi livelli del modello OSI, ed allora gli Intermediate Systems assumono nomi diversi a seconda del livello in cui avviene l'instradamento dei dati. Si parla allora di

- repeater o hub a livello 1,
- switch o bridge a livello 2,
- router a livello 3

Qui di seguito è rappresentata la collocazione di un Router nel modello OSI.



Protocolli connessi e non connessi

Per tutti i livelli superiori al livello fisico sono definite due modalità operative: una **modalità connessa** e una **modalità non connessa.**

Un dato livello può fornire al livello superiore servizi di tipo connesso, non-connesso o entrambi. Questa è una scelta progettuale che varia per ogni livello, da architettura ad architettura.
In un servizio non connesso la spedizione di un pacchetto è simile alla spedizione di una lettera ordinaria con il sistema postale. Tutto avviene in una sola fase lasciando cadere la lettera nella buca delle lettere. La lettera deve contenere sulla busta l'indirizzo completo del destinatario. Non vi è alcun riscontro diretto che la lettera giunga a destinazione correttamente.
In un servizio connesso lo scambio di dati tramite pacchetti ricorda le frasi scambiate tra due interlocutori al telefono. Vi sono tre momenti principali:

- creazione della connessione (il comporre il numero telefonico e il "pronto" alla risposta);
- trasferimento dei dati (la conversazione telefonica);
- chiusura della connessione (posare il microtelefono).

Mezzi trasmissivi e rilevamento degli errori

Uno degli aspetti più importanti di una rete è costituito dal tipo e dalle caratteristiche della linea fisica utilizzata per il collegamento degli host. La scelta del mezzo trasmissivo dipende dalle prestazioni che si vogliono ottenere, da poche centinaia di bps, a miliardi di bps. E' quindi utile essere a conoscenza delle caratteristiche fisiche ed elettriche di ogni mezzo trasmissivo. Il cavo che assicura le prestazioni migliori ha bassi valori di impedenza (*dove perimpedenza intendiamo la resistenza che offre un oggetto al passaggio di un segnale a corrente alternata*) e deve essere il più possibile indeformabile quando sottoposto a trazione durante la posa per evitare il deterioramento delle sue qualità trasmissive.
Sia il trasmettitore che il ricevitore devono adattarsi al valore di impedenza del mezzo trasmissivo per ottimizzare la trasmissione dati, cioè per aumentare il più possibile la potenza ricevuta / potenza trasmessa. Inoltre l'impedenza deve essere invariante rispetto alla frequenza di utilizzo, o avere un range di oscillazione molto limitato (adattati in impedenza).
Abbiamo tre classi di mezzi trasmissivi:

- **elettrici** doppino, cavo coassiale
- **ottici** fibra ottica
- **onde radio** trasmissioni etere (wireless)

Anche se negli anni più recenti sono andati affermandosi nuovi e più vantaggiosi supporti trasmissivi, il mezzo a tutt'oggi più diffuso è il filo di rame, utilizzato nella stragrande maggioranza dei collegamenti telefonici, sotto forma di coassiale grosso e sottile, o doppino.
Le onde elettromagnetiche sono utilizzate in situazioni particolari, ad esempio per permettere ad un utente di potersi spostare liberamente con il suo elaboratore all'interno della struttura che ospita la LAN, senza però perdere o sospendere la sua connessione. I mezzi ottici, ossia fibre ottiche e laser, hanno la proprietà di permettere collegamenti alle velocità di trasferimento più elevate, e di essere relativamente insensibili ai disturbi elettromagnetici. Per questo motivo sono utilizzate per cablare delle parti di LAN che sono sottoposte a inquinamento elettromagnetico notevole.

Doppino
Il doppino è costituito da una coppia di conduttori in rame intrecciati tra loro (twisted pair) e ricoperti da una guaina di materiale plastico allo scopo di ridurre i disturbi elettromagnetici.
Mentre nei decenni passati il doppino telefonico costituiva il mezzo trasmissivo dell'intero percorso tra i due interlocutori di una comunicazione telefonica, anche a lunga distanza, ora il mezzo trasmissivo utilizzato tra le centrali telefoniche è la fibra ottica e il doppino è il mezzo trasmissivo per quello che viene chiamato ultimo miglio, cioè per il collegamento tra l'utente e la centrale telefonica più vicina. Il doppino è nato per il trasporto della voce umana che ha una banda di frequenza molto ridotta (compresa tra 300 per i toni gravi e 3400 Hz per i toni acuti). Con la diffusione delle reti di computer si è avuta l'esigenza di una banda passante sempre maggiore e con le moderne

tecnologie trasmissive i doppini adesso sono in grado di supportare frequenze molto elevate, tali da permettere velocità trasmissive superiori a 1 Gbps.
Le tipologie di doppino sono 3 :

- **UTP** (Unshielded Twisted Pair), ossia non schermato
- **STP** (Shielded Twisted Pair) schermato coppia per coppia
- **FTP** (Foiled Twisted Pair), un solo schermo per tutto il cavetto.

I cavi sono classificati in sette categorie :

Categoria 1	**Telecommunication:** cavi per la telefonia analogica
Categoria 2	**Low Speed Data:** Cavi per trasmissione dati a bassa velocità
Categoria 3	**High Speed Data:** cavi che supportare una velocità di 10 Mb/sec, per soddisfare lo standard 10baseT
Categoria 4	**Low Loss/High Performance Data:** cavi per trasmissione dati fino a 16 Mb/sec
Categoria 5	**Low Loss/Extended Frequency/High Performance Data:** cavi per fino a 100 Mb/sec.
Categoria 5e	cavi per trasmissione dati fino a **200 Mb/sec**.
Categoria 6	cavi per trasmissione dati fino a **1Gb/sec**.
Categoria 6e	cavi per trasmissione dati in reti Ethernet fino a 10 Gb/sec.
Categoria 7	cavi per trasmissione dati fino a 10 **Gb/sec**.

Cavo coassiale
Il cavo coassiale era molto usato prima dell'avvento del doppino. Oggi si preferisce usare i doppini per medie prestazioni e le fibre ottiche per alte prestazioni. Il cavo coassiale è formato da due conduttori concentrici, isolati reciprocamente e ricoperti da materiali protettivi. A fronte di un costo nettamente superiore rispetto a quello del doppino, per una trasmissione attraverso cavo coassiale sono richiesti un minor numero di ripetitori di segnale e soprattutto si può disporre di una ampiezza di banda di circa 20 Mhz. I dati digitali sono molto soggetti al rumore e alle distorsioni di segnale che vengono introdotte quando i segnali viaggiano su grandi distanze. A causa di questo fatto le reti che usano come mezzo trasmissivo il cavo coassiale possono estendersi solo per distanze limitate a meno che non vengano utilizzati dei ripetitori di segnale che rigenerano il segnale periodicamente (repeater). Gli svantaggi di installare e mantenere un sistema in cavo coassiale includono il fatto che il cavo è difficile e costoso da fabbricare, è difficile da utilizzare in spazi confinati, in quanto non può essere piegato troppo intorno ad angoli stretti, ed è soggetto a frequenti rotture meccaniche ai connettori.

Fibra ottica

La trasmissione mediante fibra ottica è una delle soluzioni più utilizzate per la creazione di linee dorsali ad alto traffico, che necessitano di una grossa ampiezza di banda. La fibra invece che trasmettere lungo un filo di rame dei segnali elettrici, fa "viaggiare", attraverso una minuscola "galleria", singoli impulsi di luce.

La fibra ottica garantisce un **livello di attenuazione** molto basso (0,05%, ovvero 2 db/km). I continui miglioramenti del mezzo trasmissivo, hanno ad oggi consentito di effettuare una trasmissione fino a 100 km di distanza senza necessità di amplificare il segnale.

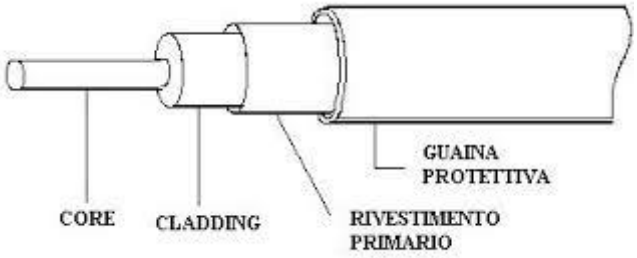
La fibra ottica presenta notevoli **vantaggi**:

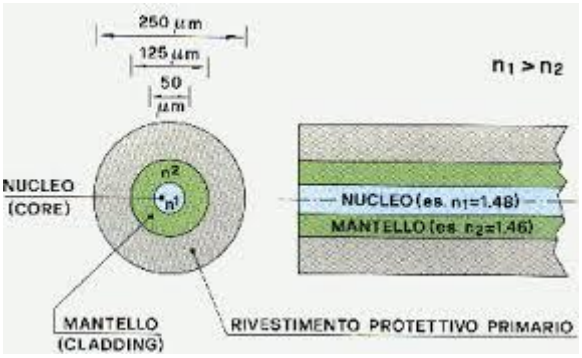
- la totale **immunità dai disturbi** elettromagnetici. Non è infatti costituita da materiale conduttore;
- **larga banda** di utilizzo. Si usa per trasmissioni dati ad alta velocità fino a 2 Gb/sec;
- **bassa attenuazione e diafonia assente** (disturbi fra cavi che viaggiano in parallelo);
- l' intercettazione del segnale è impossibile

A fronte di tutti questi vantaggi, c'è comunque qualche svantaggio.

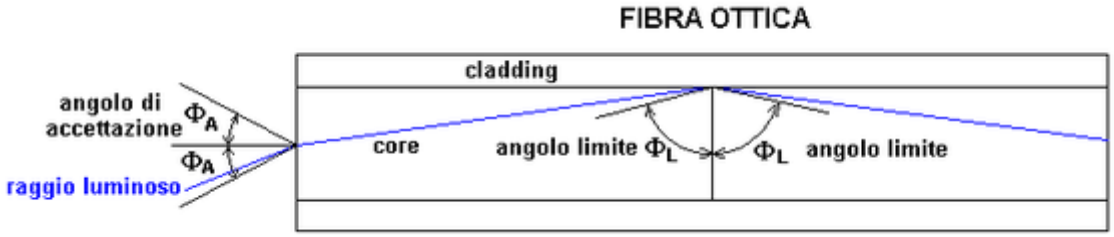
- costi, soprattutto d'installazione;
- fragilità della fibra, unita alla "impossibilità" della sua riparazione.

Un cavo in fibra ottica è costituito dal core, dal cladding, da un rivestimento primario e dalla guaina protettiva; il core è il nucleo, il cladding è il mantello. Hanno due indici di rifrazione diversi, il primo è maggiore del secondo, affinché la luce rimanga confinata all'interno del core.





Il cavo in fibra ottica consiste infatti di una parte centrale in vetro circondata da parecchi strati di materiali protettivi. Questo cavo trasmette luce anziché segnali elettrici, eliminando così il problema dell'interferenza elettrica (questo lo rende il mezzo trasmissivo ideale in ambienti che hanno un'elevata interferenza elettrica).
La fisica delle fibre ottiche è l'ottica geometrica. Molto importante è l'angolo rispetto l'asse del cavo con cui i raggi luminosi vengono indirizzati all'interno del core. Esiste infatti un angolo massimo di incidenza, detto angolo critico, al di sotto del quale i raggi vengono totalmente riflessi dal cladding e rimangono, quindi, all'interno del core.



Se la sorgente luminosa è puntiforme e se presenta raggi di differente lunghezza d'onda (per es. una sorgente che emette nell'infrarosso), i raggi percorrono cammini diversi e, a parità di tempo trascorso, si avrà uno sfasamento dei raggi. Le fibre ottiche che consentono a più raggi di entrare sono dette **multimodali** ed hanno una dimensione di 50/125 o 62.5/125 micron.
Fibra ottica multimodale
Questi problemi si eliminano del tutto se si utilizzano fibre ottiche **monomodali** .
Si chiamano fibre ottiche monomodali le fibre il cui core è sottile per permettere l'entrata di un solo raggio luminoso proveniente, però, non da un LED come le fibre ottiche multimodali precedenti, ma da un LASER. La dimensione tipica di una fibra ottica monomodale e` di 10/125 micron.

Wireless LAN
In questo tipo di reti i segnali si propagano nell'etere sotto forma di onde elettromagnetiche. Sono dette wireless. Le LAN di tipo wireless per far comunicare i computer usano segnali radio ad alta frequenza o raggi di luce infrarossa. Ogni computer deve avere un dispositivo che permette di spedire e ricevere i dati. Le reti wireless sono adatte per consentire a computer portatili o a computer remoti di connettersi alla LAN. Sono inoltre utili negli edifici più vecchi dove può essere difficoltoso o impossibile installare i cavi. Le reti wireless hanno però alcuni svantaggi: garantiscono **poca sicurezza**, sono suscettibili all'**interferenza elettrica** della luce e delle onde radio e sono più **lente** delle LAN che utilizzano la cablatura.

Velocità di trasmissione dei dati
Un elemento molto importante nella scelta e nella valutazione dei sistemi trasmissivi è la velocità di trasmissione. Per misurare la quantità di informazione trasmessa nel tempo viene utilizzata l'unità di misura bps (bit per secondo) che *misura il numero di bit trasmessi in un secondo,* esclusi eventuali

bit di servizio e di sincronismo trasmessi contestualmente, è cioè la quantità di informazione effettivamente trasmessa.
Il segnale viaggia all'interno del messo trasmissivo con una certa **frequenza** (ricordiamo che la frequenza è una grandezza che concerne fenomeni periodici o processi ripetitivi, in informatica è la misura espressa in Hertz del numero di volte al secondo in cui un segnale si ripete).
E' proprio il canale trasmissivo l'elemento di una rete che pone le restrizioni maggiori per quanto riguarda la velocità di trasmissione dei dati, infatti, in caso di distanza superiore ad un certo valore, è necessario inserire sulla linea un ripetitore del segnale.
Nella tabella si evidenziano le corrispondenze tra i mezzi trasmissivi e la massima distanza fisica tra due host senza necessità di rigenerare il segnale.

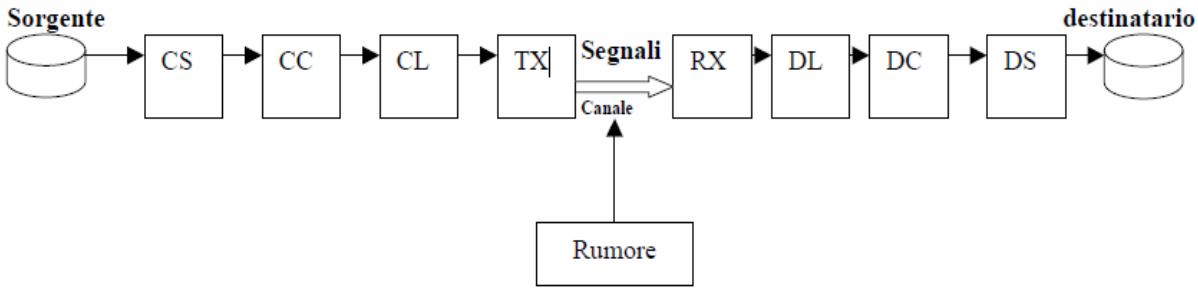
Mezzo trasmissivo	**Larghezza di banda**	**Massima distanza fisica**
Cavo coassiale	10 - 100Mbps	185 mt
UTP	100 Mbps – 1 Gbps	100 mt
Fibra ottica multimodale	100 Mbps – 1 Gbps	2000 mt
Fibra ottica monomodale	1 Gbps – 10 Gbps	3000 mt
wireless	11 – 54 Gbps	100 – 500 mt

Di seguito una tabella con la denominazione della rete il relazione al mezzo trasmissivo utilizza con con le distanze massime raggiungibili:

Cabling Standard	***Medium***	***Maximum Distance***
100BASE-FX	Mutli-mode fiber Single-mode fiber	Half duplex: 400 meters; full duplex: 2 km Full duplex: 10 km
100BASE-SX	Multi-mode fiber	550 meters
100BASE-BX	Single-mode fiber	40 km
100BASE-LX10	Single-mode fiber	10 km
1000BASE-SX	Multi-mode fiber	550 meters
1000BASE-LX	Multi-mode fiber	550 meters
1000BASE-LX	Single-mode fiber	5 km
1000BASE-LX10	Single-mode fiber	10 km
1000BASE-ZX	Single-mode fiber	Up to 70 km
1000BASE-BX10	Single-mode fiber	10 km
10GBASE-SR	Multi-mode fiber	26–82 meters
10GBASE-LR	Single-mode fiber	10–25 km
10GBASE-LRM	Multi-mode fiber	220 meters
10GBASE-ER	Single-mode fiber	40 km

Codifica dei dati

IL processo di comunicazione può essere rappresentato secondo un modello detto **Modello di SHANNON** rappresentato in figura



Quando si deve inviare un messaggio, occorre decidere come questo deve essere rappresentato, se ad esempio è un testo occorre decidere in che lingua scriverlo. Si deve quindi scegliere un CODICE. Chi trasferisce dovrà utilizzate un **codificatore di sorgente (CS)**, chi riceve dovrà utilizzare un **decodificatore di sorgente (DS).** Un possibile esempio di codice sorgente è il codice ASCII. Il sostanza ad ogni carattere viene sostituito una determinata configurazione di simboli, nel nostro caso 0 e 1.

La trasmissione del messaggio avviene attraverso un mezzo fisico di trasmissione detto anche canale. Il canale è disturbato da rumori, interferenze, distorsioni, che possono alterare il messaggio. Per fare in modo che il destinatario riconosca che il messaggio che ha ricevuto non è quello che è stato inviato e quindi fargli richiedere una ritrasmissione, al messaggio viene aggiunto con ulteriori codici (**codifica di canale – CC**) da parte del sorgente. Tali codici vengono interpretati dal ricevente tramite una **decodifica di canale (DC).** In sintesi la codifica di canale deve garantire la correttezza della trasmissione e ciò viene realizzato aggiungendo al messaggio originale un certo numero di bit che verranno utilizzati dal ricevitore per verificare la correttezza del messaggio. Esempi di calcolo di questi bit aggiunti sono codice di parità e il CRC (Cyclic Redundancy cheching).
Per ridurre i tempi di trasmissione o per aumentare la capacità di memorizzazione dei dischi rigidi, si utilizzano **codici di compressione**, che consistono nella riduzione del numero di bit necessari per rappresentare la stessa quantità di informazione. Si basano su studi statistici relativi alla frequenza con cui i vari simboli si presentano (es. il carattere a si presenta più frequentemente del carattere q, quindi viene codificato con un numero più basso di bit). Esempi :

- Codice di Huffman

Il messaggio per poter essere trasferito su un mezzo fisico, deve essere trasformato da simboli a segnali compatibili con il mezzo trasmissivo considerato:

- in segnali elettrici, se il mezzo è un cavo elettrico;
- in onde elettromagnetiche, se il mezzo di trasmissione è l'aria;
- in segnali ottici, se il mezzo è una fibra ottica.

La trasformazione in segnale si chiama **codifica di linea (CL)** e dalla parte ricevente si chiama **decodifica di linea (DL).**
La codifica di linea serve per rendere il segnale fisico digitale adatto al particolare mezzo trasmissivo utilizzato. Nel campo della trasmissione dati la codifica di linea viene di solito denominata **modulazione in banda base**. La portante (l'onda modulata) in questo caso è costituita da un'onda quadra (segnale di clock).
La modulazione in banda base agisce sulla forma del segnale digitale e ha lo scopo di :

- modificare lo spettro del segnale per adattarlo alla banda passante del canale di trasmissione;
- consentire al dispositivo ricevente di estrarre il segnale di clock, in modo da poter effettuare la demodulazione (decodifica).

Oltre all'informazione principale dei dati da trasmettere, la codifica di linea deve quindi contenere anche l'informazione per mantenere il sincronismo tra l'host che trasmette e quello che riceve i dati. Il **segnale di sincronismo** è il segnale di clock, che sincronizza gli orologi delle schede di rete degli host connessi alla rete. E' necessario che la codifica dei dati binari abbia come effetto quello di

miscelare l'informazione con il segnale di sincronismo, permettendo al ricevitore di svolgere la funzione opposta di decodifica.

Infine, dopo essere stato trasformato in segnale, il messaggio viene preso in carico da un **Trasmettitore (TX)** che lo manda materialmente sul canale e dalla parte del destinatario ci sarà un **Ricevitore (RX)** che lo preleverà dal canale.

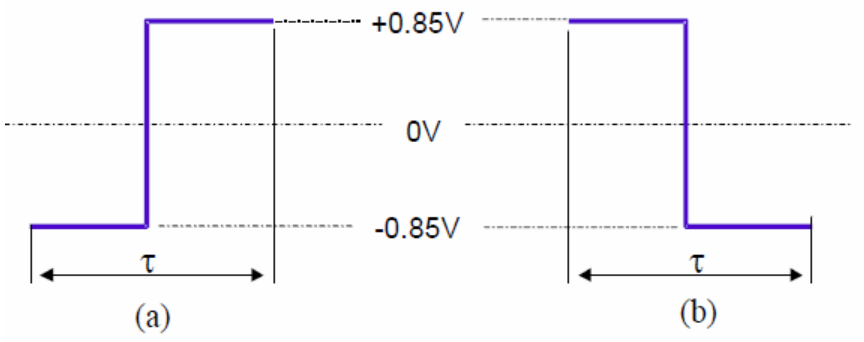
Codifica utilizzata nelle reti locali

Esistono numerosi metodi per effettuare la codifica dei dati. Un esempio e' la codifica di Mnchester.

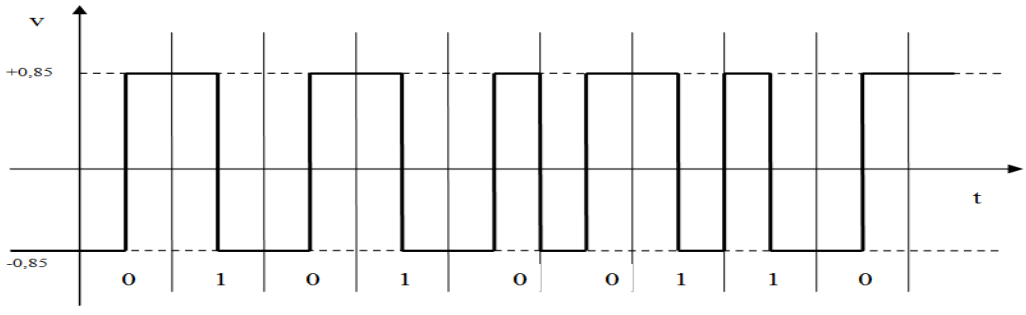
Codifica di Manchester

Consiste nel trasmettere un segnale ad onda quadra in cui per rappresentare il bit 1 si usa una transizione verso il basso a meta' del bit-time mentre per codificare lo 0 una transizione verso l'alto, sempre a meta' del bit-time.
I vantaggi sono :

- Il riconoscimento degli 1 e degli 0 e' più sicuro; infatti non si misura l'ampiezza dell'impulso (alto per 1 e basso per 0) ma si usa l'inversione di polarità, facilmente riconoscibile anche in caso di presenza di disturbi.
- Fornisce la sincronizzazione per tutte le interfacce collegate alla rete.



Codifica di Manchester : a) segnale associato a 0; b) segnale associato a 1.

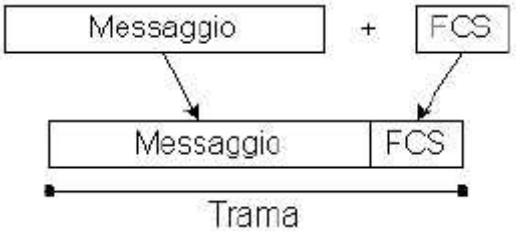


Rilevamento e correzione errori

La rilevazione d'errore consiste nella capacità da parte del destinatario del messaggio, di scoprire la presenza di errori causati dal rumore durante una trasmissione di dati. Il rilevamento degli errori di trasmissione è un' importante funzione del livello data link nelle reti.

Il ricevitore del messaggio deve essere in grado di determinare la sua integrità, per far ciò il nodo trasmittente aggiunge alcuni bit di controllo FCS (Frame Check Sequence).

- Il trasmittente calcola FCS applicando alla messaggio da inviare un determinato algoritmo;
- Il messaggio e l'FCS vengono inviate al destinatario.
- Il destinatario applica al messaggio ricevuto lo stesso algoritmo ricavando così FCS;
- Il ricevente confronta FCS calcolato con quello ricevuto se coincidono il frame di dati è arrivato integro. In caso di errore verrà richiesto il rinvio del frame al mittente.



Alcuni esempi di algoritmi per calcolare l' FCS sono il checksum e i codici CRC che descriveremo brevemente di seguito.

I bit di parità

Il bit di parità aggiunge al messaggio da trasmettere un ulteriore bit calcolato *per rendere pari il numero di bit trasmessi.* Ad esempio:

se il messaggio è *100011* il bit di parità sarà 1 e quindi al ricevente verrà trasmessa *100011* ***1***
se il messaggio è *110000* il bit di parità sarà 0 e quindi al ricevente verrà trasmessa *100011* ***0***
Il questo caso si parlera di **bit di parità pari**, ma esiste anche il calcolo del bit di parità dispari che consiste nel rendere *dispari il numero di 1 presenti nel messaggio* da trasmettere.

Il bit di parità pari si può calcolare velocemente applicando l'operatore logico XOR ai bit del messaggio. Ricordiamo la tavola della verita di questo operatore:

a	**b**	**a ^ b**
0	0	0
0	1	1
1	0	1
1	1	0

se il messaggio è *100011* allora 1^0^0^0^1^1 il bit di parità pari sara 1.
Questo sistema di controllo non è infallibile, infatti se nella trasmissione di un singolo bytedovessero verificarsi contemporaneamente 2 errori, il bit di parità non sarà in grado di rilevarli; per questo motivo esistono strumenti più avanzati di controllo che utilizzano più bit di parità.

Controllo di parità incrociato

È una sequenza di bit che viene utilizzata per verificare l'integrità di un dato o di un messaggio che può subire alterazioni.
Ad esempio un algoritmo potrebbe essere :

- Suddivido il messaggio in blocchi di lunghezza fissa
- Applico lo XOR ai bit di ciascun blocco (controllo trasversale) ottenendo T bit
- Applico lo XOR ai bit di stessa posizione nei i blocchi (controllo longitudinale) ottenendo B bit.
- Invio insieme al messaggio M , T e B cioè M:T:B

Il ricevitore calcolerà T' e B' se T = T' e B = B' allora il messaggio ricevuto è corretto altrimenti potrò rilevare e in alcuni casi correggere gli errori. Ad esempio:

controllo trasversale

1	1	1	1	1	0	1	0	0
1	1	0	0	0	0	0	1	1
0	1	1	1	1	0	1	1	0
0	0	0	0	0	0	1	1	0
0	1	0	0	1	1	1	0	0
1	1	1	1	0	1	1	1	1
1	1	1	1	1	0	1	0	

controllo longitudinale

Il messaggio originale di 48 bit è stato suddiviso in blocchi di 8 bit. E' stato applicato lo XOR ai bit di ciascuna riga (blocco). Lo Xor è stato applicato ai bit di ogni colonna (bit di uguale posizione in ciascun blocco) i bit così calcolati sono inviati al destinatario insieme al messaggio.

Codici CRC

Esiste però un altro metodo che nella pratica viene usato quasi sempre, il **Cyclic Redundancy Code** (CRC). I CRC sono basati sull'idea di considerare le stringhe di bit come rappresentazioni di polinomi a coefficienti 0 e 1 (un numero ad m bit corrisponde ad un polinomio di grado m-1).
Ad esempio, la stringa di bit 1101 corrisponde al polinomio x3 + x2 + x0.
L'aritmetica polinomiale è fatta in **modulo 2**, secondo le regole della teoria algebrica dei campi. In particolare:

· addizione e sottrazione sono equivalenti all'or esclusivo (non c'è riporto o prestito);
· la divisione è come in binario, calcolata attraverso la sottrazione modulo 2.

Per utilizzare questo metodo di controllo, il mittente ed il destinatario si mettono d'accordo su un **polinomio generatore G(x),** *che deve avere il bit più significativo e quello meno significativo entrambi uguali ad 1*. Supponiamo che G(x) abbia grado r.
Il frame M(x), del quale si vuole calcolare il checksum, deve essere più lungo di G(x). Supponiamo che abbia m bit, con m > r. L'idea è di concatenare in coda al frame un checksum tale che il polinomio corrispondente sia divisibile per G(x).
Quando il ricevente riceve il frame più il checksum, divide il tutto per G(x). Se il risultato è zero è tutto OK, altrimenti c'è stato un errore.
Il calcolo del CRC si effettua come segue:
1. Concatenare r bit a destra del frame, che quindi hanno m + r bit, e corrisponde ad $x^r M(x)$;
2. Dividere $x^r M(x)$ per G(x);
3. Accodare ad $x^r M(x)$ il resto della divisione effettuata al passo precedente. Ciò che si ottiene è il frame più il checksum da trasmettere, che è ovviamente divisibile per G(x). Si noti che di fatto questa è un'operazione di XOR fatta sugli r bit meno significativi, e quindi non modifica il frame.

Questo metodo è molto efficace, infatti un codice polinomiale con r bit:

- rileva tutti gli errori singoli e doppi;
- rileva tutti gli errori di x bit, con x dispari;

- rileva tutti i burst (gruppi contigui) di errori di lunghezza <= r.

I seguenti polinomi generatori sono diventati degli standard :

- CRC-12 : $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$;
- CRC-16 : $x^{16} + x^{15} + x^2 + 1$;
- CRC-CCITT : $x^{16} + x^{12} + x^5 + 1$;

Ad esempio la sequenza 10100011 si può rappresentare con il polinomio

$M(x) = x^7 + x^5 + x^1 + x^0$ il polinomio generatore è 1001 G(x) = x3 +1

Si divide il polinomio del messaggio per il polinomio generatore di grado
r <= s-1 **aggiungendo r zeri e dividendo in modulo 2**.

10100011000 : 1001 in modulo 2

Considerando le regole del modulo 2 sono :

0 - 0 = 0 0 - 1 = 1 1 - 0 = 1 1 - 1 = 0

Avremo come quoziente Q(x) = 10110101 e come resto R(x) = 101
Quindi il dato da trasmettere sarà 10100011 **101** cioè i coefficienti di M(x) e quelli di R(x) dove R(x) è il CRC.
Il ricevente divide modulo 2, il tutto messaggio ricevuto (compreso di CRC), se il resto è 0 la ricezione è corretta, altrimenti no.

Esempio
il messaggio da trasmettere è **1110**
- G(x) il polinomio generatore è 101
- quindi $M(x) = x^3 + x^2 + x$ e $G(x) = x^2 + 1$

Calcoliamoci il CRC :
Aggiungiamo alla stringa da trasmettere tanti 0 qunto è il grado del polinomio generatore

1110**00** : 101 = 1101 □il quoziente viene ignorato ai fini del calcolo del CRC
101

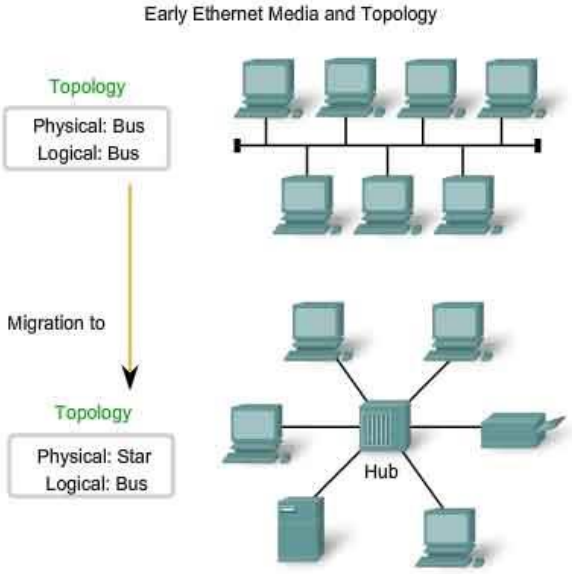
=100
101 ->□il divisore è contenuto nel dividendo perché ha gli stessi bit significativi
---
==100
101

=01 -> è il CRC, quindi il messaggio da trasferire sarà: 1110**01**

La rete Ethernet

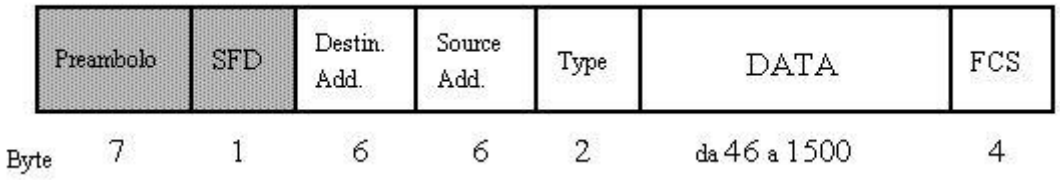
Ethernet rappresenta oggi la rete più nota e più diffusa in tutto il mondo. La nascita di Ethernet risale al 1976 quando Xerox utilizzò il protocollo CSMA/CD per realizzare una rete locale con una velocità di 2,94 Mbit/s per collegare oltre 100 stazioni. Ethernet incontrò subito un notevole successo per la sua semplicità realizzativa e le elevate prestazioni; per questo motivo Digital, Intel e Xerox formarono un consorzio DIX per elaborare le specifiche della rete Ethernet a 10 Mbit/s. Negli stessi anni il

comitato IEEE 802 iniziò a sviluppare uno standard di rete locale basato su CSMA/CD e simile alla rete Ethernet, noto come IEEE 802.3.
Ethernet e IEEE 802.3 sono molto simili, anche se esistono differenze significative. Oggi si realizzano soltanto reti IEEE 802.3 ma in molti casi si continua ad utilizzare la denominazione di rete Ethernet. In questo capitolo i due termini saranno usati indifferentemente per indicare IEEE 802.3.
Le reti Ethernet e IEEE 802.3 si basano su una struttura a bus con una velocità di 10 Mbit/s. Lo standard IEEE 802.3 specifica il livello fisico (Codifica di Manchester) e il livello MAC.



La *topologia fisica* di una rete Ethernet o IEEE 802.3 può essere sia a bus che a stella mentre la *topologia logica* è sempre a bus. Il bus può essere cavo coassiale, doppino telefonico e fibra ottica. Il metodo di accesso multiplo CSMA/CD (Carrier Sense Multiple Access with Collision Detect) che non prevede una stazione master. Questa caratteristica, insieme alla semplicità del protocollo CSMA/CD, sono i motivi della grande diffusione di Ethernet.

Formato del frame IEEE 802.3

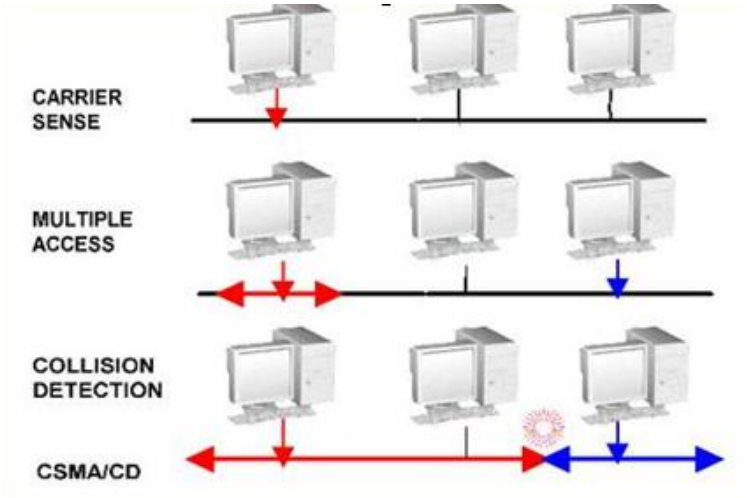


I campi che compongono il frame nella rete IEEE 802.3 sono:
Preambolo. Campo lungo 7 byte, ognuno costituito dalla sequenza 10101010.
Delimitatore di inizio del frame. campo formato dal byte10101**11** e serve ad indicare l'inizio del frame.
Destination e source MAC address . MAC address destinatario e sorgente.Se il bit più significativo del MAC di destinazione è uguale a 0, allora il MAC è *ordinario*, mentre se tale bit è uguale a 1 allora si ha una trasmissione *multicast*. Al contrario, se l'indirizzo della stazione di destinazione è formato da bit uguali a 1, allora si ha una trasmissione *broadcasting*.
Lunghezza del campo dati. Lunghezza in byte del campo dati contenuti nel frame.

Campo Dati. Questo campo ha una lunghezza variabile tra 46 e 1500 byte. La lunghezza non può essere minore di 46 byte per garantire che la lunghezza minima delle trame non sia inferiore a 64 byte. Come vedremo, questo valore minimo del pacchetto è necessario per un corretto funzionamento del protocollo CSMA/CD.
FCS. Campo di 4 byte, consente di rivelare eventuali errori di trasmissione.

Il protocollo CSMA/CD

Il protocollo prevede che la strazione *trasmittente ascolti il canale sia prima di trasmettere che durante la trasmissione. Se trasmettendo rileva una collisione, si ferma, segnala a tutte le altre stazioni la collisione e riprova dopo un intervallo di tempo pseudocasuale.*



Esempio
- L'host A deve trasmettere un frame, prima di farlo ascolta il canale (Carrier Sense)
- Trovando libero inizia a trasmettere
- Anche l'host B deve trasmettere ma non lo fa trovando occupato il canale (Multiple Access).
- L' host A mentre trasmette ascolta il canale per verificare se vi sono collisioni(Collision Detect)

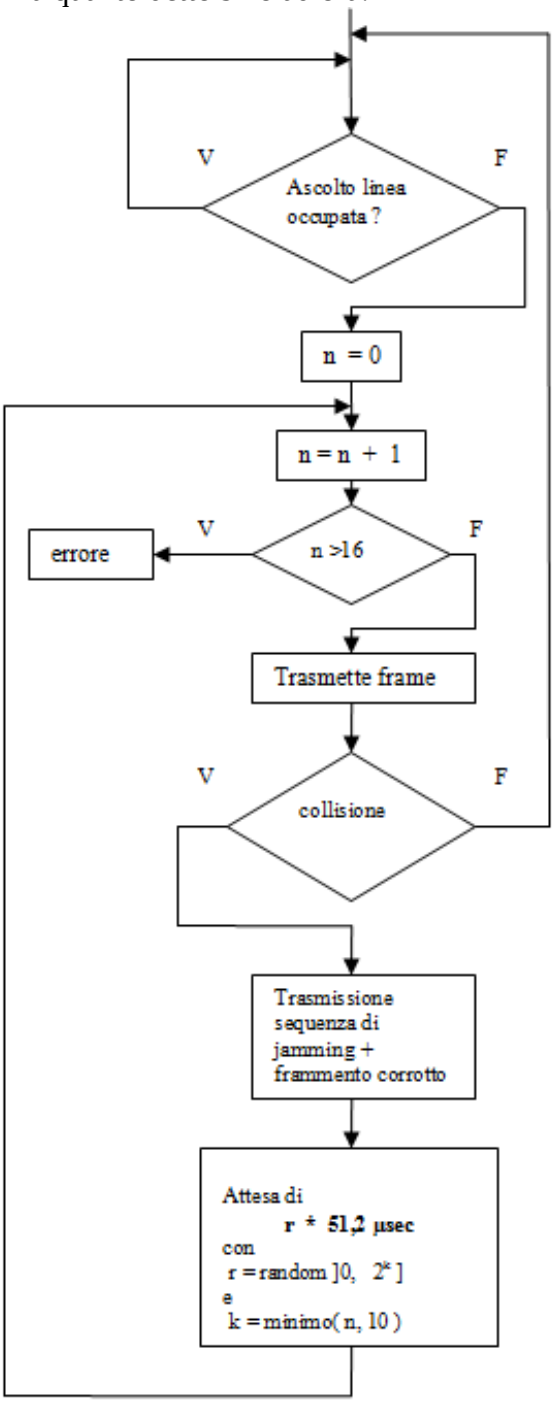
Esempio
- Gli host A e B devono trasmettere un frame, prima di farlo ascoltano il canale (Carrier Sense);
- Trovando entrambe libero (o perche lo fanno contemporaneamente o a causa del tempo di propagazione del segnale sul canale) iniziano a trasmettere;
- I segnali di A e B collidono sommandosi sul canale (Multiple Access);
- L' host A e B mentre trasmettono ascoltano il canale e rilevano la collisione (Collision Detect);
- Avvisano con un frame particolare le altre stazioni dell'avvenuta collisione. Tale frame è costituito dal frammento prodotto dalla collisione più una sequenza di 48 bit detta di *jamming*.
- Le altre stazioni ricevono la sequenza di jamming e scartano i frame corrotti.

Dopo una collisione la ritrasmissione avviene dopo un tempo pseudocasuale e ciò perché tutti gli altri host che sono contemporaneamente in attesa potrebbero simultaneamente tentare di ritrasmettere generando nuove collisioni. L**'algoritmo di backoff** stabilisce l'intervallo di tempo che intercorre fra due tentativi di trasmissione:
Nell' ipotesi di una rete a **10Mb** con
n = numero del tentativo
k = minimo (n, 10)
r = random] 0, 2^k] avremo che:

$$T_{attesa} = r * 51{,}2\ \mu sec$$

Dopo 16 ritrasmissioni viene interrotto ogni tentativo e viene trasmesso al livello superiore un messaggio di errore.

I flow chart che segue sintetizza quanto detto sino ad ora:



Condizione necessaria per il Collision Detect

Condizione necessaria per la rilevazione della collisione da parte di una stazione trasmittente è che il tempo T di immissione in rete della trama sia maggiore o al più uguale al tempo massimo di andata e ritorno *Tar* della trama ethernet più piccola (64 Byte):

$$T >= Tar$$

Calcoliamoci Tar impostando la proporzione : se in un secondo vengono trasmessi 10000000 di bit, quanti secondi saranno necessari per trasmettere la trama più piccola 64 * 8 = 512 bit

1 sec : 10000000 bit = Tar : (64 * 8)

Tar = (64 * 8) bit / 10000000 bit/sec = 512/ 10000000 = 0,0000512 sec = 51,2µsec

Che rappresenta il tempo di andata e ritorno della trama ethernet di dimensioni minime (64 Bytes) in una rete da 10 Mb.

Calcoliamoci ora quanto spazio percorre la minima trama ethernet, considerando che la velocita di propagazione del segnale nel mezzo trasmissivo è 200.000 Km/sec :

S = v * Tar

S = 200000 km/sec * Tar sec

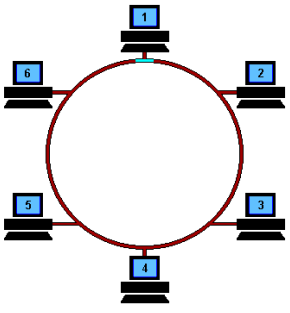
S = 0,0000512 sec * 200000 km/sec = 10,24 km

Il risultato, cioè la massima distanza fra due host per una rete da 10Mb che è circa 5km (considerando che il segnale deve andare e tornare), risulta circa il doppio della distanza stabilita nelle specifiche Ethernet v2 (2,8 Km).

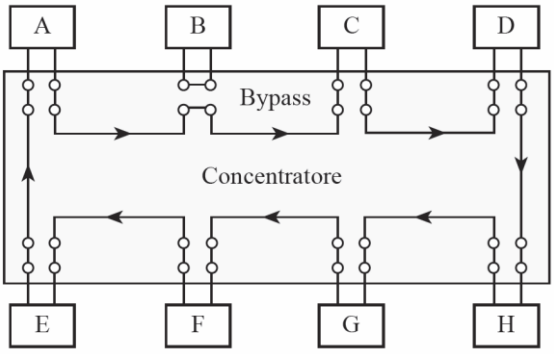
La rete Toket Ring (IEEE 802.5)

La rete Token Ring è stata sviluppata dalla IBM nel 1974 come alternativa a Ethernet. Inizialmente la velocità trasmissiva era di 4 Mb/s poi è arrivata a 16 Mb/s. Il suo sviluppo è stato limitato dall'eccessivo costo di realizzazione rispetto alla rete Ethernet.

E' una rete punto-punto, in cui ogni host trasmette i dati a quello successivo e li riceve da quello che lo precede. Quando una stazione riceve un messaggio confronta il MAC address destinatario con il proprio se sono uguali, prima di inviarlo all'host successivo ne crea una copia locale, in caso contrario lo reivia senza fare la copia.



Poiché le stazioni devono ripetere continuamente i pacchetti delle altre stazioni, per ragioni di affidabilità la rete viene cablata a stella. Quando una stazione è spenta o guasta, il centro stella (concentratore) la esclude dalla rete.



Il metodo di accesso è di tipo a token. Il token (gettone) è un particolare frame che circola sull'anello (ring), indicando che l'anello è libero. Una stazione che intenda trasmettere deve aspettare che arrivi il token, catturarlo e quindi trasmettere. Il token circola continuamente sull'anello anche se le stazioni non hanno dati da trasmettere. Esso viene generato inizialmente dalla stazione che si è guadagnata il diritto di essere l'active monitor della rete e viene ripetuto da tutte le stazioni. Quando una stazione cattura il token può trasmettere uno o più frame, in funzione della loro lunghezza e di un parametro detto THT (Timer Holding Token) che indica il tempo massimo per cui una stazione può trattenere il token.
Questo tipo di protocollo funziona in modo *deterministico* è ciò consente di calcolare il tempo massimo di consegna di un messaggio. Tale caratteristica ne consiglia l'impiego nelle applicazioni *real time*.

Trame IEEE 802.5

La trama del token è rappresentata di seguito:

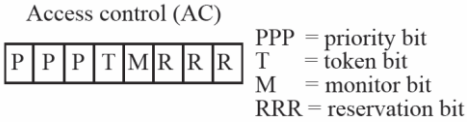
	Starting Delimiter	Access Control	Ending Delimiter
Ottetti	1	1	1

Starting Delimiter	Access Control	Frame Control	Destin. Address	Source Address	Routing Inform.	INFO	FCS	Ending Delimiter	Frame Status
Ottetti 1	1	1	6	6	da 0 a 30	da 4 a 17749	4	1	1

- ***start delimiter*** (1 byte) è un preambolo che indica l'inizio del frame;
- ***end delimiter*** (1 byte) indica la fine del frame e riporta se si tratta dell'ultimo frame di una sequenza o meno;

Ambedue questi campi violano la codifica di Manchester per indicare appunto che inizia e finisce la trame

- ***access control*** (1 byte) i 3 bit più significativi indicano la priorità (priority field), il bit adiacente indica se il frame in questione è un token od un frame di dati/comandi, quello successivo è utilizzato per il monitoraggio dei pacchetti (che potrebbero circolare nella rete all'infinito) ed i 3 meno significativi sono riservati;



- ***frame control*** (1 byte) indica il tipo di frame. Ad esempio 0 indica un frame MAC (Medium Access Control), mentre il valore 40 indica un frame LLC (Logical Link Control);

- ***destination address*** (6 byte) specifica l'indirizzo fisico dell'interfaccia di destinazione;
- ***source address*** (6 byte) specifica l'indirizzo fisico dell'interfaccia mittente;
- ***Routing information*** informazione di routing fra i bridge
- ***INFO*** è l'informazione oggetto della comunicazione;
- ***FCS*** (4 byte) il Frame-Check Sequence è un codice per il controllo degli errori di comunicazione;
- ***frame status*** (1 byte) riporta informazioni relative allo stato di ricezione del pacchetto. Grazie a questo campo il mittente può essere informato se il destinatario esiste o meno e se esso ha ricevuto il pacchetto;

Active Monitor

L' active monitor è quell'host della rete che ha il compiuto di generare il token:

- allo start della rete;
- in tutti i casi in cui è necessario rigenerarlo (cancellato, token ricevuto non sincronizzato..).

Questo particolare host viene designato per elezione dagli host della rete nel deguente modo:

- allo start della rete gli host trasmettono in broadcast un frame contenente il proprio MAC address;
- contemporaneamente ricevono i frame con i MAC degli altri host;
- quando un host riceve un MAC maggiore del proprio interrompe la trasmissione
- alla fine resta solo l'host con MAC address maggiore e questo risulta essere l' active monitor.

Rilascio normale e forzato del token

Quando il frame ritorna all'host che l'ha generato quest'ultimo lo toglie dalla rete e rimette in circolazione il token. In questo caso si parla di ***rilascio normale del token.*** Se invece, per un discorso di maggiore efficienza, il token viene rilasciato subito dopo l'invio del frame si parla di ***rilascio forzato del token.***

Procedure diagnostiche

Nel caso di fault della rete è importante che ogni host conosca il MAC address di quello che lo precede, per permettere al livello diagnostico di individuare quale tra i collegamenti punto-punto dell'anello è guasto. La procedura per consentire ad un'host di conoscere l'indirizzo del suo predecessore prende il nome ***neighbour notification*** (notifica del vicino).
L'active monitor invia un frame particolare chiamato **AMP** (Active Monitor Presence). Chi riceve il token, oltre a copiare il MAC dell'host mittente modifica il bit per segnalare che lo ha ricevuto, in coda al frame che rimette in circolo trasmette un nuovo frame AMP dove inserisce i propri dati.
Alla fine, tutte le stazioni conoscono l'indirizzo della stazione più vicina.

Indirizzo IP

Un indirizzo IP è un identificativo numerico che viene associato univocamente ad *un'interfaccia di un host* collegato a una rete che utilizza il protocollo IP. Ha due funzioni principali:

- identificare un'interfaccia di un dispositivo di rete
- fornirne il percorso per raggiungerlo da un altro host

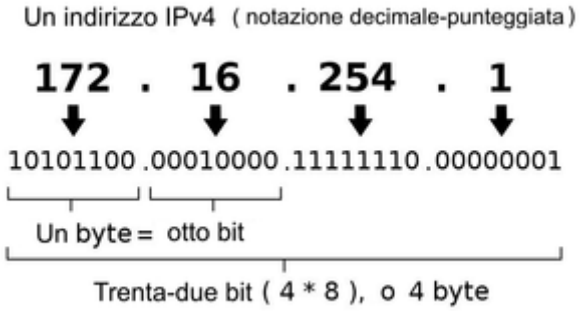
Gli indirizzi IP pubblici sono rilasciati e regolamentati dall'ICANN tramite una serie di organizzazioni delegate. A causa della saturazione dello spazio di IPv4 si è proceduto alla definizione della nuova versione del protocollo IP: IPv6. Tale versione è basata su indirizzi a 128 bit anziché a 32 e ciò permette l'assegnazione di un numero decisamente maggiore di indirizzi.

La difficile implementazione a livello globale dell'IPv6 ha portato all'introduzione di nuovi concetti che hanno cambiato la teoria delle reti. Vanno citati il progressivo abbandono del concetto di classi di indirizzi IP e il conseguente utilizzo sempre maggiore di indirizzi classless (privi del concetto di classe), la subnet mask, la riorganizzazione gerarchica degli indirizzi mediante utilizzo del NAT (Network address translation).

Composizione e formati

IPv4

L'indirizzo IPv4 è costituito da 32 bit (4 byte) suddiviso in 4 gruppi da 8 bit (1 byte o ottetto), separati ciascuno da un punto (notazione dotted) (es. 11001001.00100100.10101111.00001111).



Ciascuno di questi 4 byte è poi convertito in formato decimale di più facile identificazione (quindi ogni numero varia tra 0 e 255, infatti il massimo numero rappresentabile in 8 bit è 2^8 -1 = 255).
Ogni indirizzo IP è suddiviso in due parti:

- La prima parte identifica la rete, ed è utilizzato per l'instradamento dai router.
- La seconda parte identifica l'host ed è utilizzato, una volta raggiunta la rete di destinazione, per l'instradamento a livello locale dell'host.

network id	host id

Assegnazione degli indirizzi IP

La parte ***net_id*** degli indirizzi è assegnata dall'ICANN mentre l'assegnazione della parte ***host_id*** è delegata al richiedente che eventualmente può suddividerla ulteriormente per la creazione di altre sottoreti logiche (subnetting) evitando duplicazioni e sprechi di indirizzi.

Gli indirizzi IP possono essere assegnati in modo ***statico*** o ***dinamico***. L'indirizzo IP statico è un indirizzo che è assegnato in modo *permanente* da autorità riconosciute a livello internazionale.

L'indirizzo IP dinamico è un indirizzo che è assegnato *al momento del collegamento* dal provider questo indirizzo non rimane fisso cambia ogni volta che si affettua un collegamento alla rete.

Classi di indirizzi IP

Le classi sono un modo per caratterizzare lo spazio di indirizzamento IPv4. Il **Classful addressing** (indirizzamento basato sulla classe) prevede che dai primi bit di un indirizzo si potesse determinare il tipo di indirizzo. Questa scelta, con il crescere dell'utenza di internet, si è rivelata troppo rigida, ed è stata abbandonata a favore dell'indirizzamento senza classe (CIDR). Di seguito la tabella delle classi in cui sono stati suddivisi gli indirizzi IP:

Classe	Bit iniziali	numero di bit rete	host	reti disponibili	host disponibili	Indirizzi totali	intervallo
A	0	7 bit	24 bit	128	16.777.216 (-2)	2.147.483.392	0.0.0.0 - 127.255.255.255
B	10	14 bit	16 bit	16.384	65.536 (-2)	1.073.709.056	128.0.0.0 - 191.255.255.255
C	110	21 bit	8 bit	2.097.152	256 (-2)	532.676.608	192.0.0.0 - 223.255.255.255
D	1110	28 bit				268.435.456	224.0.0.0 - 239.255.255.255
E	11110	27 bit				134.217.728	240.0.0.0 - 255.255.255.255

La classe può quindi essere individuata dai primi bit:

- **classe A**: il primo byte rappresenta la rete, gli altri l'host; [0-127].x.x.x. La maschera di rete è 255.0.0.0, o /8. Questi indirizzi iniziano tutti con il bit più significativo a 0.
- **classe B**: i primi due byte rappresentano la rete, gli altri l'host; [128-191].y.x.x (gli y sono parte dell'indirizzo di rete, gli x dell'indirizzo di host). La maschera di rete è 255.255.0.0, o /16. Questi indirizzi iniziano tutti con i primi due bit più significativi a 10.
- **classe C**: i primi 3 byte rappresentano la rete, l'altro l'host; [192-223].y.y.x. La maschera di rete è 255.255.255.0, o /24. Questi indirizzi iniziano tutti con i primi tre bit più significativi a 110.
- **classe D**: riservata agli indirizzi multicast: [224-239].x.x.x. Questi indirizzi cominciano con la sequenza 1110. Non hanno maschera di rete, essendo tutti e 32 i bit dell'indirizzo utilizzati per indicare un gruppo, non un singolo host.
- **classe E**: riservata per usi futuri: [240-255].x.x.x. Questi indirizzi cominciano con la sequenza 11110 e non è definita una maschera di rete.

Esempio : calcolo del numero di reti e host della classe B:
reti: $2^{14} = 16.384$
hosts: $2^{16} = 65.536$ - (indirizzi di rete e di broadcast) = 65.534 indirizzi netti
Per il calcolo del numero delle reti l'esponenete dovrebbe essere 16 ma i primi due bit sono sempre a 10 (proprio per riconoscere la classe della rete) quindi 16 - 2 = 14.

Indirizzi IP privati

Alla regola secondo cui *un'indirizzo di IP è unico* in Internet, fanno eccezione gli indirizzi privati (rappresentati in tabella) che possono essere *usati liberamene* nell'ambito di reti LAN.
Di seguito la tabella degli indirizzi privati per ogni classe.

Classe	Indirizzi			Numero Totale indirizzi
A	10.0.0.0	-	10.255.255.255	16.777.216
B	172.16.0.0	-	172.31.255.255	1.048.576
C	192.168.0.0	-	192.168.255.255	65.536

Indirizzi IP riservati

Sono il *primo e l'ultimo indirizzo di una rete* di host, che non possono essere assegnati a nessun dispositivo in quanto rappresentano il primo ***l'indirizzo di rete*** il secondo ***l'indirizzo di broadcast***.

Indirizzi IP speciali

Gli indirizzi speciali sono 2 e sono le reti 127.0.0.0 e 0.0.0.0. Dalla classe A è stata esclusa l'ultima rete la **127.0.0.0** con maschera 255.0.0.0 che contiene tutti gli indirizzi detti di **loopback** o **localhost.** Questi indirizzi, in particolare **127.0.0.1**, possono essere usati dalle applicazioni per comunicare con lo stesso sistema su cui sono in esecuzione. Servono a scopo diagnostico oppure quando un programma deve fare riferimento allo stesso computer su cui "gira". Nel protocollo IPv6 l'indirizzo di loopback è **::1**

L'indirizzo speciale **0.0.0.0**, conosciuto come **default route** è il percorso predefinito per l'instradamento dei pacchetti. Quando un router deve instradare un pacchetto e non ha un riferimento specifico nella sua tavola di routing verso la rete di destinazione (ricavata dall'indirizzo di destinazione presente nel pacchetto), lo intrada verso l'interfaccia assegnata alla defaul route.

Indirizzi APIPA

Quando un client Windows è configurato dinamicamente e non riesce ad ottenere per un qualsiasi motivo un indirizzo di IP da un server DHCP, si autoassegna un indirizzo particolare appartenente alla rete di classe B **169.254.0.0** Questi indirizzi vengono detti appunto APIPA (Automatic Private IP Addressing).

Esercizi da svolgere

Questa tipologia di esercizio consiste nell'analizzare l'indirizzo di IP dato specificando :

1. la classe di appartenenza (A, B, C)
2. se e pubblico o privato
3. se è speciale
4. se è riservato cioè se è un indirizzo di *rete* o di *broadcast*.

1. 40.88.44.10
2. 199. 100. 50. 64
3. 125.255.255.255
4. 14. 0. 255. 15
5. 192.168.99.99
6. 172. 16. 255. 254
7. 72. 82. 45. 93
8. 192. 168. 12. 245
9. 193. 204. 68. 31
10. 10.0.0.0

Segmentazione della rete – Le sottoreti

Il concetto di classe nelle reti di calcolatori ed il conseguente *classfull addressing* sono stati presto superati per due motivi principali:

- Molti utilizzatori di Internet (enti o aziende) avevano bisogno di un numero di indirizzi IP pubblici superiore a quello gestibile con una rete di classe C (254), ma generalmente inferiore a quello gestibile con una rete di classe B (65534). Tipicamente si facevano assegnare una rete di classe B e molti degli indirizzi possibili rimanevano inutilizzati. Ciò ha portato ad una rapida diminuzione degli indirizzi di classe B disponibili.
- Le dimensioni delle tabelle di instradamento (routing table) dei router erano diventate enormi e difficili da gestire, in quanto contenevano un numero enorme di indirizzi di rete, di cui molti erano relativi ad una stessa organizzazione o azienda, che magari si era fatta assegnare più reti di classe C per soddisfare le proprie esigenze.

Un primo passo verso lo svincolo dalle classi è rappresentato dal ***subnettig classful*** o **FLSM** (**Fixed Length Subnet Masking**) che prevede che l'indirizzo IPV4 venga suddiviso non più in due parti ma in tre. Il nuovo campo è il campo *indirizzo di sottorete* i cui bit vengono generalmente "rubati" alla parte indirizzo dell'host.

indirizzo di rete	indirizzo di sottorete	indirizzo di host

Il passo successivo verso lo svincolo delle classi è stato il **VLSM** (**Variable Length Subnet Masking).**

VLSM

Il concetto di *classfull addressing* porta spesso ad un grande spreco di indirizzi di IP. *Se utilizziamo invece il* **FLSM** *lo spreco diminuisce ma è necessario che tutte le sottoreti di una singola rete siano della stessa dimensione,* questo perché i router devono conoscere quanto è lunga la parte rete di un indirizzo di IP per poter instradare i pacchetti verso la rete destinazione.

La **VLSM** (**Variable Length Subnet Masking**) supera il problema della dimensione fissa delle sottoreti nell'ambito di una rete tipica del *subnetting classfull*. Con l'indirizzamento VLSM i *router ricevere informazioni di routing che includono non solo l'indirizzo IP dell'host, ma anche la maschera di rete* (subnet mask) e che indica il numero di bit che costituiscono la porzione relativa alla rete dell'indirizzo IP.

Subnet mask

In origine i router distinguevano la classe di appartenenza di una rete (e quindi la parte indirizzo rete da quello dell'host) dai primi bit dell'indirizzo di IP**.** Con l'introduzione del concetto di ***classless*** l'uso dei bit iniziali viene superato dall'introduzione della subnet mask. La subnet mask è una sequenza di 32 bit (la stessa lunghezza degli indirizzi IP) ognuno dei quali indica se considerare i corrispondenti bit dell'indirizzo IP come facenti parte del *net id* (1) oppure facenti parte dell'*host id* (0). Da quanto detto, i bit della subnet mask impostati ad 1 devono essere consecutivi e devono essere quelli più significativi della maschera.

La subnet mask consente partendo dall' indirizzo di IP di un host di conoscerne la rete di appartenenza. In sostanza la subnet mask consente di separare la parte net da quella host in un'indirizzo di IP.

I 32 bit della subnet mask in AND con l' indirizzo di IP di un interfaccia ci restituisce la rete a cui appartiene l'host che la possiede. Ad esempio dato l'indirizzo 192.168.1.100 con la maschera 255.255.255.0

192.168. 1.100 ***and***
255.255.255. 0

192.168. 1. 0 che rappresenta l'indirizzo della rete di appartenenza dell'host 192.168.1.100

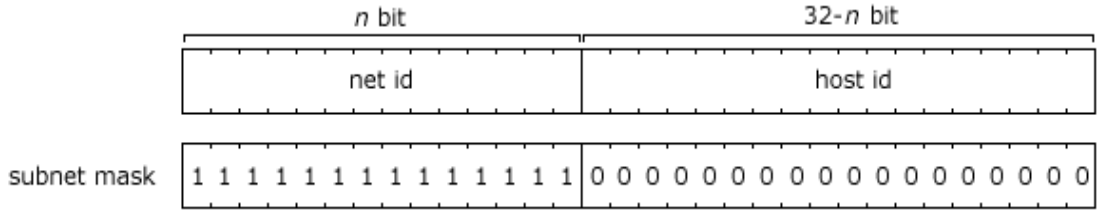
La stessa cosa si ottiene utilizzando il

net prefix *che rappresenta il numero di bit a 1 della corrispondente maschera di rete*

Nota: Nell' IPV6 viene detto ***prefix-length***

Ad esempio *192.167.58.147 / 20*
Gli indirizzi IP vengono quindi specificati ponendo dopo l'indirizzo stesso la relativa subnet mask o il net prefix.

Oltre a VLSM, nella RFC 1519 è stato proposto e accettato il **CIDR (Classless Inter-Domain Routing)**. CIDR non considera più le classi di rete. CIDR identifica la rete basate unicamente sul numero di bit del *net prefix* , (corrispondente al numero di bit 1 nella maschera di sottorete). Infatti come sappiamo è sufficente mettere in and l'indirizzo dell'host con il net prefix per ottenere l'indirizzo della rete di appartenenza.



Un esempio di un indirizzo IP scritto utilizzando la *notazione* CIDR è 172.16.1.1 / 16, dove l'/ 16 rappresenta il numero di bit nel prefisso di rete.
Ignorando le tradizionali classi di indirizzi, CIDR consente all' ISP di assegnare un blocco di indirizzi di IP un blocco di indirizzi in base al numero di indirizzi di host richiesti.

Supernetting

Abbiamo detto che net subnetting i bit per la sottorete vengono "*rubati*" alla parte indirizzo dell'host, se questo avviene per la parte rete si ha il ***supernet***.

Esercizi di analisi degli indirizzi IP

Questa tipologia di esercizio consiste nell'analizzare l'indirizzo di IP dato specificando :

1. la classe di appartenenza (A, B, C)
2. se è subnettato o no
3. se è pubblico o privato
4. se è speciale

5. numero di reti, sottoreti e host
6. a quale rete, sottorete appartiene
7. se è un indirizzo riservato (indirizzo di rete o di broadcast) .

Esempio :
Analizzare l'indirizzo 198.68.15.18 255.255.255.248

Svolgimento:

- è un indirizzo di **classe C**, infatti appartiene all'intervallo 192.0.0.0 - 223.255.255.255
- è **subnettato** la sua maschera è diversa dalla maschera natutale della classe C 255.255.255.0
- è un **indirizzo pubblico**, infatti NON appartiene all'intervallo 192.168.0.0 – 192.168.255.255
- **non è un' indirizzo speciale**, infatti è diverso da 0.0.0.0 e 127.0.0.1
- il net prefix è / 29 $248 = 11111000_2$ sono 5 bit a 1 che sommati ai 24 dela classe C danno 29
- numero di bit per le reti = 24 tanti quanti ne prevede la Classe C
- numero di bit per le sottoreti = 29 – 24 = 5 subnet mask meno la subnet mask della classe C
- numero di bit per gli host = 32 – 29 = 3 bit totali di un indirizzo IP meno la subnet mask
- numero reti $2^{24 - 3} = 2^{21}$ la classe C ha i primi 3 bit riservati del valore 110
- numero sottoreti $2^5 = 32$
- numero host $2^3 = 8$
- numero host netti 8 – 2 = 6 cioè meno il primo e l'ultimo indirizzo di rete che rappresentano l'indirizzo della rete e quello di broadcast
- il magic number o block size = 256 - 248 = 8
- Enumeriamo le sottoreti partendo dalla major network che si ottiene dall' AND bit a bit fra l'indirizzo dato e la subnet mask naturale della classe C e cioè :

198.68.15.0 major network
198.68.15.1 indirizzo 1^ host
198.68.15.2 indirizzo 2^ host
....
198.68.15.6 indirizzo 6^ host
198.68.15.7 indirizzo di broadcast della rete 198.68.15.0

198.68.15.8 indirizzo di sottorete
198.68.15.9 indirizzo 1^ host
198.68.15.10 indirizzo 2^ host
....
198.68.15.14 indirizzo 6^ host
198.68.15.15 indirizzo di broadcast della rete 198.68.15.8

198.68.15.16 indirizzo di sottorete
198.68.15.17 indirizzo 1^ host
198.68.15.18 indirizzo 2^ host <<<<<
....
198.68.15.22 indirizzo 6^ host
198.68.15.23 indirizzo di broadcast della rete 198.68.15.16

Quindi potremo concludere che 198.68.15.18 / 29 è l'indirizzo del 2^ host della 3^ sottorete 198.68.15.16 / 29

Altri esercizi da svolgere:

1	192.168.100.15	255.255.255.248
2	71. 81. 47. 93	255.255.224.0
3	172.16.96.254	/19
4	200. 100. 50. 64	/ 26
5	12. 0. 255. 15	/ 24
6	192.168.12.27	/29
7	78.32.0.0	/19
8	31.63.0.0	/27

Esercizi di sintesi
Questa tipologia di esercizio, dati il numero di sottoreti e/o dal numero di host, consente di di trovare quali reti soddisfano le necessità.
Esempio:
Vorremmo 7 subnet e 63 host
Nota
Prima di procedere al calcolo dei bit necessari per le subnet e gli host dobbiamo osservare che ai 63 indirizzi effettivi (netti) degli host vanno sempre aggiunti 2 indirizzi relativi al indirizzo di sottorete e di broadcast ottenendo così in numero di host lordi.

Vediamo quanti bit sono necessari per le subnet e gli host
$2^x \geq 7$ => con x = 4 avremo 16 subnet
$2^y > 63 + 2$ => con y = 7 avremo 128 host
7+4 =11 è possibile una rete in classe B (16 bit). Ma in questo caso avremo 16-11 = 5 bit in più da redistribuire fra subnet e host. Ad esempio potremmo :
con x = 4 + 1 = 5 avremo 32 subnet
con x = 7 + 4 = 11 avremo 2048 - 2 host netti

16 ! 5| 11 |
! net !sn| h |
172. 16. 0. 0
Il net prefix sarà uguale ai 16 bit della classe B più gli 5 della subnet quindi **/ 21**
oppure come subnet mask 255.255.248.0

il magic number (block size) è 256 - 248 = 8

172. 16. 0. 0 è la major subnet
172. 16. 8. 0
172. 16. 16. 0
172. 16. 24. 0
172. 16. 32. 0

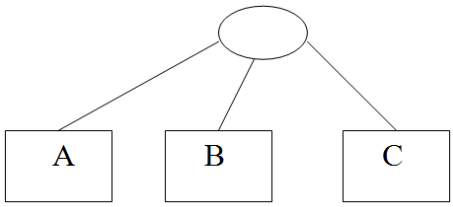
172. 16. 248. 0
172. 16. 255.255 broadcast ultima rete

Altri esercizi da svolgere :

	Numero sottoreti	**Numero host**
1	12	7
2	39	2010
3	4	128

4	7	729
5	15	17
6	68	2427
7	2723	5423
8	15	2061

Esercizio
L' esercizio che segue è riassuntivo delle caratteristiche del FSLM (maschera fissa) comparato alla VLSM (maschera variabile).
Supponiamo di avere 3 reti A, B e C costituite da 280, 80 e 50 host Si chiede di assegnare tramite la tecnica di subnetting, gli indirizzi alle 3 sottoreti, *ottimizzando l'efficienza di utilizzazione dello spazio degli indirizzi.*



SOLUZIONE 1 FSLM
Considerando la rete più grande
$2^Y > 280 + 2$ $Y = 9$ cioè avremo a disposizione non 282 ma 512 host per sottorete. In questo partiremo da un rete di classe B. Considerando che la classe B ha 16 bit per gli host
16 - 9 = 7 saranno i bit per le sottoreti quindi $2^7 = 128$ sottoreti ciascuna da 512 -2 host validi.

Il net prefix è 16 della classe B + 7 "rubati" per le 128 sottoreti avremo quindi /23 la maschera di rete 255.255.11111110. 0 quindi 255.255.254.0 .
Il block size è 256 - 254 = 2
Partendo dalla rete 130.10. 0.0 viene assegnato alla rete A
130.10. 2.0 viene assegnato alla rete B
130.10. 4.0 viene assegnato alla rete C

Calcoliamo per ogni sottorete glli indirizzi sprecati :
512 – 282 = 230 indirizzi inutilizzati per la rete A
512 – 82 = 430 indirizzi inutilizzati per la rete B
512 – 52 = 460 indirizzi inutilizzati per la rete C
230 + 430 + 460 = **1120** indirizzi inutilizzati totali rispetto ai 282 + 82 + 52 = 416 necessari.

SOLUZIONE 2 VLSM
Consideriamo il vincolo sugli host per sottorete:
Per la rete A di 282 host $2^Y > 282$ $Y = 9$ bit quindi 512 – 2 host netti
Per la rete B di 82 host $2^Y > 82$ $Y = 7$ bit quindi 128 – 2 host netti
Per la rete C di 52 host $2^Y > 52$ $Y = 6$ bit quindi 64 – 2 host netti

Quindi per A i bit della sottorete saranno 16 - 9 = 7 che sommati ai 16 della rete daranno /23
Quindi per B i bit della sottorete saranno 16 - 7 = 9 che sommati ai 16 della rete daranno /25
Quindi per C i bit della sottorete saranno 16 - 6 = 10 che sommati ai 16 della rete daranno /26
La subnet mask di A sarà 255.255.254.0 il block size =256-254 = 2
La subnet mask di B sarà 255.255.255.128 il block size =256-128 = 128
La subnet mask di C sarà 255.255.255.192 il block size =256-192 = 64

130.10. 0. 0 indirizzo di rete della rete A
130.10. 0. 1 primo indirizzo valido
.
.
130.10. 1.254 ultimo indirizzo valido
130.10. 1.255 indirizzo broadcast della rete A

130.10.2. 0 indirizzo di rete della rete B (ottenuta sommando 2 alla rete precedente)
130.10.2. 1 primo indirizzo valido
.
.
130.10.2.126 ultimo indirizzo valido
130.10.2.127 indirizzo broadcast della rete B

130.10.2.128 indirizzo di rete della rete C (ottenuta sommando 128 alla rete precedente)
130.10.2.129 primo indirizzo valido
.
.
130.10.2. 190 ultimo indirizzo valido
130.10.2. 191 indirizzo broadcast della rete C (ottenuta sommando 64-1 alla rete precedente)

Calcoliamo per ogni sottorete gli indirizzi sprecati :
512 – 282 = 230 indirizzi inutilizzati per la rete A
128 – 82 = 46 indirizzi inutilizzati per la rete B
64 – 52 = 12 indirizzi inutilizzati per la rete C
230 + 46 +12 = **288** indirizzi inutilizzati totali rispetto ai 282 + 82 + 52 = 416 necessari.

In conclusione con la prima soluzione restano inutilizzati 1120 indirizzi con la seconda solo 288.

N.B. All'indirizzo http://vlsm-calc.net/ troverete una calcolatrice che vi consentirà di verificare la correttezza degli esercizi svolti.

Esercizi da svolgere
1.Determinare la netmask ottimale per una rete di 10 host

2.Determinare la netmask ottimale per una rete di 60 host

3.Si debba suddividere la rete 192.168.14.0 in sottoreti aventi un massimo di 10 host. Indicare, spiegando il procedimento, quale netmask è opportuno utilizzare.

4.Si debba suddividere la rete 192.168.14.0 in sottoreti aventi un massimo di 50 host. Indicare, spiegando il procedimento, quale netmask è opportuno utilizzare.

5.Si debba suddividere la rete 170.10.20.0 in sottoreti aventi un massimo di 1000 host. Indicare, spiegando il procedimento, quale netmask è opportuno utilizzare.

6.Dato l'indirizzo 198.168.40.0 con subnet mask 255.255.255.224 specificare quante sottoreti si possono ottenere, quanti host per sottorete ed, infine, elencare tutte le possibili sottoreti con l'indicazione dell'indirizzo di rete e di broadcast.

7.Dato l'indirizzo 140.10.0.0 con subnet mask 255.255.252.0 specificare quante sottoreti si possono ottenere, quanti host per sottorete ed, infine, elencare tutte le possibili sottoreti con l'indicazione dell'indirizzo di rete e di broadcast.

8.Dato l'indirizzo 198.30.40.0 partizionare la rete da esso individuata in 9 sottoreti specificando il numero di host che costituiscono ciascuna sottorete e indicare l'host 5 della sottorete 6 e l'host 2 della sottorete 4.

9.Dato l'indirizzo 160.196.0.0 partizionare la rete da esso individuata in 20 sottoreti specificando il numero di host che costituiscono ciascuna sottorete e indicare l'host 5 della sottorete 5 e l'host 7 della sottorete 8.

10. Suddividere la rete 193.168.1.0/24 in 2 sottoreti A e B aventi rispettivamente 100 e 50 host. volgere l' esercizio con subnetting (con maschera fissa)

11. Suddividere la rete 193.168.1.0/24 in 2 sottoreti A e B aventi rispettivamente 100 e 50 host. volgere l' esercizio con VLSM (con maschera variabile)

12. Suddividere la rete 193.160.10.0/24 in 4 sottoreti A, B, C e D aventi rispettivamente 60, 50, 30 e 20 host. Svolgere l' esercizio con subnetting (con maschera fissa)

13. Suddividere la rete 193.160.10.0/24 in 4 sottoreti A, B, C e D aventi rispettivamente 60, 50, 30 e 20 host Svolgere l' esercizio con VLSM (con maschera variabile)

14. Suddividere la rete 192.160.30.0/24 in 4 sottoreti A, B, C e D aventi rispettivamente 90, 40, 25 e 10 host.

15. Suddividere la rete 190.10.20.0/24 in 6 sottoreti A, B, C, D, E e F aventi rispettivamente 50, 30, 20, 20, 10 e 5 host.

16. Suddividere la rete 192.168.30.0/24 in 6 sottoreti A, B, C, D, E e F aventi rispettivamente 50, 30, 25, 2, 2, 2 host.

17. Si supponga di avere a disposizione il blocco di indirizzi IP specificato da 195.20.10.128/25 (255.255.255.128). Si supponga di avere 4 reti con le seguenti dimensioni: A 30 host, B 12 , C 6 e D 2. Indicare una possibile suddivisione del blocco di indirizzi IP a disposizione per poter assegnare gli indirizzi IP alle sottoreti usando il subnetting statico (SM a lunghezza fissa) e variabile (VLSM).

18. Un router collega tre sottoreti : A, B, C. Tutte le interfacce delle tre sottoretidevono avere Net-id 213.10.27.0/24; la sottorete A deve supportare fino a 110 host e le sottoreti B e C fino a 60 host. Fornire i Subnet-id e le maschere di sottorete nella forma a.b.c.d/x.

19. Si supponga di avere a disposizione il blocco di indirizzi IP specificato da 193.30.10.192/26 (255.255.255.192). Si supponga di avere 5 reti con le seguenti dimensioni: A 6 host, B 5, C 5, D 2, E 2. Indicare una possibile suddivisione del blocco di indirizzi IP a disposizione per poter assegnare gli indirizzi IP alle sottoreti usando il subnetting statico (SM a lunghezza fissa) e variabile (VLSM).

20. Si supponga di avere a disposizione il blocco di indirizzi IP specificato da 140.240.32.0/20 (255.255.240.0). Si supponga di avere 7 reti con le seguenti dimensioni:

Rete	Numero di hosts
A	950
B	800
C	600
D	250
E	200
F	180
E	150

Indicare una possibile suddivisione del blocco di indirizzi IP a disposizione per poter assegnare gli indirizzi IP alle sottoreti usando il subnetting.

21. Si supponga di avere a disposizione il blocco di indirizzi IP specificato da 170.16.64.0/18 (255.255.192.0). Si supponga di avere 3 reti con le seguenti dimensioni:

Rete	Numero di hosts
A	1000
B	400
C	200

Indicare una possibile suddivisione del blocco di indirizzi IP a disposizione per poter assegnare gli indirizzi IP alle sottoreti usando il subnetting.

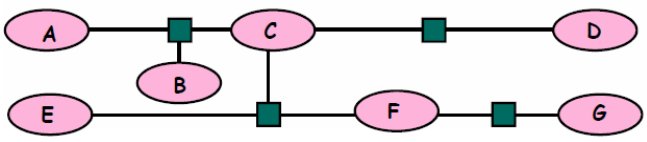
22. Si supponga di avere a disposizione il blocco di indirizzi IP specificato da 180.30.20.0/22 (255.255.252.0). Si supponga di avere 4 reti con le seguenti dimensioni:

Rete	Numero host
A	300
B	100
C	60
D	30

Indicare una possibile suddivisione del blocco di indirizzi IP a disposizione per poter assegnare gli indirizzi IP alle sottoreti usando il subnetting.

23. Si richiede il partizionamento della rete 192.168.10.0 in 4 sottoreti (A, B, C, D) con host e tre sottoreti con link punto-punto. Le sottoreti hanno rispettivamente 60, 28, 12, 12 host.

24. Si consideri topologia di rete mostrata in figura.



Supponendo che per l'intera rete si abbia a disposizione l'indirizzo di classe B: 150.200.0.0, si chiede di assegnare, a partire dall'indirizzo a disposizione, tramite la tecnica di subnetting, gli indirizzi alle 7 sottoreti (A,B,C,D,E,F,G) in modo da soddisfare le condizioni in tabella e, e ottimizzare l'efficienza di utilizzazione dello spazio degli indirizzi.

Rete	Numero host
A	30
B	30
C	220

D	220
E	10
F	70
G	1300

IP Internet Protocol

IP è un protocollo a datagrammi che:

In trasmissione:
- Riceve i dati dal livello trasporto (il 4^) e li struttura in datagrammi di massimo 64 KB;
- Instrada i pacchetti, eventualmente frammentandoli lungo il percorso.

In ricezione:
- Riassembla i frammenti in datagrammi;
- Estrae i dati del livello trasporto e li consegna nell'ordine in cui sono arrivati.

Un pacchetto IP è costituito da un header e da un area dati

La dimensione di un datagramma è variabile: infatti mentre l'header è di dimensione fissa (20 byte) l'area dati può contenere fra 1 e 65.535 byte.

Invio di datagrammi

L'header contiene le informazioni necessarie per inviare il datagramma all'host destinatario:
- Indirizzo destinazione
- Indirizzo mittente

Ogni router esamina l'header di ogni datagramma e lo invia lungo un cammino verso la destinazione.

Tabelle di routing

Le informazioni relative all'inoltro dei datagrammi sono memorizzate in *tabelle di routing* presenti appunto nei router che
- vengono create con l'inizializzazione del sistema;
- aggiornate a seguito di variazioni della rete.

Contengono una lista di reti di destinazione e la porta attraverso la quale raggiungere il nodo successivo (next hop) per ogni destinazione.

Route Table

Type	Network	Port	Next Hop IP	Metric
C	**192.168.3.0/2**	Ethernet1/1	**---**	**0/0**
C	**172.16.1.0/24**	Ethernet1/2	**---**	**0/0**
C	**10.5.5.0/24**	Ethernet1/3	**---**	**0/0**

Le tabelle sono mantenute piccole elencando le reti di destinazione invece degli host
Possono essere ulteriormente ridotte impostando una *default route*, utilizzata se la rete di destinazione non è esplicitamente elencata

Maschere di indirizzi

Come abbiamo visto i router instradano i datagrammi utilizzando l'indirizzo della rete e non l'indirizzo dell'host presente nell'header. Per identificare la rete di destinazione si opera l'AND bit

a bit dell'indirizzo dell'host di destinazione con la maschera di rete ottenendo così la rete di destinazione

190. 67. 85.121 and < --- indirizzo dell' host
255.255.255. 0 = < --- maschera di rete
190. 67. 85. 0 < --- indirizzo della rete di appartenenza dell'host 190.67.85.121

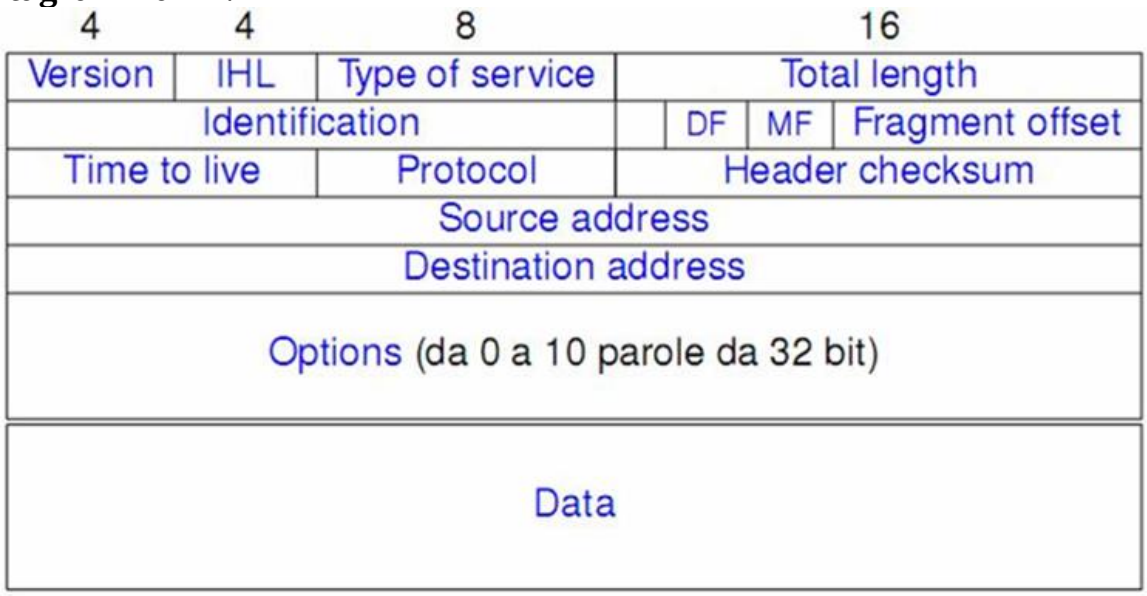
Consegna a costo minimo

• IP fornisce un servizio equivalente a una LAN
Il protocollo IP **NON garantisce** di prevenire

- Duplicazione di datagrammi
- Consegna ritardata o non ordinata
- Errori nella trasmissione dei dati
- Perdita di datagrammi

La consegna affidabile é assicurata dallo strato di trasporto (il 4^) cioè dal TCP.
Lo strato di rete (il 3^ caratterizzato dal protocollo IP) può individuare e riportare errori, ma non correggerli.
Lo strato di rete si occupa essenzialmente della consegna dei datagrammi.

Il datagramma IPv4



Version (4 bit) la versione di IP usata. Vale 4 per l'IP standard.
IHL (4 bit) IP Header Lenght; indica la lunghezza dell'header in parole di 32-bit.
Type of Service (8 bit) scarsamente adottato è costituito da:

precedence	precedence	precedence	delay	throughput	reliability	non usato	non usato

Precedence (3 bit) stabilisce l'importanza del datagramma 000 = normale 111= alta prorità.
Delay (1 bit) se settato basso ritardo
Throughput (1 bit) se settato alto throughput
Reliability (1 bit) se settato alta affidabilità
Total Length (16 bit) lunghezza totale del datagramma (header + dati) espressa in byte.

I tre campi successivi sono necessari per gestire i datagrammi in caso di frammentazione
Identification (16 bit) I frammenti di un datagramma hanno lo stesso numero di Identification.
Flags (3 bit) contiene 3 flag

1. non usato
2. **DF** (don't fragment), settato indica che il datagramma non deve essere frammentato
3. **MF** (more fragments), se resettato indica che è l'ultimo frammento.

Fragment Offset (13 bit) contiene, misurato in unità di 8 byte, *quanti dati sono presenti nei frammenti precedenti a questo*. Se il frammento è il primo o è l'unico, il valore è zero.
Time To Live (8 bit) un numero che decresce ogni volta che viene attraversato un router. Arrivato a zero il datagramma viene scartato e un messaggio ICMP viene inviato mittente. E' utile per evitare che un datagramma finisca in un loop infinito e quindi che col tempo la rete saturi.
Protocol Number (8 bit) indica quale protocollo di alto livello incapsula il datagramma. Ad esempio per ICMP è 1, per TCP è 6, per UDP è 17.
Header Checksum (16 bit) per sapere se l'header contiene errori.
Source Address/Destination Address; (32 bit) indica l'indirizzo IP dell'host sorgente/destinatario.
Option (lunghezza variabile) opzioni per l'IP;
Padding (lunghezza variabile) una serie di 0 inserita perchè la lunghezza dell'header IP sia un multiplo di 32 bit.
Data attenzione, qui non c'è solo il datagramma iniziale bisogna ricordare che IP INCAPSULA TCP quindi c'è anche l'header di TCP.

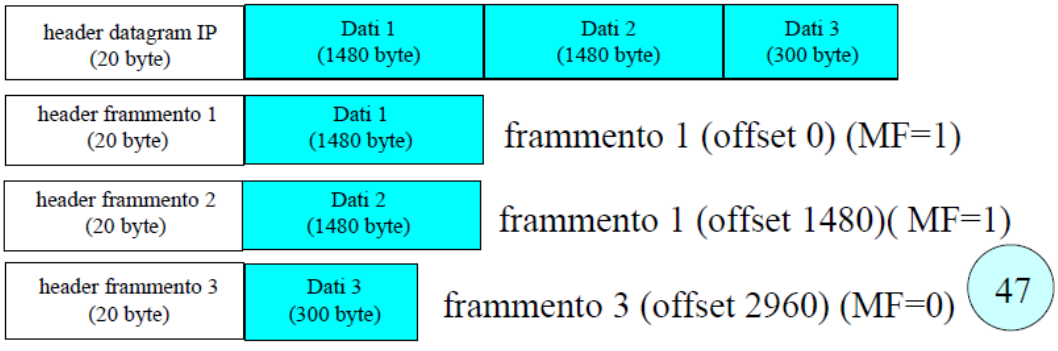
Un header senza opzioni ha HLEN = 5 (5*4 = 20 Bytes). I dati cominciano subito dopo Destination Address. Le opzioni sono aggiunte fra DESTINATION IP ADDRESS e i dati, in multipli di 32 bit.Un header con 96 bit di opzioni ha HLEN = 8(8*4 = 32 Bytes).

La frammentazione dei datagrammi IP

Un datagramma IP, quando viene passato al livello Data Link, viene incapsulato in un header Ethernet per formare un frame fisico. Il problema è che le dimensioni massime di questo frame sono limitate. Il valore di questo limite è chiamato **MTU**, **maximum transfer unit** e per Ethernet MTU=1500 bytes.

Se *Dimensione datagramma* > *MTU* si frammenta il datagramma

cioè il datagramma IP viene spezzettato in tanti frammenti. Ogni frammento possiede un proprio header simile all'header IP del datagramma originale, ma con i campi Identification, Flags, Fragment Offset settati opportunamente. Se DF = 1 il datagramma non viene frammentato ma scartato. Il flag MF = 1 in tutti i frammenti tranne l'ultimo. Il campo offset viene riempito con il valore opportuno per ogni frammento. Inoltre viene ricalcolato il contenuto di tutti quei campi dell'header IP contenenti valori variabili in funzione della lunghezza (checksum compreso).
Nell'esempio un pacchetto IP con 3260 byte con MTU di 1500 Byte.



Ora che i frammenti sono diventati nuovi datagrammi IP vengono inoltrati normalmente. Sul computer ricevente viene riassemblato il datagramma IP originale facendo uso di questi campi.

La frammentazione è piuttosto fragile: *la perdita di un solo frammento comporta la perdita dell'intero datagramma.*
In generale avere un frame molto grande permette di avere una maggiore efficienza di trasmissione (anche se nei collegamenti soggetti a problemi di lentezza e corruzione dei pacchetti, avere un MTU grande sarà solo fonte di guai, dato che ogni pacchetto perso contiene molti dati). Normalmente si cerca di evitare di dover ricorrere alla frammentazione. La lunghezza finale del datagramma dipende sia dallo spazio occupato dagli header TCP e IP, che dalla lunghezza del segmento creato da TCP. Per evitare la frammentazione bisogna tenere conto di questi valori confrontandoli con il valore di MTU minimo tra quello delle due macchine. Anche così però non possiamo avere la garanzia che non avverrà una frammentazione, perchè è sempre possibile che alcune reti intermedie abbiano limiti MTU inferiori. E in alcuni casi avverrà anche più di una frammentazione.

Comunicazione affidabile su rete

Una consegna affidabile di blocchi di dati deve:

- non deve perdere dati;
- non deve modificare i dati;
- deve mantenere i dati in ordine;
- non deve duplicare dati.

Consegna non ordinata

I datagrammi possono essere consegnati fuori ordine, specialmente in sistemi che includono più reti. Una consegna disordinata può essere individuata e corretta attraverso l'uso di un numero di sequenza. Il mittente aggiunge un numero di sequenza ad ogni pacchetto in uscita. Il ricevente usa i numeri di sequenza per riordinare i pacchetti e individuare quelli mancanti

Consegna duplicata

I pacchetti possono venire duplicati durante la trasmissione. La sequenzializzazione può essere usata per individuare i pacchetti duplicati e scartare le copie.

Pacchetti persi

Questo é il problema trasmissivo più comune. Qualsiasi errore (errore sui bit, lunghezza sbagliata) fa scartare il pacchetto dal ricevitore

Ritrasmissione

La positive acknowledgment with retransmission permette di individuare e correggere la perdita di pacchetti

- Il ricevitore invia brevi messaggi di riscontro della ricezione dei pacchetti
- Il mittente inferisce i pacchetti persi dal mancato riscontro
- Il mittente ritrasmette i pacchetti persi

Il mittente impone dei tempi massimi per ogni pacchetto

- salva una copia del pacchetto
- se passa il tempo massimo senza aver ricevuto il riscontro, si ritrasmette la copia salvata

Viene definito un numero massimo di ritrasmissioni per indicare una disconnessione della rete.

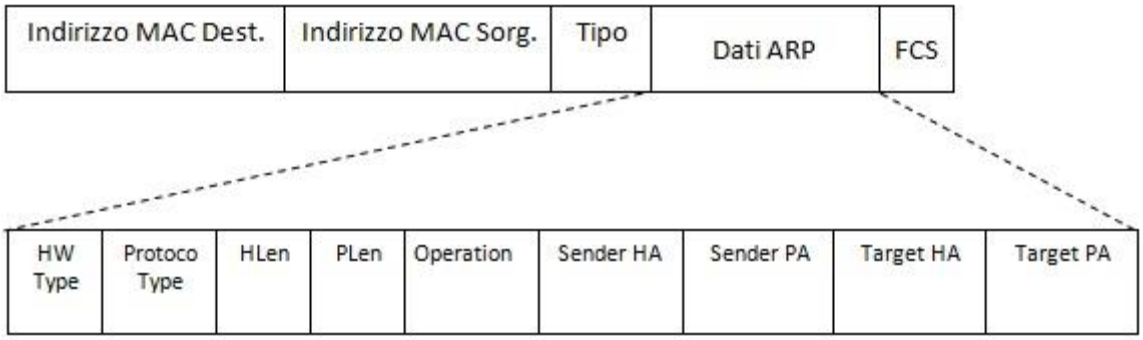
Altri protocolli del terzo livello
ICMP (Internet Control Message Protocol): Controlla l'operatività delle subnet, tipi di messaggi:

- Destination unreachable
- Time exceeded - time to live ha raggiunto 0
- Redirect - routing
- Echo request, reply
- Time stamp request, reply

ARP (Address Resolution Protocol): conosciuto l'indirizzo IP dell'host di destinazione, ci consente di conoscere il MAC address dell'interfaccia dell'host.
RARP (Reverse Address Resolution Protocol): dal MAC address trova quale indirizzo IP che gli corrisponde.

Il protocollo ARP

Sappiamo che ogni interfaccia di rete dispone di un indirizzo fisico impresso durante il processo di costruzione in fabbrica. *Il compito del protocollo ARP è quello di identificare il MAC Address a cui è associato l'indirizzo IP a cui il livello network vuole trasmettere i dati*.
Un frame ARP viene incapsulato in un frame IEEE802 :



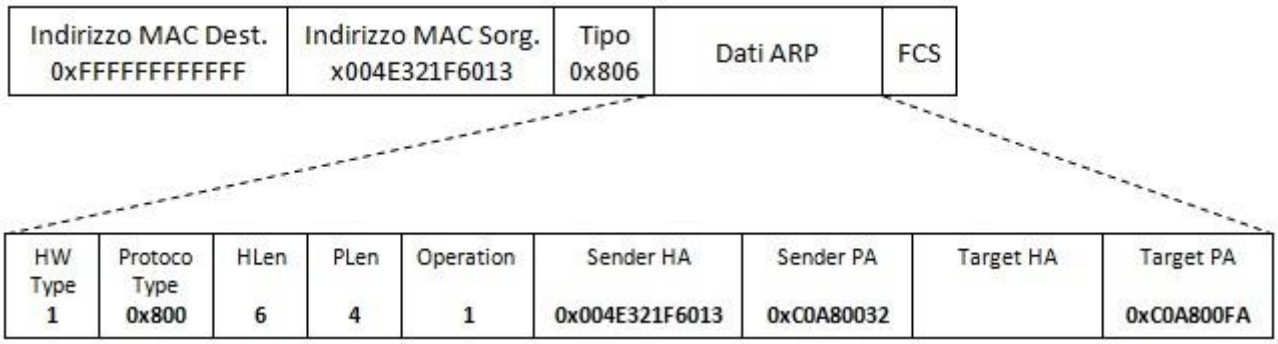
Mancano nella rappresentazione Preamble e SFD poiché non fanno propriamente parte del frame. Il campo **Tipo** in caso Ethernet incapsiuli ARP assume il valore **0×806**.
I campi specifici del protocollo ARP sono :
HW Type: specifica la tipologia di indirizzo fisico presente nel frame. Il valore 1 indica un indirizzo fisico di tipo ethernet;
Protocol Type: indica il protocollo di livello network al quale appartiene l'indirizzo di cui vogliamo conoscere il corrispondente MAC Address . Il valore **0×800** indica il protocollo IP;
HLen: lunghezza in byte dell'indirizzo fisico (6 byte per MAC) ;
PLen: lunghezza in byte dell'indirizzo logico (4 byte per IP);
Operation: assume valore 1 se si tratta di richiesta di informzioni (ARP_request), valore 2 se si tratta di una risposta (ARP_reply);
Sender HA: indica il MAC address dell'interfaccia del mittente;
Sender PA: indica l'indirizzo di IP dell'interfaccia del mittente;
Target HA: indica il MAC address dell'interfaccia del destinatario;
Target PA: indica l'indirizzo di IP dell'interfaccia del destinatario.

Supponiamo che un host A debba comunicare con B, ne conosce l'IP address ma non il MAC address. Le configurazioni dei due host sono:

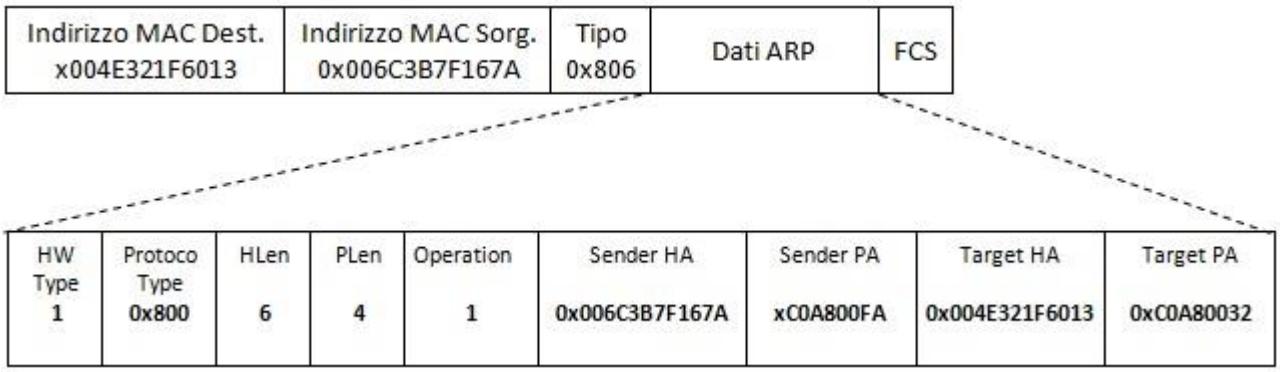
HOST A
MAC Addr.: 00:4E:32:1F:60:13
Indirizzo IP: 192.168.0.50

HOST B
MAC Addr.: 00:6C:3B:7F:16:7A
Indirizzo IP: 192.168.0.200

L' host A costruisce il seguente frame ARP.



Analizziamo il contenuto dei singoli campi del frame:
MAC Dest.: ha il valore 0xFFFFFFFFFF. Vi faccio notare che tutti i frame ARP_request utilizzano l'indirizzo di broadcast 0xFFFFFFFFFF e, pertanto, la richiesta ARP verrà inviata a tutte le periferiche presenti sulla rete;
MAC Sorg.: ha il valore 0x004E321F6013 e corrisponde all'indirizzo MAC della postazione che ha originato la richiesta;
Tipo: ha il valore 0×806 ad indicare che si tratta di un frame di tipo ARP;
I campi seguenti sono specifici del protocollo ARP:
HW Type: ha il valore 1 per indicare che il protocollo data-link è ethernet;
Protocol Type: ha il valore 0×800 per indicare che il protocollo network è l'Internet Protocol (IP);
HLen: ha il valore 6 per indicare che il protocollo ethernet utilizza indirizzi di 6 byte;
PLen: ha il valore 4 per indicare che il protocollo IP utilizza indirizzi di 4 byte;
Operation: ha il valore 1 per indicare che si tratta di una richiesta ARP;
Sender HA: ha il valore 0x004E321F6013 e corrisponde all'indirzzo ethernet della postazione trasmittente;
Sender PA: ha il valore 0xC0A80032 e corrisponde all'indirizzo IP del mittente espresso in esadecimale.
Target HA: è ***vuoto poiché è proprio il campo che si vuole conoscere con questa richiesta ARP***;
Target PA: ha il valore 0xC0A800FA è corrisponde all'indirizzo IP del destinatario, ovvero l'indirizzo IP del quale si vuole conoscere il corrispondente indirizzo fisico.
Quando le postazioni presenti sulla rete riceveranno il frane ARP andranno a controllare il campo Operation e capiranno che si tratta di una richiesta ARP. Il passo successivo è quello di verificare il campo Target HA per controllare se coincide con il loro indirizzo MAC. Vi ricordo che gli indirizzi MAC sono univoci sulla rete e che solo una postazione potrà trovare un riscontro positivo. Le postazioni il cui indirizzo MAC non corrisponde al campo Target HA scarteranno semplicemente il frame.
L' host B, il cui indirizzo MAC corrisponde al campo Target PA, risponde alla richiesta creando il seguente frame ARP.



Operation: ha il valore 2 per indicare Arp-reply;
il campo Target HA dell' frame Arp-request assumerà valore uguale al MAC address ell'interfaccia di host B, e quindi i campi target HA e PA mittentre e destinatario vengono invertiti, ed il frame

inviato all'host A. La postazione A ricevuto l'ARP_reply estrapolerà l'indirizzo fisico MAC, per il quale aveva inviato la richiesta, dal campo Sender HA e creerà il frame ethernet per inviare il datagramma IP.

La tabella ARP

Per rendere il protocollo ARP più efficiente ogni postazione conserva una tabella ARP in cui vengono inseriti gli indirizzi MAC ed i corrispondenti indirizzi IP. Questa tabella viene aggiornata sia con le ARP_request che con le ARP_reply.

Il comando arp –a ci consente di conoscere il contenuto di tale tabella:

```
C:\Windows\system32>arp -a

Interfaccia: 192.168.1.10 --- 0xa
  Indirizzo Internet    Indirizzo fisico      Tipo
  151.45.170.33         00-25-9c-88-f4-a7     dinamico
  192.168.1.1           00-25-9c-88-f4-a7     dinamico
  192.168.1.255         ff-ff-ff-ff-ff-ff     statico
  224.0.0.22            01-00-5e-00-00-16     statico
  224.0.0.252           01-00-5e-00-00-fc     statico
  224.0.1.60            01-00-5e-00-01-3c     statico
  226.178.217.5         01-00-5e-32-d9-05     statico
  239.255.255.250       01-00-5e-7f-ff-fa     statico
```

Gratuitous ARP

Il Gratuitous ARP, un frame ARP che viene trasmesso quando viene attivata l'interfaccia di rete di un host. Si tratta di una ARP_request con cui la postazione chiede il possessore del *proprio indirizzo* IP. Il Gratuitous ARP ha una duplice funzione:

- scoprire se l'indirizzo IP è già utilizzato sulla rete. L'host non si aspetta alcuna risposta per la richiesta poiché l'indirizzo IP è il suo così se dovesse ricevere una ARP_reply notificherà al sistema operativo che l'indirizzo IP è già utilizzato;
- aggiornare la tabella ARP di tutti gli host sulla rete.

Il protocollo ICMP

Il protocollo IP non garantisce una corretta consegna dei datagrammi se necessario si affida a protocolli affidabili dei livello superiore (TCP)

E' quindi necessario un protocollo di controllo per :

- gestire situazioni anomale
- notificare errori o irraggiungibilità della destinazione
- in generale scambio di informazioni sulla rete
- controllo della congestione del flusso dei datagrammi IP.
- comunicare periodicamente i cambiamenti delle tabelle di routing.
- determinare cammini circolari o troppo lunghi
- stima del tempo di trasmissione tra mittente e destinatario

Questo protocollo è l' ICMP (Internet Control Message Protocol). ICMP Segnala errori e malfunzionamenti ma non esegue

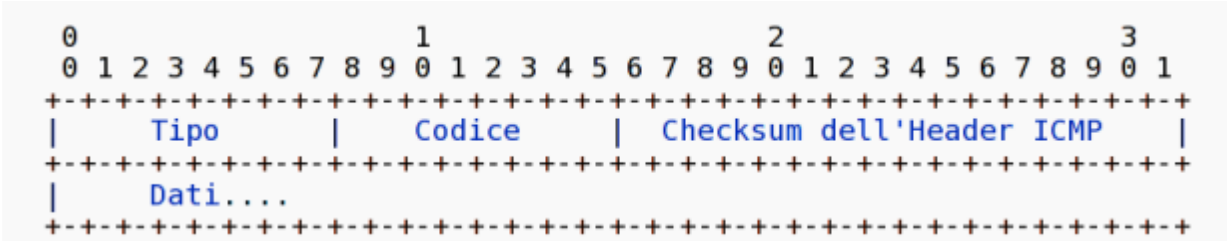
correzioni quindi non rende affidabile IP. L' ICMP svolge funzioni di controllo per IP, quindi offre un servizio ad IP.

I pacchetti ICMP sono incapsulati in datagrammi IP.

Nel generare pacchetti ICMP vergono seguite queste tre regole

1. Nessun messaggio ICMP viene generato a seguito di errori rilevati su messaggi ICMP
2. Se un pacchetto viene frammentato solo il primo frammento può generare messaggi ICMP
3. Broadcast e multicast non generano ICMP

Il formato di un pacchetto ICMP è:



Principali messaggi di errore

TIME_EXCEEDED tipo = 11 (tempo scaduto).

Viene inviato quando un pacchetto arriva ad un sistema intermedio con un TTL = 0 (codice = 0). Può essere mandato anche quando un pacchetto viene frammentato da un gateway, ma la macchina che si occupa di ricomporlo, non riesce a portare a termine l'operazione entro un ragionevole lasso di tempo (codice = 1).

DESTINATION_UNREACHABLE tipo = 3

L'host di destinazione è irraggiungibile. Nel pacchetto il campo CODICE specifica una causa tra le seguenti:

- ad un gateway intermedio la rete destinazione può risultare a distanza infinita (net unreachable).
- l'host può essere spento, o comunque non rispondere ad una chiamata ARP (host unreachable).
- l'host può rifiutare il pacchetto perché non conosce il protocollo che viaggia a bordo del pacchetto stesso (protocol unreachable

). In questo caso il messaggio ICMP viene mandato alla sorgente dallo stesso host-destinazione.

- il pacchetto potrebbe fermarsi lungo la via perché per passare dovrebbe essere frammentato, ma questo non è possibile a causa del suo DON'T_FRAGMENT_FLAG posto al valore ON, che appunto lo proibisce (fragmentation needed and DF set).

REDIRECT tipo = 5 (ridireziona).

Un gateway intermedio si accorge che il prossimo gateway cui dovrebbe inoltrare il pacchetto sta sulla stessa sottorete del mittente, e avvisa quest'ultimo dell'esistenza di una via più breve per la destinazione.

PARAMETER_PROBLEM (problema con i parametri).

Il gateway ha ricevuto un pacchetto IP il cui valore del checksum è corretto , ma non riesce ad interpretare i valori dei parametri. Questo messaggio viene mandato alla sorgente del pacchetto solamente se il problema ha causato la perdita del pacchetto stesso.

Principali messaggi di informazione

ECHO_REQUEST tipo = 8 (richiesta di echo).

Un host chiede alla macchina destinazione di rimandare indietro lo stesso pacchetto.

ECHO_REPLY tipo = 0 (risposta ad una richiesta di echo).

La macchina destinazione di un precedente ECHO_REQUEST rimanda indietro il pacchetto ricevuto (dopo aver scambiato il campo sorgente con quello destinazione e dopo aver modificato il campo che specifica se si tratti du un request o di un reply).

INFORMATION_REQUEST e INFORMATION_REPLY. tipo = 15 e 16

Ricordiamo qui che alcuni indirizzi IP hanno significati convenzionali:
00000000.00000000.00000000.00000000 Questa macchina
000000 .... 0000 host-number la macchina host-number di questa rete
11111111.11111111.11111111.11111111 broadcast su questa rete
network-number 111 .... 11111 broadcast sulla rete network-number
127 qualsiasi altra cosa loopback (il pacchetto non esce nemmeno sulla rete)
Il messaggio ICMP INFORMATION_REQUEST è uno dei rari casi in cui si usa l'indirizzo 0.0.0.host-number:
l'host sorgente pone gli indirizzi della sorgente e della destinazione entrambi a zero per quanto riguarda l'indirizzo di rete, vincolando così il pacchetto alla rete locale (un eventuale router non lo manderebbe fuori). Il destinatario del pacchetto, nel rispondere con un INFORMATION_REPLY,

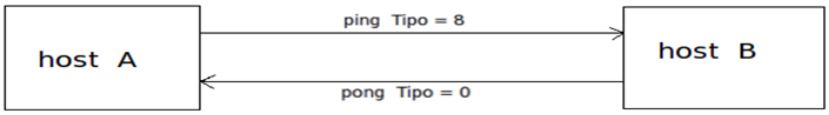
completa entrambi gli indirizzi con il giusto indirizzo di rete, scambia l'indirizzo della sorgente con quello della destinazione, ricomputa il checksum e rimanda indietro il pacchetto. In questo modo la macchina sorgente ha acquisito il numero IP della rete in cui si trova.
NOTA: la macchina sorgente deve conoscere a priori la netmask della rete, altrimenti non avrebbe la possibilità di sapere dei 32 bit quali riguardano la rete (quindi da mettere a 0) e quali gli host.
TIMESTAMP REQUEST e TIMESTAMP_REPLY. tipo = 13 e 14
La stazione trasmittente invia un messaggio Timestamp Request in cui riempie il campo Timestamp Origine con l'ora dell'invio. La destinazione risponde con un messaggio Timestamp Reply in cui riempie il campo Timestamp Ricezione con l'ora di ricezione della richiesta e il campo Timestamp Trasmissione con l'ora di trasmissione del responso. La stazione trasmittente nota l'ora di ricezione del messaggio responso e calcola lo sfasamento tra le due stazioni. I campi ora sono espressi in millisecondi a partire dalla mezzanotte scorsa GMT. Se la risoluzione dell'ordine dei millisecondi non e' disponibile, il bit piu' significativo del campo e' settato. I messaggi Timestamp non sono piu'e usati in Internet, sostituiti dal protocollo Network Time Protocol (NTP).
ADDRESS MASK REQUEST e ADDRESS MASK REPLY. tipo = 17 e 18
Inviato dall'host sorgente all' indirizzo 255.255.255.255 per ottenere la subnet mask da utilizzare con l'indirizzo di IP ricevuto da BOTP o RARP.

Tipo	**Codice**	**Nome**	**Chi lo utilizza**
0		echo-reply	risposta a un ping (pong)
1			
2			
3		destination-unreachable	traffico TCP e UDP
3	0	network-unreachable	
3	1	host-unreachable	
3	2	protocol-unreachable	
3	3	port-unreachable	
3	4	fragmentation-needed	
3	5	source-route-failed	
3	6	network-unknown	
3	7	host-unknown	
3	8		
3	9	network-prohibited	
3	10	host-prohibited	
3	11	TOS-network-unreachable	
3	12	TOS-host-unreachable	
3	13	communication-prohibited	
3	14	host-precedence-violation	
3	15	precedence-cutoff	
4		source-quench	
5		edirect	instradamento dei pacchetti
5	0	network-redirect	
5	1	host-redirect	

Tipo	Codice	Nome	Chi lo utilizza
5	2	TOS-network-redirect	
5	3	TOS-host-redirect	
6			
7			
8		echo-request	ping
9		router-advertisement	
10		router-solicitation	
11		time-exceeded (ttl-exceeded)	traceroute
11	0	ttl-zero-during-transit	
11	1	ttl-zero-during-reassembly	
12		parameter-problem	
12	0	ip-header-bad	
12	1	required-option-missing	
13		timestamp-request	
14		timestamp-reply	
15		information-request	
16		information-reply	
17		address-mask-request	
18		address-mask-reply	

Applicazioni ICMP: Il comando PING

Supponiamo che dalla stazione A si voglia controllare l'integrità della connessione fino alla stazione B. Si esegue il comando ping, passandogli come argomento l'indirizzo della stazione B. Il programma manda una serie di messaggi ICMP ECHO_REQUEST (generalmente uno al secondo) dalla stazione A verso la stazione B. Quando la stazione B riceve un pacchetto ECHO_REQUEST, il suo strato internet si occupa di rispondere con un nuovo datagramma ICMP ECHO_REPLY, che viene mandato indietro alla macchina A. il programma ping userà le informazioni così collezionate (esistenza dei pacchetti di ritorno, tempo intercorso per ogni pacchetto, etc.) per calcolare dei valori statistici sulla bontà della connessione e presentarli all'utente.



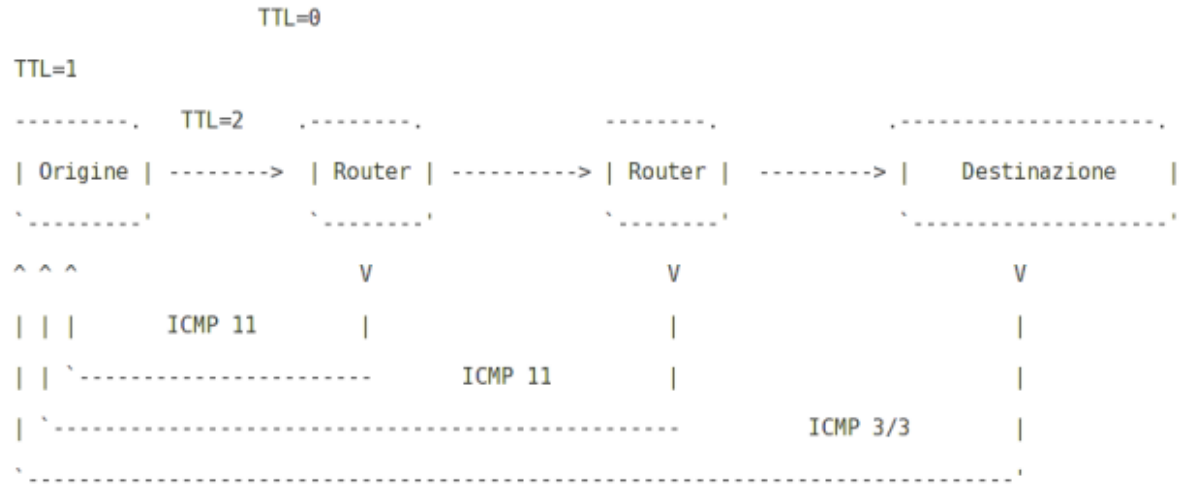
```
C:\Windows\system32>ping libero.it

Esecuzione di Ping libero.it [151.1.67.216] con 32 byte di dati:
Risposta da 151.1.67.216: byte=32 durata=21ms TTL=245
Risposta da 151.1.67.216: byte=32 durata=20ms TTL=245
Risposta da 151.1.67.216: byte=32 durata=20ms TTL=245
Risposta da 151.1.67.216: byte=32 durata=21ms TTL=245

Statistiche Ping per 151.1.67.216:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 20ms, Massimo =  21ms, Medio =  20ms
```

I pacchetti Ping, sia di richiesta di eco, sia di risposta, possiedono un numero di ordine, icmp_seq, che consente di verificare con quale sequenza vengono restituiti i pacchetti di risposta dalla rete.

Applicazioni ICMP: Il comando Traceroute

Lo scopo del traceroute (tracert, trace) è quello di segnalare quali siano le macchine attraversate dai pacchetti per giungere ad una determinata destinazione, nonché di utilizzare tutte le informazioni disponibili per valutare l'affidabilità della connessione.
A invia a B una serie di pacchetti ICMP di tipo ECHO_REQUEST con TTL progressivo da 1 a 30
TTL viene decrementato da ciascun nodo intermedio
Il nodo che rileva TTL = 0 invia ad A un pacchetto ICMP di tipo TIME_EXCEEDED
A costruisce cosi una lista dei nodi attraversati per arrivare a B
L'output mostra TTL, nome di DNS e IP ed il ROUND TRIP TIME (RTT)
In sintesi A attende i TIME_EXCEEDED, mandati indietro dai nodi intermedi lungo il percorso del pacchetto. I primi pacchetti lanciati avranno un TTL di un solo hop, poi questo valore viene incrementato finché non si raggiunge il numero massimo di hops, oppure finché la stazione destinazione, raggiunta dal pacchetto, non risponde con un DESTINATION_UNREACHABLE di tipo port_unreachable per segnalare che non sa cosa fare del pacchetto ricevuto (riferito ad una porta logica non attivata). Per ogni valore di TTL il traceroute manda tre pacchetti (ma il loro numero può essere modificato dall'utente), e mostra i seguenti valori:

- TTL dei pacchetti (espresso in numero di hops)
- indirizzo del gateway che ha rimandato indietro i pacchetti ICMP TIME_EXCEEDED
- per ogni pacchetto il tempo intercorso tra la sua spedizione e la sua ricezione (o un asterisco se questo è maggiore di 3 secondi)



```
C:\Windows\system32>tracert libero.it

Traccia instradamento verso libero.it [151.1.67.216]
su un massimo di 30 punti di passaggio:

  1    <1 ms    <1 ms    <1 ms  192.168.1.1
  2     9 ms     9 ms     9 ms  151.23.225.10
  3     9 ms    14 ms    10 ms  10.0.41.1
  4    42 ms     8 ms     9 ms  151.6.238.65
  5    22 ms    22 ms    22 ms  151.6.4.125
  6    21 ms    21 ms    21 ms  151.6.4.201
  7    21 ms    20 ms    19 ms  151.7.84.10
  8    20 ms    20 ms    20 ms  151.7.85.173
  9    20 ms    20 ms    20 ms  172.31.0.19
 10    20 ms    21 ms    20 ms  172.31.0.3
 11    20 ms    20 ms    20 ms  vhp-d6.rmce.libero.it [151.1.67.216]

Traccia completata.
```

Il protocollo IPv6

IPv6 è la versione dell'Internet Protocol che succede a IPv4. Questo protocollo introduce alcuni nuovi servizi e semplifica molto la configurazione e la gestione delle reti IP.

- IPv6 è un sistema a 128 bit rispetto ai 32 di IPv4 per cui
  IPv4 consente circa 4,3 miliardi di indirizzi
  IPv6 3,4 * 1038 indirizzi
- IPV6 incorpora altri protocolli come l' ARP
- IPv6 configura automaticamente il gateway di default ed altri parametri
- Supporta nativamente il QOS (Quality of Service)
- Introduce l'anycast che permette ad un host di raggiungere il più vicino server disponibile. (es. DNS).
- header di lunghezza fissa (40 byte);
- pacchetti non frammentabili dai router;
- eliminazione del campo checksum, ridondante perché presente in altri layer.

Queste ultime tre caratteristiche migliorano l'instradamento e il numero di pacchetti instradati al secondo dei router (throughput). Insieme all'IPv6 inoltre viene definito anche l'ICMPv6, molto simile all'ICMPv4.

Il datagramma IPv6

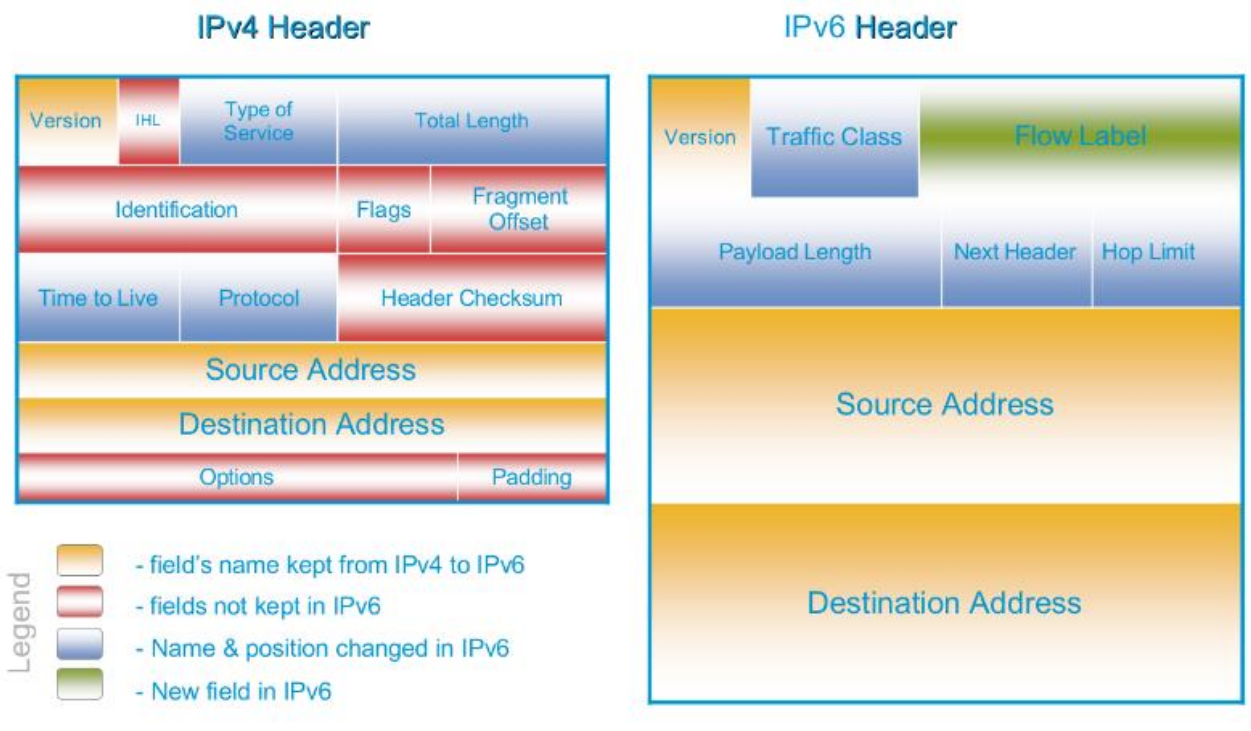
Il datagramma IPv6, come ogni altro datagramma, si compone di due parti principali: l'**header** e il **payload**. L'header è costituito dai primi 40 byte del pacchetto e contiene 8 campi, 5 in meno rispetto all'IPv4.

- **Version** [4 bit] - Indica la versione del datagramma IP.

- **Traffic Class** [8 bit] - Viene usato per il controllo della congestione. Alle risorse capaci di saturare la banda saranno attribuite delle priorità da 0 a 7 in caso di congestione. I valori da 8 a 15 sono assegnati ad esempio allo streamming audio e video.
- **Flow Label** [20 bit] - Usata dal mittente per etichettare una sequenza di datagrammi per indicarne l'apparetnenza ad uno stesso flusso. E' l'implementazione nativa del QoS (Quality of Service) in IPv6, consentendo ad esempio di specificare quali etichette (flussi di dati) abbiano una priorità rispetto ad altre (ad esempio VoIP estrammig video).
- **Payload Length** [16 bit] - È la dimensione del payload, ovvero il numero di byte di tutto ciò che viene dopo l'header.
- **Next Header** [8 bit] - simile al campo protocol dell'header IPv4, del quale usa gli stessi valori.
- **Hop Limit** [8 bit] - È il limite di salti consentito, praticamente il Time to live. Il suo valore viene decrementato di 1 ogni volta che il datagramma passa da un router: quando arriva a zero viene scartato.
- **Source Address** [128 bit] - Indica l'indirizzo IPv6 del mittente del datagramma.
- **Destination Address** [128 bit] - Indica l'indirizzo IPv6 del destinatario del datagramma.

La parte dati (payload in inglese) ha una lunghezza minima di 1280 byte o 1500 byte se la rete supporta un MTU variabile. Il payload può raggiungere i 65.535 byte in modalità standard o può essere di dimensioni maggiori in modalità "jumbo payload".

Confronto fra i header IPv4 e IPv6



La transizione all'IPv6

Il piano di transizione per trasformare la rete Internet basata su IPv4 in quella basata su IPv6 risale al luglio del 2007. Nella realtà molti vecchi calcolatori rimarranno online senza venire aggiornati, e macchine IPv6 ed IPv4 dovranno convivere sulla rete per decenni. Il meccanismo adottato per gestire questo periodo transitorio è il cosiddetto **dual stack**: ogni sistema operativo supporta ambedue gli stack IPv6 e IPv4. Quando un host si dovrà connettere ad un altro, il server DNS la informerà su quale stack dovrà usare.

Vantaggi:

- Transizione morbida: salvaguardia degli investimenti;
- Piena compatibilità fra vecchie e nuove macchine e applicazioni;

Svantaggi:

- Necessità di supportare IPv4 in Internet e negli apparati connessi.
- Per essere raggiungibili dall'universo IPv4 durante la fase di transizione costringe a mantenere un indirizzo IPv4 o una qualche forma di NAT nei router. Si aggiunge quindi un livello di complessità.

Altri metodi che è possibile utilizzare sono:

Tunneling - tunneling è un metodo di trasporto di un pacchetto IPv6 su una rete IPv4. Il pacchetto IPv6 viene incapsulato all'interno di un pacchetto IPv4 con il valore **41** nel campo protocol..

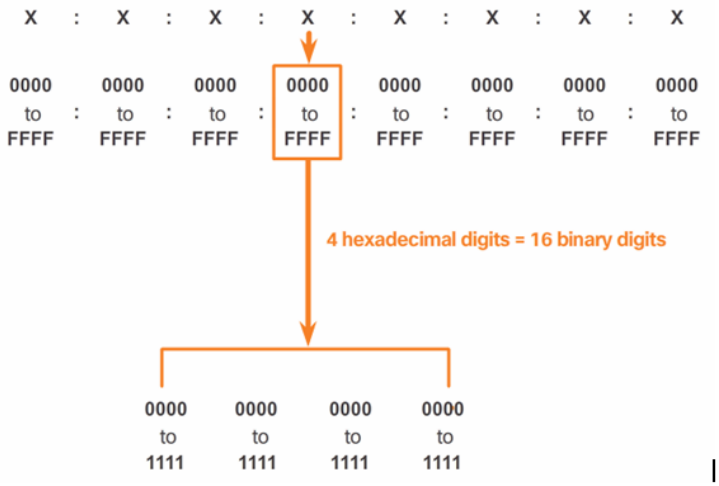
Translation - Viene utilizzato il NAT 64 e quindi un pacchetto IPv6 è tradotto in un pacchetto IPv4 e viceversa.

Nota: Tunneling e la traslation sono usati solo dove necessario. L'obiettivo dovrebbe essere comunicazioni IPv6 native dalla sorgente alla destinazione.

Rappresentazione di un indirizzo IPv6

Un indirizzo IPv6 è lungo 128 bit ed è rapprentato come una stringa di cifre esadecimali. Sappiamo che ogni 4 bit si ha una cifra esadecimale quindi 128 / 4 = 32 cifre esadecimali. Infine le 32 cifre esadecimali vengono divise da un : ogni 4 quindi avremo:

32 / 4 = 8 **hextet**



Preferred Format

Questo formato per la scrittura di un indirizzo IPv6 è x: x: x: x: x: x: x: x, con ogni "x" composto da quattro valori esadecimali. Un **hextet** è il termine ufficiale utilizzato per riferirsi a un segmento di 16 bit o quattro cifre esadecimali.

Possiamo ottenere una rappresentazione "compatta" di un indirizzo IPv6 applicando due regole:

1 - togliere gli 0 non significativi in uno o più hextet

2 - togliere ***uno o più*** hextet a 0 ***consecutivi*** e sostituirli con :: (doppio due punti) ***una sola volta,*** per non creare ambiguità nel ricostruire l'indirizzo completo.

Quindi 2001: 0DB8 :: ABCD ::1234 è errato mancano 4 hextet a 0 dove li mettiamo ?

Sintetizzando potremo dire che :

- gli zeri iniziali possono essere omessi in ogni gruppo

- se un gruppo è 0000 allora posso mettere un solo zero
- se abbiamo uno o più gruppi di 0000 possono essere sostituiti da **::** (una sola volta)

Gli indirizzi IPv4 sono facilmente trasformabili in formato IPv6 ponendo
- 80 bit a 0
- 16 bit a 1
- 32 bit sono l'indirizzo IP in notazione decimale puntata.

Quindi l'indirizzo IPv4 **10.254.254.1** corrisponde all indirizzo IPv6 **::FFFF: 10.254.254.1**
Ed è detto *IPv4-compatible address.*

Come esempio vengono riportate varie rappresentazioni dello stesso indirizzo:
2001:0db8:0000:0000:0000:0000:1428:57ab
2001:0db8:0000:0000::1428:57ab
2001:0db8:0:0:0:0:1428:57ab
2001:0db8:0::0:1428:57ab
2001:0db8::1428:57ab

Tipologie di di indirizzi IPv6:

Unicast - Un indirizzo IPv6 unicast identifica in modo *univoco* una interfaccia su un dispositivo abilitato per IPv6.

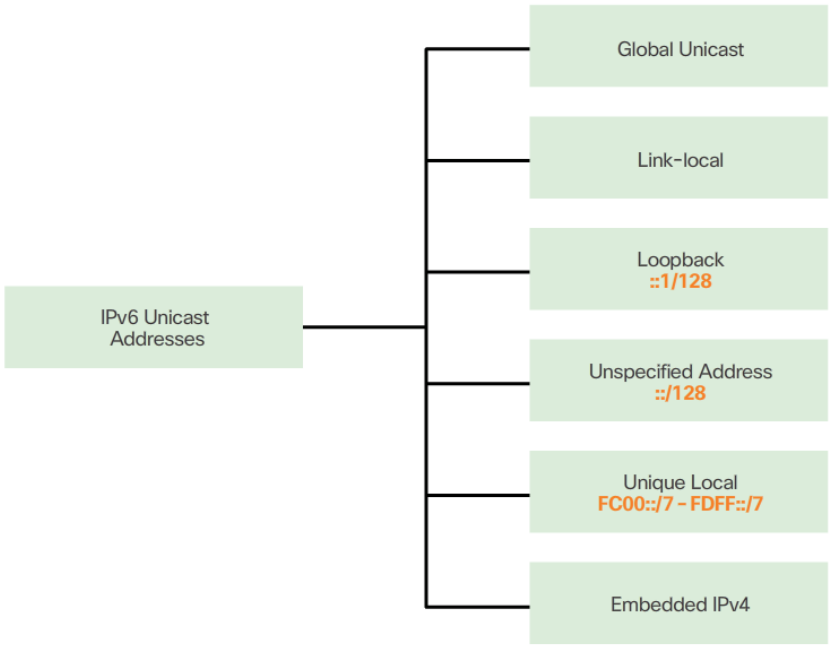
Multicast - Un indirizzo IPv6 multicast viene utilizzato per inviare un singolo pacchetto IPv6 a *più destinazioni*.

Anycast - Un indirizzo anycast IPv6 *è un qualsiasi indirizzo unicast IPv6 che può essere assegnato a più interfacce di diversi dispositivi*. Un pacchetto inviato a un indirizzo anycast viene instradato al dispositivo più vicino con tale indirizzo. Infatti lo scopo di un indirizzo anycast è quello di poter raggiungere il dispositivo che *risponde prima* (il più vicino).
A differenza di IPv4, IPv6 non dispone di un indirizzo di broadcast, comunque lo stesso risultato può essere ottenuto inviando un pacchetto al gruppo multicast tutti i nodi di un link-local.

IPv6 e il Net prefix
Sappiamo che per individuare la rete di destinazione con IPv4 mettevamo in AND l'indirizzo dell' host con la maschera di rete o il net-prefix.
192.168.1.10 con net mask 255.255.255.0 è equivalente a 192.168.1.10 / 24.
IPv6 utilizza il net- prefix per indicare la porzione di rete di un indirizzo IPv6.
La lunghezza del net-prefix può variare da 0 a 128. Una tipica lunghezza del prefisso IPv6 per reti LAN e molti altri tipi di reti è / 64. Questo significa che il prefisso o rete porzione dell'indirizzo è di 64 bit di lunghezza, lasciando altri 64 bit per l'ID di interfaccia (parte host) dell'indirizzo.

Indirizzi unicast IPv6

Un indirizzo unicast IPv6 identifica in modo univoco una interfaccia su un dispositivo abilitato per IPv6. Un pacchetto inviato a un indirizzo unicast viene ricevuto dall'interfaccia che ha assegnato quell' indirizzo. Un indirizzo IPv6 mittente deve essere un indirizzo unicast. L'indirizzo IPv6 di destinazione può essere un unicast o multicast.



Indirizzi speciali
Alcuni indirizzi hanno un significato particolare:

- **::1/128** indirizzo di **loopback** corrisponde a 127.0.0.1 in IPv4;
- **FE80::/10** prefisso link-local specifica che l'indirizzo è valido sullo specifico link fisico;
- **FF00::/8** il prefisso di multicast è utilizzato per gli indirizzi di multicast.
- **::/128** composto da tutti zeri indica *un qualsiasi indirizzo* e viene usato solo a livello software;
- **::/96** utilizzato per interconnettere le due tecnologie IPv4 / IPv6 nelle reti ibride;

I tipi più comuni di indirizzi unicast IPv6 sono:

Global Unicast Address (GUA)
Un indirizzo unicast globale è simile a un indirizzo IPv4 pubblico.

- è unico a livello globale
- è instradabile *a livello globale* da un router

I GUA possono essere configurati in modo statico o dinamico.

Indirizzi unicast Link-Local

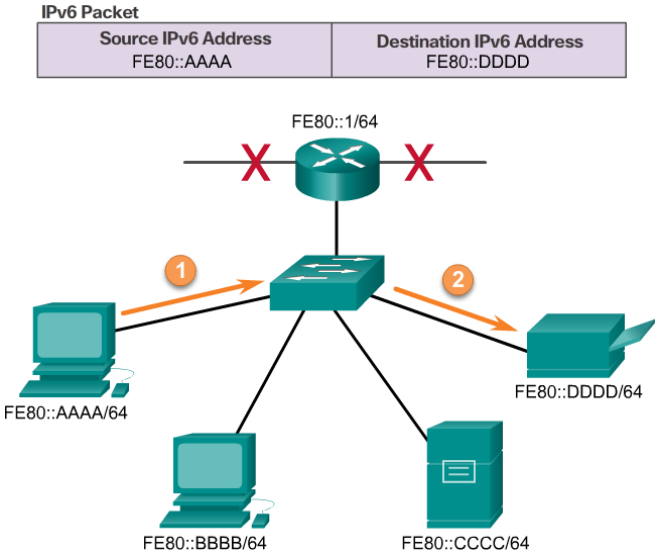
Per link-local si intende sottorete, quindi gli indirizzi link-local sono utilizzati per comunicare con altri dispositivi sullo stesso link-local .

- è unico a livello link-local (sottorete)
- è instradabile solo al livello link-local

Una interfaccia può non avere un GUA ma dovrà ***sempre avere un link-local address***. Se non configurato manualmente il dispositivo creerà automaticamente il link-local address per l'interfaccia (senza l'uso di un DHCP).
In questo modo tutti i dispositivi di una sottorete (link-local) potranno comunicare.
Gli indirizzi unicast link-local sono nell'intervallo:

FE80 :: / 10 a **FEBF :: / 10**

La / 10 indica che i primi 10 bit sono a 1 quindi
1111 1110 1000 0000 (FE80) è il limite sinistro dell'intervallo
1111 1110 1011 1111 (FEBF) è quello destro.



Unique local (ULA)

Gli unique local IPv6 hanno qualche somiglianza con indirizzi privati IPv4 (RFC 1918), ma a differenza di essi sono usati per l'indirizzamento locale *all'interno di un sito o tra un numero limitato di siti (* per sito si intende una o più reti sotto un unica amministarzione *)* . Questi indirizzi non dovrebbero essere instradabili e *non può essere nattato in un indirizzo IPv6 globale*. Gli indirizzi ULA sono compresi nell'intervallo :

FC00 :: / 7 a **FDFF :: / 7**.

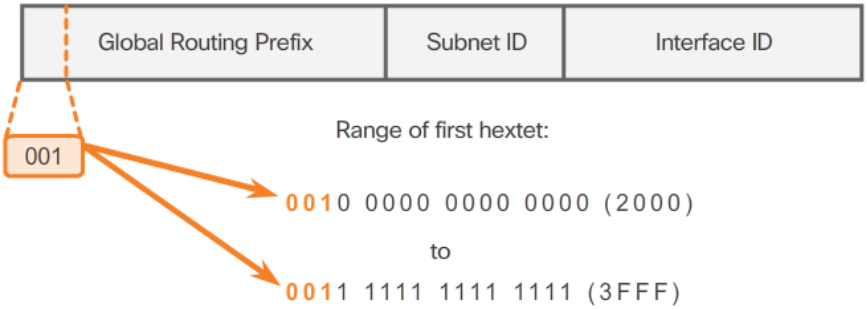
Gli indirizzi unique local possono essere utilizzati per i *dispositivi che non avranno mai bisogno o che non hanno l'accesso a un'altra rete.*

Struttura di un Global Unicast Address

Sono indirizzi equivalenti agli indirizzi IPv4 pubblici e sono:

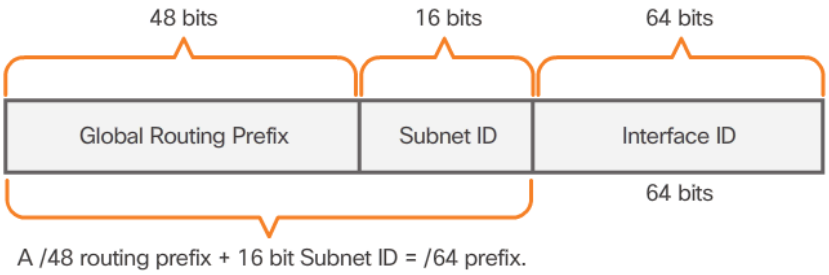
- unici a livello globale
- instradabili

Il Comitato Internet Assigned Names e Numbers (ICANN), l'operatore per IANA, assegna indirizzi IPv6 blocchi ai cinque RIR. Attualmente vengono assegnati come indirizzi global unicast quelli che iniziano con i primi 3 bit a 001.



Nota: Il 2001:0DB8 :: / 32 è stato riservato a fini di documentazione, compreso l'uso di esempi.
Un GUA si divide in tre parti:

Global Routing Prefix
E' la parte dell'indirizzo che viene assegnato dal provider, a un cliente o di un sito ed indica la rete di appartenenza. In genere viene assegnato un /48.



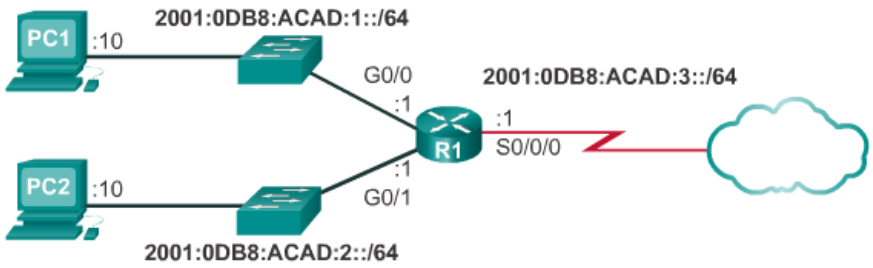
Ad esempio, l'indirizzo IPv6 **2001: 0DB8: ACAD :: / 48**
ha un prefisso che indica che i primi 48 bit (3 hextets) sono il global routing prefix o rete. *La dimensione del prefisso di routing globale determina la dimensione dell'subnet ID.*
E' consigliabile usare per la parte rete /64.

Subnet ID
L'ID subnet è utilizzato da un'organizzazione per identificare le sottoreti all'interno di un sito. Più grande è la ID di sottorete, più sono le sottoreti disponibili.

Interface ID
L' Interface ID è equivalente alla parte di host di un indirizzo IPv4. La sua denominazione è derminata dal fatto che un host può avere più interfacce e quindi più indirizzi IPv6.

Configurazione statica di un Global Unicast Adderess

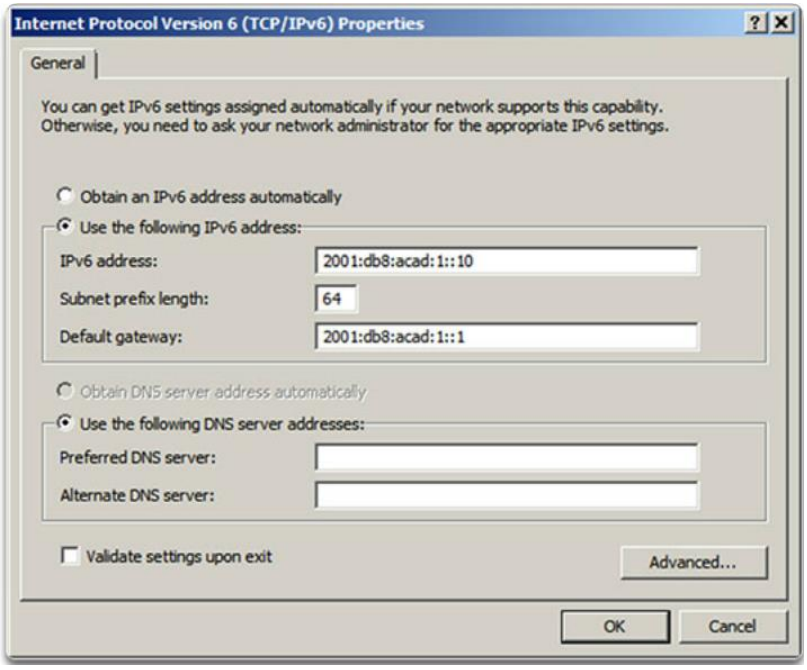
Per la configurazione statica del router avremo:




```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ipv6 address 2001:db8:acad:1::1/64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ipv6 address 2001:db8:acad:2::1/64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ipv6 address 2001:db8:acad:3::1/64
R1(config-if)#clock rate 56000
R1(config-if)#no shutdown
```

La maggior parte di configurazione e di verifica comandi IPv6 in Cisco IOS sono simili alle loro controparti IPv4. In molti casi, l'unica differenza è l'uso di IPv6 al posto di ip nei comandi.

Per la configurazione statica dell' host



Configurare manualmente l'indirizzo IPv6 su un host è simile a quella di un indirizzo IPv4.
L' indirizzo del gateway predefinito configurato per PC1 è il 2001: DB8: ACAD: 1 :: 1. Questo è il GUA dell'interfaccia della g0/0 del router R1 . La configurazione sarà funzionante anche se metteremo il local-link address dell' interfaccia.
Come per IPv4 configurare una rete staticamente è poco scalabile perciò in reti di grandi dimensioni è preferibile l' assegnazione dinamica degli indirizzi IPv6.

Configurazione dinamica di un Global Unicast Adderess
Le modalità per un'interfaccia di ottenere un GUA sono due:

- Stateless Address Autoconfiguration (SLAAC)
- DHCPv6

Nota: In tutti e due i casi il local-link address del router locale sarà l' indirizzo gateway predefinito.

Configurazione dinamica - SLAAC
Stateless Address autoconfiguration (SLAAC) è un metodo che consente a un dispositivo di ottenere il global routing prefix, la sua lunghezza, il gateway ed altre informazioni.
Questo metodo fa uso di messaggi provenienti da un router locale IPv6 detti *Router Advertisement* (***RA***) del protocollo ICMPv6: non viene usato un server DHCPv6.
Il router IPv6 invia un RA in due casi:

- ad intervalli di 200 secondi a tutti i dispositivi abilitati per IPv6 sulla rete.
- in risposta ad un messaggio *Router Solicitation* (**RS**) di un host.

Il messaggio RA è un suggerimento per un host su come ottenere un indirizzo GUA. La decisione finale spetta al sistema operativo dell' host. Il messaggio RA comprende:

- global prefix
- dimensione del global prefix: comunica al dispositivo quale rete appartiene.
- Indirizzo gateway predefinito : indirizzo link-local IPv6 (indirizzo sorgente del messaggio RA).

- indirizzi DNS
- nome di dominio

Abbiamo tre opzioni per i messaggi RA:
Opzione 1: SLAAC
Opzione 2: SLAAC con un server DHCPv6 stateless
Opzione 3: Stateful DHCPv6 (senza SLAAC)

Opzione 1: SLAAC (impostazione di default)
Tutte le informazioni per creare il GUA vengono ricevute dal client con un messaggio RA senza l'uso del DHCP. Quindi le due parti dell'indirizzo vengono create:

- Il global routing prefix e le altre informazioni vengono ricevuto nell' RA;
- Interface ID viene generato come numero casuale o calcolato con un procedimento EUI-64.

Opzione 2: SLAAC e Stateless DHCPv6
Con questa opzione, il messaggio RA suggerisce dispositivi uso:

- global routing prefix
- lunghezza del global routing prefix - comunica al dispositivo quale rete appartiene.
- Indirizzo gateway predefinito - indirizzo link-local IPv6 (indirizzo sorgente del messaggio RA).

Il server server DHCPv6 stateless invia :

- indirizzi DNS
- nome di dominio

Opzione 3: DHCPv6 stateful
Lo stateful DHCPv6 è simile a DHCP di IPv4.
Con questa opzione il messaggio RA *suggerisce* ai dispositivi :

- L'indirizzo del server DHCPv6 stateful
- L' indirizzo link-local del router, cioè l'indirizzo del mittente dell' RA.

Il server DHCPv6 invia al client:

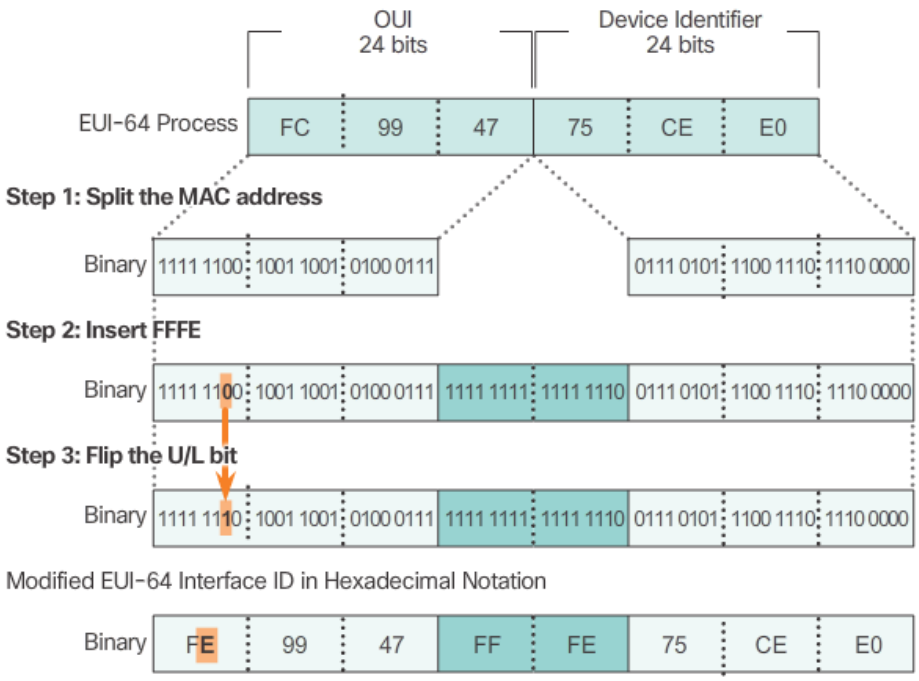
- Global Unicast Adderess
- lunghezza del global prefix
- indirizzi DNS
- nome di dominio
- altre informazioni

Il server DHCPv6 stateful alloca e mantiene una lista dispositivo - indirizzo IPv6 assegnato.
Nota: L'indirizzo del defauly gateway può essere ottenuto *solo dinamicamente dal messaggio RA,* non può essere fornito da un server DHCPv6 stateless o statefull.

Generazione dell' interface ID
Un client attraverso il messaggio RA conosce il global prefix ma non l' interface ID che deve creare utilizzando due metodi:

- generazione tramite il processo EUI-64
- generazione di un numero casuale

Interface ID: generazione tramite il processo EUI-64
Questo processo serve come abbiamo detto per generare i 64 bit dell' interface ID.
IL MAC address dell' interfaccia (48 BIT) viene diviso in due parti e in mezzo vengono inseriti 16 bit per ottenere i 64 bit finali.



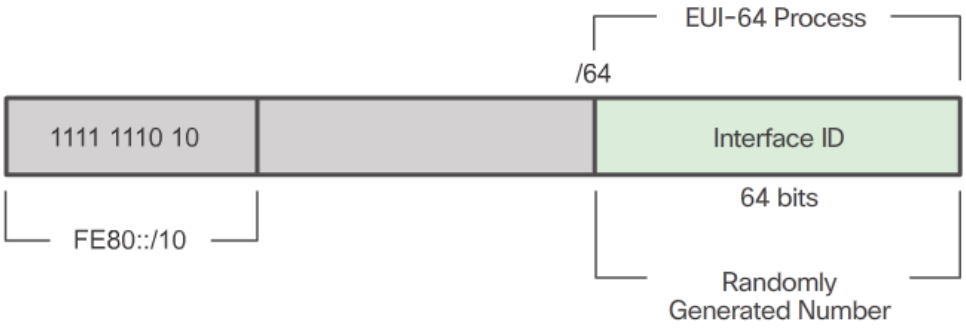
La figura illustra le fasi del processo EUI-64 :

Fase 1: Dividere l'indirizzo MAC tra l'OUI (id del produttore) e numero progressivo.
Fase 2: Inserire il valore **FFFE** esadecimale, che in binario è: 1111 1111 1111 1110.
Fase 3: Convertire le prime due cifre esadecimamali in binario e invertire il valore del bit 7 (detto U / L). In figura lo 0 del bit 7 viene modificato in 1.

Esso consente agli amministratori di rete di tenere facilmente traccia di un indirizzo IPv6 di un client. Tuttavia, vi sono problemi di privacy per l'utente che teme che i pacchetti possono essere ricondotti al computer su cui lavora.

Interface ID : generazione di un numero casuale
In questo caso i 64 bit dell interface ID vengono generati casualmente. Questo sistema viene usato da Windows Vista e successivi, in precedenza era usato l' EUI-64.
Nota: per garantire l'unicità di ogni indirizzo unicast IPv6, il client può utilizzare un processo noto come ***Duplicate Address Detection*** (DAD). Il client invia un messaggio DAD con il proprio indirizzo calcolato se riceve risposta vuol dire che non è unico e deve cambiarlo.

Generazione dinamica di un Link-Local Addresses
Tutti i dispositivi IPv6 devono avere un link-local address IPv6. Anche questo tipo di indirizzo può essere assegnato dinamicamente o staticamente.



La figura mostra la modalità dinamica di generazione di un link-local address:

- FE80 :: / 10 è il prefisso
- l' interface ID viene generata o attraverso il processo EUI-64 o generata in modo casuale.

Generalmente i sistemi operativi usano lo stesso sistema sia per global address che per i link-local address.
I router Cisco ogni volta che un GUA viene assegnato ad una interfaccia creano automaticamente un indirizzo link-local IPv6. Per impostazione predefinita i router Cisco IOS utilizzano EUI-64 per generare l'ID interfaccia per tutti gli indirizzi link-local.
Per i link local è consigliabile un'assegnazione statica vista la difficolta di ricordare un indirizzo da 128 bit.

Indirizzi IPv6 multicast
Un indirizzo multicast viene utilizzato per inviare un singolo pacchetto a una o più destinazioni (gruppo multicast). Possiamo utilizzare il multicast per simulate il boadcast che IPv6 non contempla. Questi indirizzi hanno prefisso:

FF00 :: / 8

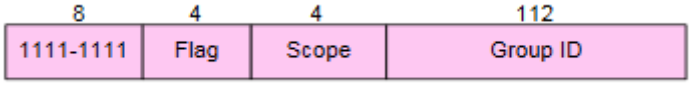
Nota: gli indirizzi multicast possono essere solo indirizzi di destinazione e non mittente.

Ci sono due tipi di indirizzi IPv6 multicast: *Assigned multicast* e *Solicited node multicast*

Assigned Multicast
Un indirizzo multicast assegnato *è un singolo indirizzo utilizzato per raggiungere un gruppo di dispositivi che eseguono un protocollo o un servizio* comune come ad esempio il DNS. Il formato generale è:

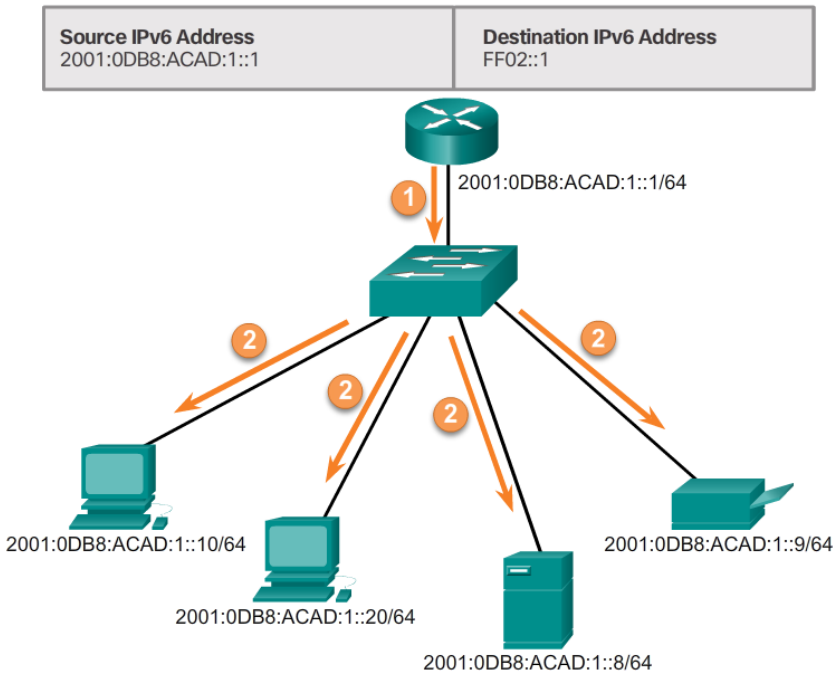
FF<flag><scope>::<group id>

- 1111 1111
- Flag 0000 = indirizzo permanente, 0001 = indirizzo temporaneo
- Scope 1= node 2 =link 5 = site 8 = organization E = global
- Group ID: identifica un gruppo multicast in un dato scope



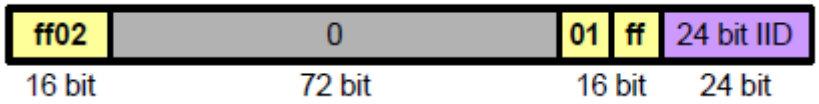
Quindi considerando Group ID = 1 avremo:

FF01 :: 1 *tutti* ***i nodi*** *di un gruppo multicast*. Un pacchetto inviato a questo gruppo viene ricevuto ed elaborato da tutte le interfacce IPv6 sul link o la rete: ha lo stesso effetto di un indirizzo broadcast in IPv4.
FF02 :: 1 *tutti* ***i router*** *di un gruppo multicast*. Tutti i router IPv6 sul link o la rete ricevono il pacchetto inviato a questo gruppo. Un router diventa un membro di questo gruppo quando è abilitato all' IPv6 con il comando **ipv6 unicast-routing**.
FF05::1 partecipano tutte le interfacce sullo stesso *sito (* una o più reti sotto un unica amministrazione *)*
FF0E::1 partecipano tutte le interfacce su *internet*



Sollecited Node multicast

Un indirizzo *sollecited node multicast* è utilizzato per conoscere il MAC address di un'interfaccia IPv6 conoscendone l' indirizzo IPv6 (l'ARP dell'IPv4). La forma di questo tipo di indirizzo multicast è :



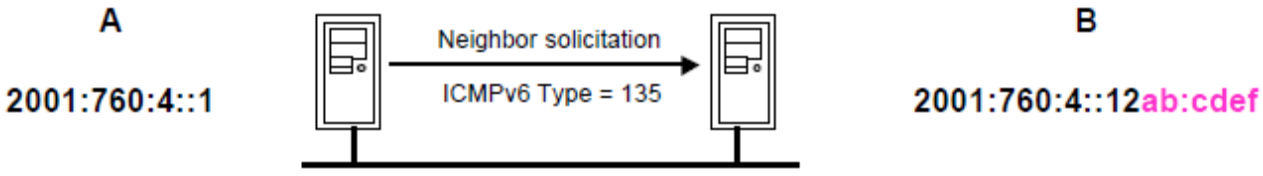
FF02::1:FF *xx:xxxx*

Gli ultimi 24 bit xx:xxxx sono gli ultimi 24 bit dell'indirizzo unicast IPv6 .

Ricordiamo che gli ultimi 24 bit di un indirizzo IPv6 = ultimi 24 bit dei 64 dell'interface ID ottenuta del coaso dell'EUI-64 dal MAC address.

Quindi potremo affermare che ad ogni indirizzo IPv6 unicast o anycast corrisponde un indirizzo Sollecited Node multicast.

Ad esempio l'host A vuole conoscere il MAC Address di B conoscendone l'indirizzo IPv6 (ARP in IPv4)



- A calcola l'indirizzo indirizzo sollecited node multicast di B cioè ff02::1:ff***ab:cdef***
- A invia a B un NS con indirizzo destinatario uguale al sollecited node di B calcolato
- B risponde ad A con NA contenente il MAC address richiesto.

La tabella che segue riassume gli indirizzi di multicast riservati

Address	Scope	Use
FF01::1	Interface-local	All Nodes
FF02::1	Link-local	All Nodes
FF01::2	Interface -local	All Routers
FF02::2	Link-local	All Routers
FF05::2	Site-local	All Routers
FF02::1:FFXX:XXXX	Link-local	Solicited-Node

ICMPv4 e ICMPv6

Sappiamo che IPv4 si affida al protocollo ICMP per i messaggi di errore o informativi. ICMP è disponibile anche per IPv6 ed è denominato ICMPv6 e fornisce funzionalità aggiuntive rispetto all' ICMPv4.

Nota: ci riferiremo con il termine ICMP a entrambi ICMPv4 e ICMPv6.

Sappiamo che i più comuni messaggi per ICMP sono:

- Host confirmation
- Destination or Service Unreachable
- Time exceeded
- Route redirection

Host confirmation

Questo tipo di messaggio è alla base programma di utilità ping. Un messaggio ICMP Echo (tipo 8) viene inviato ad un host che se disponibile risponde con un messaggio echo reply (tipo 0).

Destinazione o servizio non raggiungibile

Quando un host o router riceve un pacchetto che non può trasportare, è possibile utilizzare un messaggio ICMP per notificare al mittente che la destinazione o il servizio non è raggiungibile. Nel messaggio il campo codice indica il motivo per cui il pacchetto non può essere consegnato.

Alcuni dei codici di destinazione irraggiungibile per ICMPv4 sono:

0 - Rete irraggiungibile
1 - Host irraggiungibile
2 - Protocollo irraggiungibile
3 - Porta irraggiungibile

Time Exceeded (Tempo scaduto)

TTL (IPv4) o hop limit (IPv6) e' uguale a 0 dopo il decremento unitario fatto dal router. In questa situazione il il pacchetto viene scartato e al mittente viene inviato un messaggio di tempo scaduto.

Funzionalita aggiuntive dell' ICMPv6

I messaggi informativi e di errore che si trovano in ICMPv6 sono molto simili ai messaggi di controllo e di errore attuate da ICMPv4. Tuttavia, ICMPv6 ha nuove caratteristiche e funzionalità migliorate che non si trovano in ICMPv4. Di seguito una tabella con i messaggi ICMPv6:

1	Destination Unreachable	133	Router Solicitation
2	Packet Too Big	134	Router Advertisement
3	Time Exceeded	135	Neighbor Solicitation
4	Parameter Problem	136	Neighbor Advertisement
		137	Redirect Message
128	Echo Request	138	Router Renumbering
129	Echo Reply	139	ICMP Node Information Query
130	Multicast Listener Query	140	ICMP Node Information Response
131	Multicast Listener Report	141	Inverse Neighbor Disc. Solicitation
132	Multicast Listener Done	142	Inverse Neighbor Disc. Advertisement

I messaggi ICMPv6 come nel caso dell'ICMP, sono incapsulati in IPv6. ICMPv6 include quattro nuovi messaggi come parte del Neighbor Discovery Protocol (NDP).

Messaggistica tra un router IPv6 e un host IPv6:
- Router Solicitation (RS) codice 133
- Router Advertisement (RA) codice 134

Messaggistica tra dispositivi IPv6:
- Neighbor Solicitation (NS) codice 135
- Neighbor Advertisement (NA) codice 136

NS e NA vengono utilizzati per l' Address resolution e il Duplicate Address Detection (DAD).

Address Resolution
A conosce l' IPv6 di B e ne vuole conoscere il MAC Address. A calcola l'indirizzo indirizzo sollecited di B e invia un NS all'host all'host B e l'host risponde con un NA che contiene il proprio MAC address.
Duplicate Address Detection
Quando ad un interfaccia di un host *viene assegnato un GUA o un link-local address*, per garantirne l' univocità si raccomanda l'uso del DAD. Il dispositivo invierà un messaggio di NS con l' proprio indirizzo IPv6 (da controllare) come indirizzo di destinazione. Se un altro dispositivo in rete ha lo stesso indirizzo, risponderà con un messaggio di NA e quindi notificherà la duplicazione dell'indirizzo.

Instradamento
Come per IPv4 anche IPv6 prevede che l'instradamento sia realizzato sulla base di lunghezza variabile sino ad un massimo 64 bit, lasciando i restanti 64 meno significativi per l'indirizzamento degli host.

Importante
Ogni host IPv6 deve riconoscere come propri i seguenti indirizzi :

- Un indirizzo *link-local* per ogni interfaccia
- Un indirizzo *unicast / anycast* assegnati manualmente o automaticamente
- L' indirizzo ::1 / 128 di *loopback*
- Gli indirizzi di *multicast* di tutti i gruppi a cui appartiene l'host
- Gli indirizzi *Solicited node multicast* per ogni indirizzo unicast / anycast assegnato

Il protocollo TCP

E’ un protocollo orientato alla connessione ed ha il compito di trasmettere in modo affidabile i dati tra due nodi della rete.
Ricordiamo che in un servizio con modalita’ connessa lo scambio di dati tramite pacchetti avviene attraverso tre fasi principali:

• creazione della connessione
• Scambio di dati
• Chiusura della connessione

Destination Port	16 bit	porta host destinatario
Source Port	16 bit	porta host sorgente
Sequence Number	32 bit	Definisce l'ordine in cui i segmenti devono essere riassemblati. E' utilizzato anche nella fase di connessione
Ack Number	16 bit	Contiene il prossimo numero di sequenza che l'host destinatario si aspetta di ricevere dall'host mittente. Esprime il numero di segmenti ricevuti correttamente fino a quel momento
Data Offset	4 bit	Indica dove inziano i dati (l' unità di misura è 32 bit)
Reserved	6 bit	Riservato per utilizzi futuri
URG	1 bit	se settato indica che nel flusso sono presenti dati urgenti alla posizione (offset) indicata dal campo Urgent pointer;
ACK	1 bit	se settato indica che il campo Acknowledgement Number è significativo è deve essere letto
PSH	1 bit	se settato significa che il pacchetto deve essere inviato immediatamente, senza aspettare il riempimento del buffer di TX. Il ricevente a sua volta invia immediatamente il pacchetto al livello applicazione.
RST	1 bit	se settato la connessione deve essere reinizializzata, solitamente a seguito di problemi

SYN	1 bit	viene utilizzato per stabilire una sessione, indica al destinatario di leggere il campo Sequence number e sincronizzare il proprio con esso
FIN	1 bit	indica che l'host mittente non ha più dati da spedire, e vuole terminare la connessione
Windows Size	16 bit	Contiene la dimensione del buffer di dati del mittente
Checksum	16 bit	Stabilisce la correttezza delle informazioni (Intestazione+Dati)
Urgent Pointer	16 bit	Indica quale porzione dati è urgente
Options		campo di dimensione variabile, contiene le opzioni per la comunicazione
Padding		campo di dimensioni variabili, è utilizzato per far raggiungere all'area d'intestazione una dimensione di 32 bit o un suo multiplo
Data		dati trasportati dal protocollo

Il numero di porta del destinatario (numero da 0 a 65535) rappresenta il servizio che si vuole dall' host di destinazione. In particolare i primi 1024 sono conosciuti come Well Known Port e sono riservati a protocolli noti. Di seguito una tabella con i principali numeri di porta ed i relativi protocolli associati.

Port Number	***Associated Protocol***	***Full Name***
21	FTP	File Transfer Protocol
22	SSH	Secure Shell
23	Telnet	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name System
80	HTTP	Hypertext Transfer Protocol
88	Kerberos	Kerberos
110	POP3	Post Office Protocol Version 3
119	NNTP	Network News Transfer Protocol
137–139	NetBIOS	NetBIOS Name, Datagram, and Session Services, respectively
143	IMAP	Internet Access Message Protocol
161	SNMP	Simple Network Management Protocol
389	LDAP	Lightweight Directory Access Protocol
443	HTTPS	Hypertext Transfer Protocol Secure (uses TLS or SSL)
445	SMB	Server Message Block
1701	L2TP	Layer 2 Tunneling Protocol
1723	PPTP	Point-to-Point Tunneling Protocol
3389	RDP	Remote Desktop Protocol (Microsoft Terminal Server)

I FLAG TCP
Durante una sessione TCP è di fondamentale importanza lo stato dei 6 flag che possono assumere combinazioni differenti:

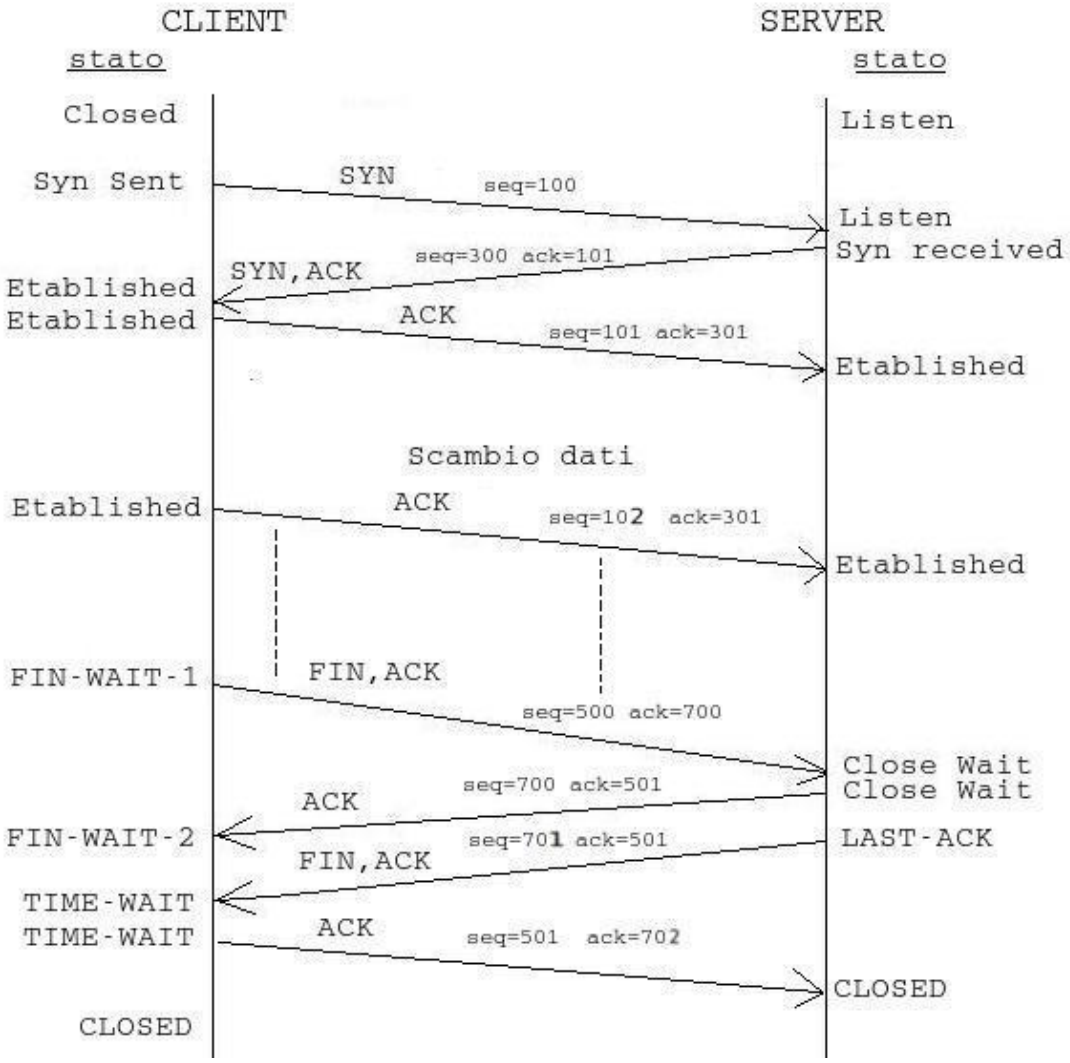
SYN	è settato nel primo pacchetto di un host che intende stabilire la connessione
SYN | ACK	è la risposta di un host contattato che accetta la connessione;
ACK	a connessione stabilita, ogni pacchetto è confermato tramite i flag ACK settato

FIN	è inviato da un host che intende chiudere una sessione;
FIN | ACK	è la risposta di un host che conferma la chiusura di una connessione;
RST	viene inviato da un host che riceve un pacchetto inatteso e che quindi non accetta la connessione;

STATI DI UNA SESSIONE TCP
Una sessione TCP attraversa diversi stati in seguito al verificarsi di determinati eventi:

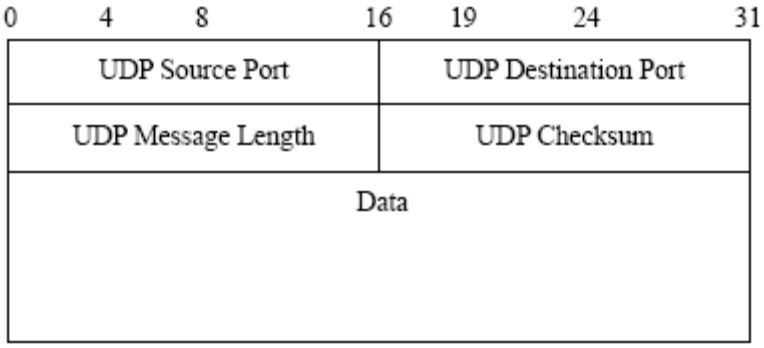
LISTEN	host in attesa di connessione;
SYN-SENT	host che richiede la connessione ed è in attesa di risposta;
SYN-RECEIVED	host in attesa di conferma per la richiesta di connessione dopo aver ricevuto ed inviato una richiesta di conferma;
ESTABLISHED	host con una connessione aperta, durante la quale si cambiano dati.
FIN-WAIT1	host in attesa di una richiesta di termine della sessione o di conferma di richiesta di termine della connessione;
FIN-WAIT2	host in attesa di una richiesta di termine della sessione da parte di un host remoto;
CLOSE-WAIT	host in attesa di terminare la sessione;
CLOSING	host in attesa della conferma della richiesta di termine di connessione;
LAST-ACK	host in attesa della conferma dellle richiesta di termine della connessione già inviata all'host remoto;
TIME-WAIT	host in attesa (per un determinato lasso di tempo) per garantire che l'host remoto abbia ricevuto la conferma della richiesta di termine della connessione;
CLOSED	non esiste connessione tra host;

SESSIONE TCP
Il presupposto per instaurare una connessione è l'esistenza di un server con un socket attivo in stato di listen (attesa).



UDP - User Datagram Protocol

E' un protocollo non orientato alla connessione utilizzato in sostituzione del TCP, quando l'affidabilità non è lo scopo principale
(l'affidabilità viene garantita dai protocolli applicativi che ne fanno uso).
I vantaggi nell'utilizzo di UDP
• velocità
• la minore congestione di rete rispetto al TCP (non ci sono pacchetti di conferma)
• la possibilità di trasmettere in multicast (invio di un pacchetto ad un gruppo di host) e broadcast (invio di un pacchetto a tutti
gli host di un segmento di rete).
TRASMISSIONE UDP
La trasmissione di un pacchetto UDP avviene incapsulandolo all'interno di un pacchetto IP. Giunto a destinazione, il pacchetto
viene inviato alla porta di destinazione indicata nell'intestazione UDP. Qualora la porta non fosse disponibile, viene inviato un
paccheto ICMP all'host mittente con messaggio di port unreachable (porta irraggiungibile).
Formato



Porta Sorgente	se presente rappresenta la porta del mittente a cui indirizzare la risposta.
Porta di Destinazione	rappresenta il servizio richiesto all'host di destinazione.
Lunghezza	è la lunghezza totale (header + dati) del datagramma utente (il valore minimo è otto).
Checksum	è il risultato a 16-bit della somma del blocco di ottetti formati dall'header IP, dall'header UDP e dai dati. Infine, se necessario, vengono inseriti ottetti di riempimento fino ad arrivare ad un multiplo di 2 ottetti.
Dati	Contiene i dati del segmento

Esempio di checksum
Il mittente
- calcola la somma di tutte le parole (16 bit) contenute nel messaggio
- calcola il complemento ad 1 della somma
- invia il risultato al destinatario

01100110 01100110 +
01010101 01010101 +
00001111 00001111 =

11001010 11001010
Complemento ad 1 = checksum 0011010100110101

Il destinatario
calcola la somma di tutte le parole contenute nel segmento + il checksum
11001010 11001010 +
00110101 00110101 =

11111111 11111111 se il risultato è composto da tutti 1 non vi sono stati errori nella trasmissione.

UDP e TCP

In generale, le principali differenze tra i protocolli UDP e TCP per quanto riguarda la consegna dei dati sono simili alle differenze che intercorrono tra una telefonata e una cartolina. TCP funziona come una telefonata, ovvero verifica che la destinazione sia disponibile e pronta a comunicare. UDP può essere paragonato a una cartolina: i messaggi sono brevi e il recapito è probabile ma non sempre

garantito. Nella tabella seguente vengono indicate le differenze nella gestione delle comunicazioni TCP/IP a seconda che venga utilizzato UDP o TCP per il trasporto dei dati.

UDP	**TCP**
Servizio senza connessione; non viene stabilita alcuna sessione tra gli host.	Servizio orientato alla connessione; viene stabilita una sessione tra gli host.
UDP non garantisce la consegna dei pacchetti ne la loro consegna nella giusta sequenza .	TCP garantisce la consegna ed il recapito dei pacchetti nella giusta sequenza .
UDP non garantisce l'affidabilità, sono i programmi che utilizzano UDP che la devono fornire.	Ai programmi che utilizzano TCP viene garantito un trasporto affidabile dei dati.
UDP è rapido e supporta la comunicazione multicast, broadcast oltre che unicast.	TCP è più lento e supporta solo la comunicazione unicast.

L’ UDP viene utilizzato da protocolli come TFTP (Trivial File Transfer Protocol), SNMP (Simple Network Managment Protocol), DNS (Domain Name Server), per l’invio di stream audio/video, ed è ampiamente usato nelle applicazioni di gioco.

Protocolli del livello applicativo

Protocollo HTTP / HTTPS

HTTP (Hyper Text Transfert Protocol) ed è uno dei più importanti protocolli di livello applicativo. Tutte le volte che un utente richiede una qualsiasi pagina Web utilizza il protocollo HTTP sia per fare la richiesta al server che ospita la pagina sia per ricevere i dati provenienti in risposta dal server.
HTTP è presente nello stack TCP/IP al livello applicativo sia del client che del server. Il client richiede una risorsa, il server elabora la richiesta e fornisce a video la risposta alla richiesta. Il browser vede una pagina web come un insieme di oggetti legati tra di loro attraverso collegamenti ipertestuali. La pagina web è formata da un corpo HTML più altre risorse che potrebbero essere script, immagini, applet Java e così via.
Tutta le risorse sono raggiungibili tramite un percorso detto **URL** (Uniform Resurce Locator) che fa riferimento ad uno solo e specifico oggetto o *risorsa* (testo, immagine fissa e in movimento, suono ecc..ecc..). La struttura delle URL è descritta nell' esempio seguente :

http:// www.miosito.it / informazioni / curriculum.html

- *www.miosito.it*: identifica il server sul quale viene ospitata la risorsa che cerchiamo.
- */ informazioni / curriculum.html* : identifica il percorso da seguire (all'interno del server) per raggiungere la risorsa cercata. Nel percorso possono essere passati dei parametri (QueryString) per effettuare interrogazioni a script lato server (programmi veri e prori).

Questo protocollo viene definito ***stateless*** perché la connessione dura solo il tempo per trasmettere e in seguito viene chiusa. Quindi il server non tiene nota dei dati precedenti, ogni richiesta del client viene trattata come se provenisse da un client diverso.
Il protocollo HTTP si affida al protocollo TCP del livello di trasporto per gestire il trasferimento di dati dal server al client. La scelta di utilizzare il TCP è dovuta al fatto che il TCP garantisce un servizio di trasferimento affidabile dei dati, cosa che non garantisce l'altro protocollo di trasporto che è l'UDP. La maggiore affidabilità dell'uso del TCP ha un costo perchè comporta un tempo di attesa maggiore per ricevere la risorsa richiesta in quanto il TCP, prima di trasmettere i dati, deve instaurare una connessione.
Per l'HTTP è fondamentale il trasferimento affidabile dei dati in quanto se non tutto il corpo HTML di una pagina web venisse trasferito al browser, a causa di qualche errore di trasmissione, la pagina richiesta sarebbe impossibile da visualizzare.
L'HTTP scambia i dati con il TCP attraverso le *socket*. Ogni applicazione su un determinato host avrà una specifica socket di interfacciamento verso il livello di trasporto, quindi se per esempio un utente richiede una pagina web e contemporaneamente sta inviando una e-mail saranno presenti due socket una che gestiscce lo scambio dati tra HTTP e il livello di trasporto e l'altra tra i protocolli di posta elettronica e il livello di trasporto.
Un esempio di socket potrebbe essere ***88.33.45.2, 3456, 121.34.34.34, 80***
Indirizzo IP e porta del mittente, Indirizzo IP e porta del destinatario.

La comunicazione client/server

L'HTTP è un protocollo di *tipo request/response*. Per prima cosa, quindi, deve essere fatta una richiesta per una risorsa e poi avviene la risposta con il trasferimento dei dati.
Ad esempio, se un utente richiede una pagina web che contiene testo HTML con riferimento a tre immagini GIF:

1. Il client richiede di aprire una connessione di tipo TCP verso il server che ospita la risorsa cercata sulla porta 80 che è la porta assegnata di dafault per l'HTTP.
2. Il server riceve la richiesta di connessione, se le condizioni lo permettono, accetta la richiesta di connessione e invia una notifica al client.

3. Il client allora formula una richiesta HTTP per la risorsa desiderata e passa la richiesta, tramite la socket, al protocollo TCP che si preoccuperà di gestire la connessione con il server.
4. Il server riceve la richiesta HTTP del client e se non ci sono errori di sintassi nella richiesta invia al client il testo HTML della pagina richiesta. A questo punto il Server notifica al TCP di chiudere la connessione con il client non appena arrivi la notifica di corretta ricezione.
5. Il client riceve correttamente i dati contenenti il testo HTML, lo analizza e nell'analisi trova il riferimento alle tre immagini GIF.
6. Per ogni immagini GIF vengono ripetuti i passi dall'1-5.
7. Il client, una volta scaricato il corpo HTML della pagina e tutti gli oggetti refereziati dalla stessa, assembla il tutto mostrando a video la pagina web richiesta precedentemente dall'utente.

Come abbiamo visto il procedimento per acquisire una pagina web è particolarmente articolato e, a prima vista, potrebbe sembrare anche molto lungo a seconda della complessità della pagina. Questo aspetto di complessità del procedimento per acquisire una risorsa, generalmente, non viene percepita dall'utente in quanto le attuali alte velocità dei collegamenti alla Rete consentono un trasferimento dati elevato e, quindi, il procedimento di richiesta di una risorsa e la visualizzazione della stessa a video è quasi immediato.
Inoltre i moderni browser possono gestire più connessione TCP in parallelo (si pensi a quando si richiedano due pagine web contemporaneamente) e quindi, nel caso d'esempio visto sopra, il browser quando analizzerà il listato HTML e trovarà i riferimenti alle tre immagini GIF aprirà immediatamente tre connessioni, in parallelo, per il download delle immagini richieste. Il fatto di lavorare in parallelo accorcia notevolmente il tempo di acquisizione totale della pagina.

Metodi
I metodi principali del protocollo http sono:

GET	richiede l'invio della risorsa che corrisponde ad un determinato URL
POST	aggiunge una risorsa all'insieme di risorse che corrisponde ad un determinato "URL". I dati vengono inviati all'interno del body della request.
PUT	aggiorna una risorsa che corrisponde ad un url. Anche in questo caso i dati sono nel body della request.
DELETE	cancella la risorsa all'URL specificato

Alcuni dei metodi http si definiscono sicuri in quanto non provocano modifiche sul server ad esempio GET e HEAD (una get che restituisce solo l'header). Altri metodi si definiscono idempotenti in quanto è possibile richiamarli più volte senza che ci sia una differenza nello stato del server dopo la prima volta. Ad esempio se invio due DELETE il primo eliminerà una determinata risorsa ed il secondo non farà nulla in quanto non troverà la risorsa da cancellare. Tutti i metodi http sono idempotenti tranne POST che è anche insicuro.
Ad esempio

GET /dati/saluti.html http/1.0

Il server risponde con l'invio della pagina html

http/1.0 200 OK
Content-Type: text/html
Content-Length: 1354
<html>
<body>
……..
……..

</body>
</html>

Codici di risposta
Il protocollo http prevede un ampia gamma di codici di risposta. Questi vengono usati per controllare il flusso tra client e server. Si dividono in gruppi in base alla prima cifra.
100 Informazioni
200 Successo
300 Redirezione
400 errore del client
500 errore del server
I più noti sono:
200 OK - Successo
201 Created - spesso inviato in risposta ad una post seguito dall'header "location"
301 Moved Permanently - la risorsa è stata spostata (segue il nuovo indirizzo)
304 Not Modified - la risposta si trova nella cache del browser (vedi più avanti)
404 Not Found - la risorsa cercata non esiste
401 Unauthorized - utilizzato per gestire l'autenticazione
500 Internal Server Error - errore generico del server

Il protocollo HTTPS
Quando si utilizza il protocollo https (Hyper Text Transfert Protocol Secure), la trasmissione dei dati avviene in modalità criptata e quindi non intercettabile da terzi. Di solito l'https viene utilizzato dove la trasmissione dei dati sensibili deve essere garantita come nei casi di banche o negozi online. Il sintesi viene creato un canale di comunicazione criptato tra il client e il server attraverso uno scambio di certificati; una volta stabilito questo canale al suo interno viene utilizzato il protocollo HTTP per la comunicazione. Questo tipo di comunicazione garantisce che solamente il client e il server siano in grado di conoscere il contenuto della comunicazione.

Protocollo DHCP
Il Dynamic Host Configuration Protocol (**DHCP**) è un protocollo di rete di livello applicativo che permette ad un host di ricevere dinamicamente a ogni richiesta di accesso a una rete basata sui Protocolli TCP/IP la configurazione necessaria per stabilire una connessione.
La configurazione inviata all'host prevede fra gli altri l'invio :
- Indirizzo IP
- Maschera di sottorete
- Default gateway
- Indirizzi dei server DNS
- Nome di dominio DNS di default
- Indirizzo di un server tftp e nome di un file da caricare per calcolatori che caricano dalla rete l'immagine del sistema operativo (si veda PXE Preboot Execution Environment).

I componenti del protocollo sono :
- Il **Client DHCP** è l'host che richiede l'indirizzo IP per la rete a cui è collegato, ma con quasto nome viene anche indicato il programma che si occupa di richiedere l'indirizzo IP e configurarlo.
- Il **Server DHCP** è il calcolatore che assegna gli indirizzi IP, ma con esso viene anche indicato il processo che svolge questa funzione.

Il DHCP relay è il calcolatore che si occupa di inoltrare le richieste DHCP ad un server, qualora questo non sia sulla stessa sottorete. DHCP utilizza il protocollo UDP, le porte registrate sono la **67** per il server e la **68** per il client.

Una sessione DHCP è un processo per ottenere un contratto di locazione costituito da 4 fasi conosciute come **D.O.R.A.**

DISCOVER (DHCPDISCOVER)
Il client inizia il processo con questo tipo di messaggio che contiene il proprio indirizzo MAC per scoprire i server DHCPv4 disponibili. Poiché il client all'avvio non ha configurazione IPv4, utilizza il *broadcast* Layer 2 per comunicare con il/i server.

OFFER (DHCPOFFER)
Il/i server DHCPv4 riceve/ono il messaggio DHCPDISCOVER,

1 - riservano un indirizzo IPv4 disponibile da affittare al client,
2 - creano una voce ARP con il MAC del client richiedente e l'indirizzo IPv4 riservatogli
3 - inviano un messaggio t DHCPOFFER in *unicast* visto che conoscono il MAC address del client.

REQUEST (DHCPREQUEST)
Quando il client riceve il DHCPOFFER dal server, gli invia un messaggio DHCPREQUEST. Questo messaggio viene utilizzato sia per l'iniziale richiesta di lease che per il renew. Nel primo caso serve come un avviso di accettazione vincolante al server selezionato per i parametri che ha offerto e un implicito declino a qualsiasi altro server che possono aver fornito al cliente una offerta vincolante. Questo tipo di messaggio viene inviato in *broadcast* per informare tutti i server DHCPv4 circa l'offerta accettata.

ACK (DHCPACK)
Alla ricezione del messaggio DHCPREQUEST,

1- il server verifica le informazioni di locazione con un ping all'indirizzo per assicurare che non sia già in uso.
2- crea una nuova voce ARP per la locazione client,
3- risponde in *unicast* con un messaggio DHCPACK.

Quando il client riceve il messaggio DHCPACK, registra le informazioni di configurazione ed esegue una ricerca ARP per l'indirizzo assegnato. Se non c'è risposta al ARP, il client sa che l'indirizzo IPv4 è valido e inizia ad usarlo come proprio.

Rinnovo contratto di affitto

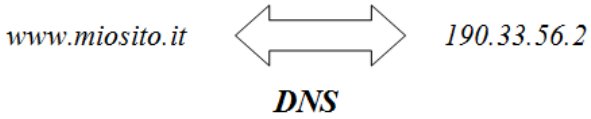
REQUEST (DHCPREQUEST)
quando la locazione è scaduta, il client invia un messaggio DHCPREQUEST direttamente al server DHCPv4 che inizialmente offerto l'indirizzo IPv4. Se un DHCPACK non viene ricevuto entro un determinato periodo di tempo, il client trasmette un altro DHCPREQUEST in modo che uno degli altri server DHCPv4 possibile estendere il contratto di locazione.

ACK (DHCPACK)
Alla ricezione del messaggio DHCPREQUEST, il server verifica le informazioni di locazione restituendo un DHCPACK.

A questo punto, il client è autorizzato a usare l'indirizzo ricevuto per un tempo limitato, detto tempo di lease. Prima della scadenza, dovrà tentare di rinnovarlo inviando un nuovo pacchetto DHCPREQUEST al server, che gli risponderà con un DHCPACK se vuole prolungare l'assegnazione dell'indirizzo. Questi sono normali pacchetti IP unicast scambiati tra due calcolatori che hanno indirizzi validi. Se il client non riesce a rinnovare l'indirizzo, tornerà allo stato iniziale cercando di farsene attribuire un altro.
Nei sistemi Windows il comando ***ipconfig*** oltre a darci le informazioni complete relative al sistema e alla configurazione delle interfacce (/ALL), attraverso le opzioni /RELEASE e /RENEW *rilascia e rinnova* la richiesta ad un server DHCP della configurazione IP.
I sistemi Unix e Unix-like come LINUX offrono il comando ***ifconfig*** che oltre che a configurare le interfacce, permette di conoscere la configurazione IP del proprio computer.

Protocollo DNS

Il Domain Name System è' un protocollo di livello applicativo, con cui l'utente generalmente non interagisce direttamente, ma viene utilizzato da altri protocolli come l'http, ftp, smtp per "*risolvere*" i nomi di dominio in indirizzi di IP e viceversa.



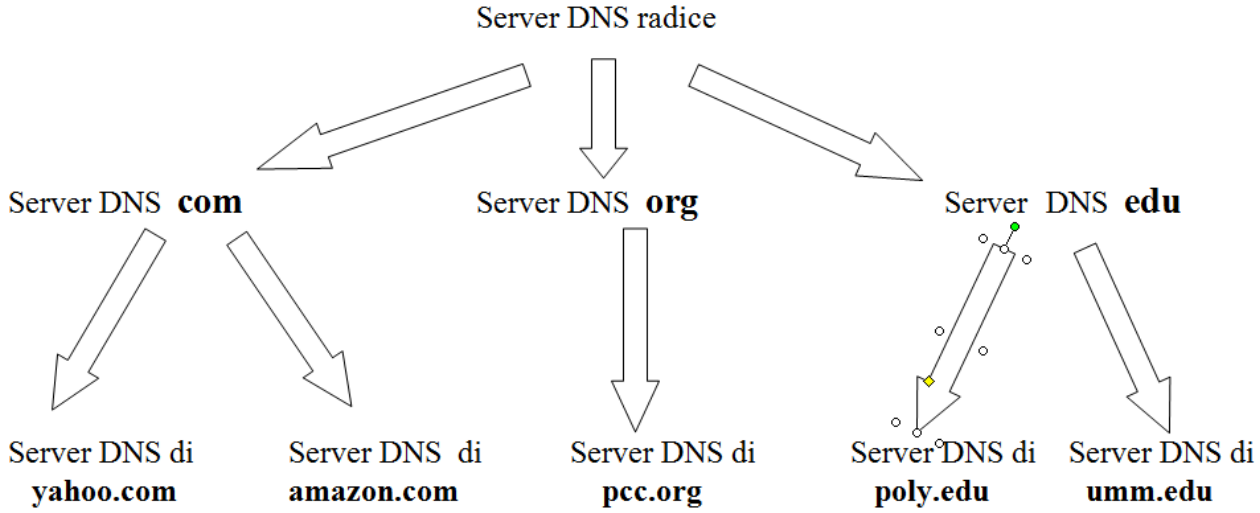
Il DNS per il trasporto utilizza l'UDP e indirizza la porta 53. un possibile interazione con l'http è schematizzata nell'esempio seguente:

1. L'host esegue il lato client dell'applicazione DNS
2. Il browser estrae il nome dell' host, www.miosito.com dall'URL e lo passa al lato client dell'applicazione DNS
3. Il client DNS invia una query contenente l' hostname a un server DNS
4. Il client DNS riceve l'indirizzo IP corrispondente all'hostname
5. Ottenuto l'indirizzo IP dal DNS, il browser può dare inizio alla connessione TCP verso il server HTTP localizzato a quell'indirizzo IP

Il mapping (corrispondenza) per tutti gli host di Internet viene distribuito su più server in modo gerarchico. I server vengono classificati in 3 gruppi:

- Root
- Top level domain (TLD)
- Authoritative

Ci possono essere anche dei server dns locali che interagiscono direttamente con le applicazioni.



Se volessimo conoscere l'indirizzo di IP di Yahoo.com :

- Il client interroga il server root per trovare il server **com**
- Il client interroga il server com per trovare il server **yahoo.com**
- Il client interroga il server **yahoo.com** per ottenere l'indirizzo di IP di yahoo.com

Con il comando:

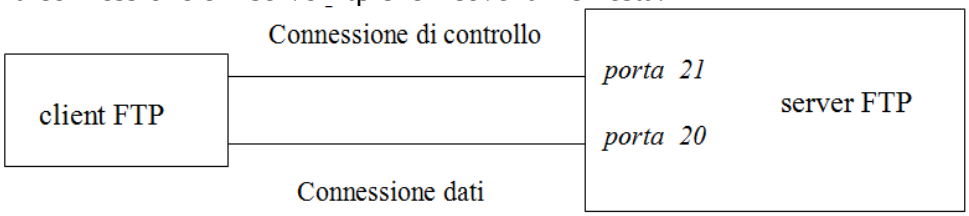
nslookup nomeDominio | indirizzo IP

potremo interrogare direttamente la gerarchia di server per ricercare un nome di domioni o un'indirizzo di IP

```
C:\Windows\system32>nslookup yahoo.com
Server:  resolver1.opendns.com
Address:  208.67.222.222

Risposta da un server non autorevole:
Nome:    yahoo.com
Addresses:  206.190.36.45
          98.138.253.109
          98.139.183.24
```

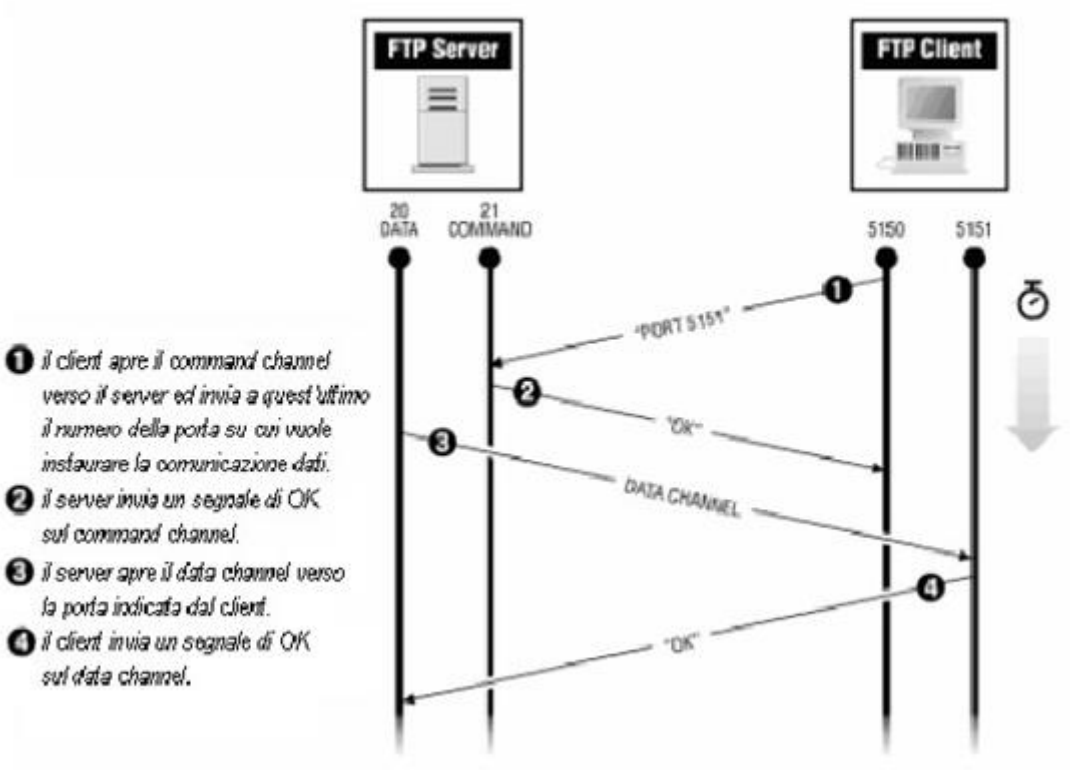
Protocollo FTP

Il protocollo FTP (file transfer protocol) è utilizzato per trasferire file fra due host, il client FTP che richiede la connessione e il server ftp che riceve la richiesta.

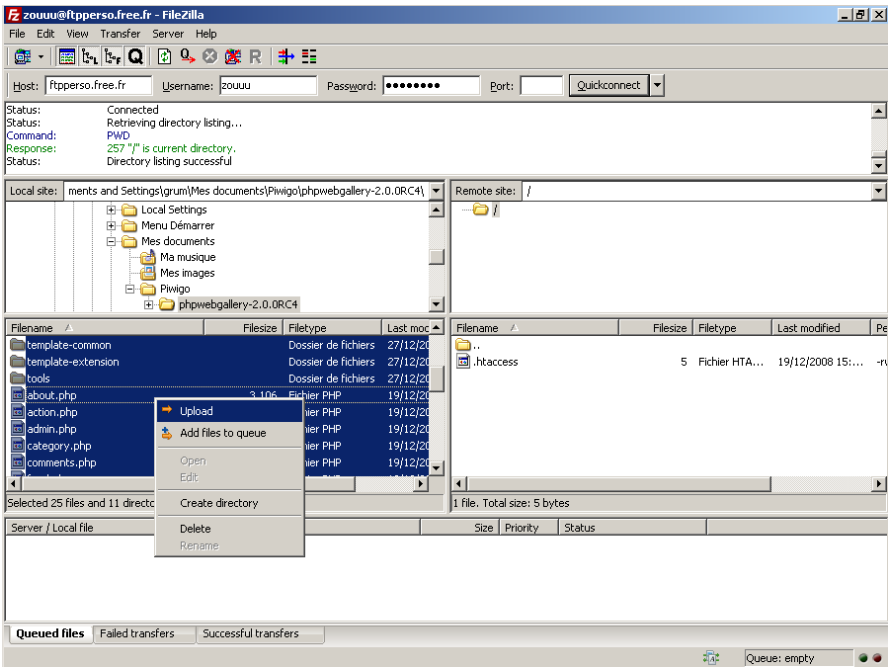


L'FTP una due tipi di connessione : la prima è detta *connessione di controllo* serve al client per inviare comandi e ricevere risposte, la seconda *connessione dati* , viene stabilita successivamente quando inizia il trasferimento dei file.

Quando un utente avvia una sessione FTP con un server remoto, il lato client dell'FTP come prima cosa instaura una connessione TCP di controllo con il lato server (host remoto) sulla porta 21 del server. Il lato client dell'FTP invia l'identificazione dell'utente (nome utente) e la password su questa connessione di controllo, nonché gli eventuali comandi per cambiare la directory remota. Quando il lato server riceve sulla connessione di controllo un comando per il trasferimento di un file (sia verso, sia dall'host remoto) il lato server inizia una connessione dati verso il lato client, utilizzando la porta 20, e poi la chiude. Se, durante la stessa sessione, l'utente vuole trasferire un altro file, l'FTP apre un'altra connessione dati. In generale, quindi, per tutta la durata della sessione la connessione di controllo rimane sempre aperta, mentre viene stabilita una nuova connessione dati per ciascun file trasferito all'interno della sessione (cioè, la connessione dati non è persistente).



Esistono dei software GUI , come ***filezilla client***, che consentono di utilizzare questo protocollo in modo semplice ed intuitivo.



Se vogliamo utilizzare il client testuale ***ftp*** dobbiamo conoscere i comandi:

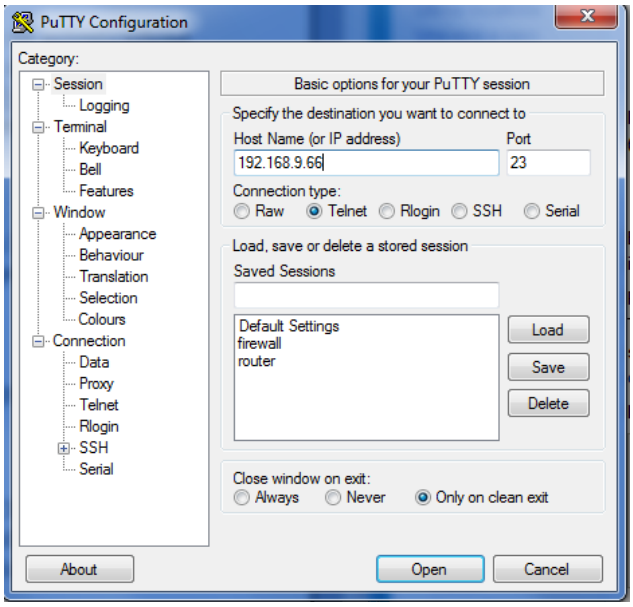
- USER, serve ad inviare il nome utente per l'autenticazione;
- PASS, utilizzato per completare la fase di autenticazione inviando la password;
- PASV, necessario per impostare la connessione passiva acquisendo quindi la porta da utilizzare per lo scambio di files e liste di files;
- PWD, per acquisire la directory corrente;

- CWD, per spostarsi tra le directory;
- TYPE, permette di impostare il tipo di codifica usato per lo scambio dei files;
- RNFR e RNTO, indicano rispettivamente Rinomina Da e Rinomina A permettendo di rinominare un file o una directory;
- MKD, crea una cartella;
- RMD, elimina una cartella esistente;
- DELE, permette l'eliminazione di files;
- HELP, restituisce l'help dei comandi;
- SYST, ritorna il sistema operativo utilizzato dal server FTP;
- NOOP, comando per bloccare un'eventuale chisura di connessione dovuta al timeout di trasferimento dei file;
- LIST, per elencare le directory ed i file presenti nella cartella corrente sul canale dati;
- STAT, che sostituisce LIST permettendo di acquisire l'elenco delle cartelle sul canale dei comandi;
- STOR, che indica al server FTP di acquisire il file che verrà sul canale dati;
- RETR, per scaricare il file indicato tramite il canale dati;
- QUIT, esegue la disconnessione.

Ad esempio
ftp serverFtp
ftp>USER pippo
ftp>PASS ******
ftp>GET testo.txt
ftp>DELE paperino.doc
…..
……
ftp>quit

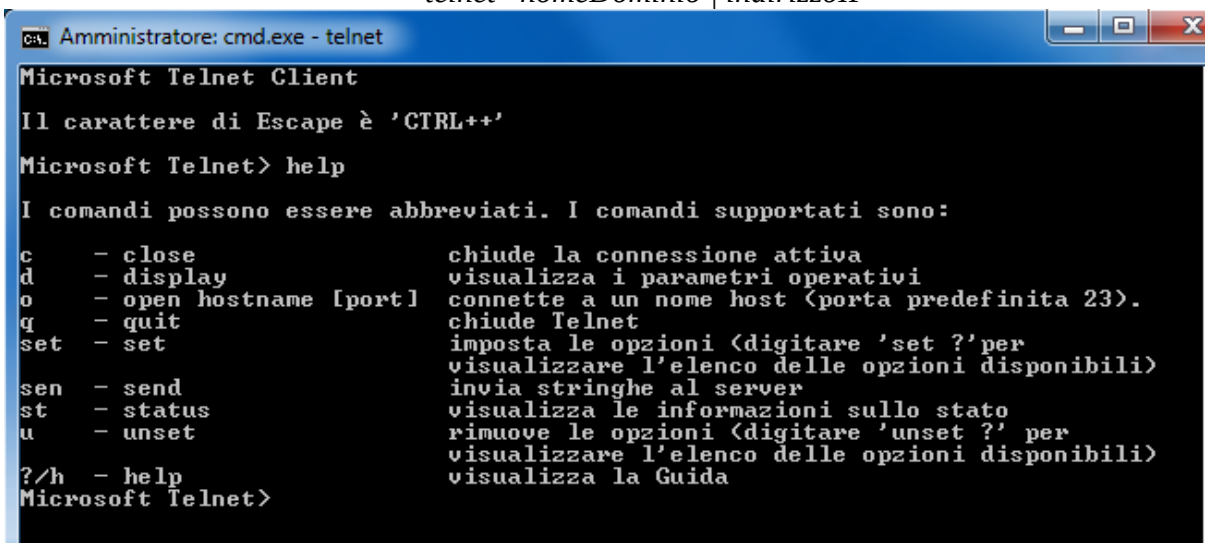
Protocollo TFTP
E' una versione semplificata (e poco sicura) del protocollo FTP che non prevede ne autenticazione ne verifica sulla correttezza del traferimento. Il suo utilizzo prevalente è in fase di configurazione di dispositivi programmabili come switch e router o per avviare stazioni diskless (prive di memoria di massa e quindi impossibilitate a memorizzare un sistema operativo che in questo caso risiederà su un server e da esso verrà caricato attraverso TFTP).

Protocollo Telnet
Telnet è un protocollo client-server basato su TCP (porta 23) che consente da un host locale di connettersi, dopo autenticazione, ad un server remoto. Esistono client GUI come ***putty*** che consentono l'uso del protocollo in maniera semplice ed intuitiva.



Ma possiamo anche usate il client testuale ***telnet***. La cui sintassi è:

telnet nomeDominio | indirizzoIP



Il protocollo telnet non è sicuro perchè i dati vengono trasmessi "in chiaro", e dovrebbe esser generalmente evitato.

Protocollo SSH

L' insicurezza del protocollo Telnet viene superata con l'introduzione del protocollo SSH (Secure Shell). La connessione SSH si basa sulla criptografia asimmetrica detta anche crittografia a coppia di chiavi o, più semplicemente, a chiave pubblica/privata che consiste nella generazione di una coppia di chiavi. In pratica, ciò che viene codificato con la chiave pubblica (a disposizione di tutti) può essere decodificato solo dalla corrispondente chiave privata (in possesso di un'unica persona). La coppia di chiavi viene generata usando degli algoritmi asimmetrici RSA e DSA. Le connessioni che usano tali coppie di chiavi prodotte da questi algoritmi asimmetrici sono dette connessioni SSH. RSA e DSA vengono usati solo per instaurare una connessione cifrata fra il client SSH e il server SSH , per il trasferimento dei dati, vengono usati simmetrici, come AES o 3DES, più efficienti nella codifica nel trasferimento dei dati. La connessione SSH avviene tipicamente sulla porta 22 e, di norma, il server risponde aprendo una shell di comando.

Protocollo SMTP

SMTP (Simple Mail Transfer Protocol) è un protocollo di livelo 7 dello stack TCP/IP che permette di inviare posta elettronica (e-mail) ad utenti della rete. E' un protocollo monodirezionale, infatti dopo aver stabilito la connessione solo il client può inviare email al server che le riceve. Ogni utente è identificato da un indirizzo di posta del tipo***:*** ***nome@gestoredelservizio***.
Per inviare un'email non vi è bisogno di nessuna autorizzazione ed il processo viene ripetuto fino a che il server non diventa raggiungibile.

Protocollo POP3
IL protocollo POPO3 (Post Office Protocol version 3) è il protocollo che permette di ricevere email in una casella postale: la posta rimane in giacenza nello spazio riservato (mail box) del server pop3 a disposizione dell'utente. I messaggi vengono scaricati nel computer locale, rendendo piu' facile la lettura offline e liberando lo spazio occupato sul server di posta. E' possibile decidere se conservare copie dei messaggi sul server. Gli svantaggi derivano dal fatto che ogni volta che ci si connette al server, vengono scaricati tutti i nuovi messaggi (con possibili problemi nel caso di messaggi pesanti e connessioni lente). Se si utilizza piu' di un computer, i messaggi possono essere scaricati nell'uno o nell'altro, ma non in entrambi.
Come nel caso del protocollo SMTP, il protocollo POP3 funziona grazie a dei comandi inviati al server POP. Ciascuno dei comandi inviati dal client (validato dalla sequenza CR/LF) è composto da una parola-chiave, eventualmente accompagnata da uno o più argomenti ed è seguito da una risposta del server POP composta da un numero e da un messaggio descrittivo.

Comandi POP3	
Comando	**Descrizione**
USER identificativo	Questo comando permette di autentificarsi. Esso deve essere seguito dal nome dell'utente. cioè da una stringa di caratteri che identificano l'utente sul server. Il comando USER deve precedere il comando *PASS.*
PASS password	Il comando *PASS* permette di indicare al password dell'utente il cui nome è specificato ad un comando *User* precedente.
STAT	Informazione sui messaggi contenuti sul server
RETR	Numero del messaggio da recuperare
DELE	Numero del messaggio da cancellare
LIST [msg]	Numero del messaggio da visualizzare
NOOP	Permette di mantenere le connessioni aperte in caso di inattività
TOP <messageID> <n>	Comando che visualizza *n* linee di messaggio, il cui numero è dato in argomento. In caso di risposta positiva da parte del server, questo rinvia le intestazioni del messaggio, poi una linea vuota e infine le *n* prime linee del messaggio.
UIDL [msg]	Richiesta al server di rinviare una linea contenente delle informazioni sul messaggio eventualmente dato in argomento. Questa linea contiene una stringa di caratteri, detta *listing d'identificatore unico*, che permette di identificare in modo univoco il messaggio sul server, indipendentemente dalla sessione. L'argomento opzionale è un numero corrispondente ad un messaggio esistente sul server POP, cioè un messaggio non cancellato).
QUIT	Il comando *QUIT* chiede l'uscita del server POP3. Esso implica la cancellazione di tutti i messaggi segnati come eliminati e rinvia lo stato di questa azione.

Il protocollo POP3 gestisce così l'autenticazione attraverso il nome utente e password, ma non è invece sicuro dato che le password, come le mail, circolano in chiaro (in modo non criptato), sulla rete. Si può anche accedere alla propria posta grazie ad un semplice telnet sulla porta del server POP (110 per default) :

```
telnet  ilmioserverpop3  110

S: +OK ilmioserverpop3  POP3 service
S: (Netscape Messaging Server 4.15 Patch 6 (built Mar 31 2001))
C: USER jeff
S: +OK Name is a valid mailbox
C: PASS mia_pass
S: +OK Maildrop ready
C: STAT
S: +OK 2 0
C: TOP 1 5
S: Subject: Ciao
S: Ciao Meandus,
S: come va?
S :
S: A presto!
C: QUIT
S: +OK
```

Protocollo SNMP

SNMP (Simple Network Management Protocol) è un protocollo per la gestione degli apparati di una rete, basato su UDP (porta 161) e IP. SNMP prevede tre componenti:

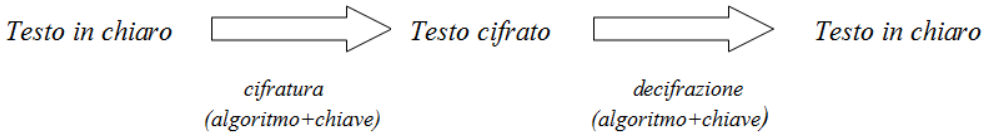
- sistemi gestiti
- agenti di gestione
- sistema di gestione (anche in remoto)

Su ogni sistema gestito deve risiedere un agente di gestione e una MIB (Management Information Base) in cui l'agente trova le variazioni richieste da sistema di gestione. Il sistema di gestione dialoga con i sistemi inviando **richieste SNMP e** ricevendone **notifiche SNMP.** Alcuni esempi di richieste sono:

GET, usata per leggere uno o più dati di MIB;
GETNEXT, usata per leggere iterativamente una sequenza di dati di MIB;
GETBULK, usata per leggere con una sola richiesta grandi porzioni di MIB;
SET, usata per scrivere (modificare) uno o più dati di MIB.
Le notifiche sono messaggi asincroni inviati dall'agent per segnalare eventi occorsi nel sistema gestito (p.es. allarmi in caso di guasti).

La sicurezza nelle reti
La crittografia

Tutte le attività che richiedono riservatezza per svilupparsi su Internet hanno bisogno di garanzie quali
l'autenticità e l'integrità dei messaggi trasmessi.
La crittografia è un insieme di tecniche che consentono di rendere visibili le informazioni soltanto alle persone a cui sono destinate.



Il *testo in chiaro* è quello che può esser letto da tutti, il *testo cifrato* è quello che può, a certe condizioni esser letto dal solo destinatario. Le operazioni *di cifratura* e *decifratura* portano il testo da chiaro in cifrato e viceversa e sono ottenute mediante l'applicazione di un algoritmo più l'uso di una chiave, in genere un numero molto grande.
Una classificazione dei metodi di cifratura è **cifrari a sostituzione** e **cifrari a trasposizione.**
Un esempio di *cifrario a sostituzione* è quello di Giulio Cesare nel quale la lettera chiara viene sostituita dalla lettera che la segue di tre posti nell'alfabeto: la lettera A è sostituita dalla D, la B dalla E e così via fino alle ultime lettere che sono cifrate con le prime come nella tabella che segue.

Chiaro *a b c d e f g h i j k l m n o p q r s t u v w x y z*
Cifrato *d e f g h i j k l m n o p q r s t u v w x y z a b c*

Volendo cifrare per esempio la frase Auguri di buon compleanno otterremo il seguente messaggio cifrato:
Chiaro *auguridibuoncompleanno*
Cifrato *dxjxulglexrqfrpsohdqqr*

Più in generale si dice codice di Cesare un codice nel quale la lettera del messaggio in chiaro viene spostata di un numero fisso di posti, non necessariamente tre.
Un *cifrario a trasposizione* è un metodo di cifratura in cui le posizioni occupate dalle lettere o gruppi di esse del testo in chiaro sono cambiate secondo un determinato schema, così che il testo cifrato costituisca una permutazione del testo in chiaro.
Un esempio è il *cifrario a percorso* in cui il testo in chiaro viene prima scritto in una griglia di dimensioni prefissate e poi letto seguendo uno schema dato dalla chiave. Ad esempio, se il testo da cifrare è *piantare il campo dietro la collina* potremmo avere:

p	a	c	d	o	l
i	r	a	i	l	l
a	e	m	e	a	i
n	i	p	t	c	n
t	l	o	r	o	a

La chiave potrebbe specificare di "*leggere in spirali concentriche, in senso antiorario, partendo dall'angolo in alto a destra*". Il testo cifrato sarà : *lodcapiantloroanilliareiptcaem*

Crittografia a chiave simmetrica

I sitemi di crittografia a chive simmetrica prevedono l'uso *di una sola chiave per cifrare e decifrare* il messaggio, quindi mittente e destinatario devono conoscere la stessa chiave per potersi scambiare i messaggi. Un semplice esempio è il metodo che prevede la codifica e la decodifica sulla base della considerazione che applicando 2 volte l'operatore XOR si ottiene il messaggio di partenza:

(messaggio XOR chiave) XOR chiave = messaggio

chi trasmette	chi riceve
messaggio XOR chiave = cifrato	cifrato XOR chiave = messaggio
110011010 XOR 101110001 = 011101011	011101011 XOR 101110001 = 110011010

Un sistema crittografico a chiave simmetrica molto conosciuto è il DES (Data Encryption Standard) che utilizza una chiave a 56 bit. Successivamente per renderlo più sicuro è stato introdotto il Triple DES che ha una chiave più lunga.
Nei sistemi di crittografia a chiave simmetrica è importante la segretezza della chiave, che comunque è a conoscenza sia del mittente che del destinatario. In questo caso è fondamentale il problema di consegnare la chiave in modo *sicuro* cioè senza che altri possano in qualche modo venirne a conoscenza.

Crittografia a chiave asimmetrica

In questo caso vengono generate da appositi software 2 chiavi :
chiave pubblica questa chiave è contenuta il un database pubblico a disposizione di tutti
chiave privata questa chiave è a conoscenza del solo richiedente

Le due chiavi sono correlate matematicamente, per cui i messaggi codificati con la chiave pubblica possono esser decodificati solo da chi possiede la *corrispondente* chiave privata. La sicurezza di questo sistema e che dalla chiave pubblica si può risalire a quella privata ma con una capacità di calcolo che richiede tempi molto elevati. L'uso delle 2 chiavi combinate determina diversi livelli di sicurezza:

Garanzia dell'identità del mittente
Il mittente con la sua chiave privata codifica il messaggio e lo invia al destinatario che con la chiave pubblica lo decodifica. In questo caso è garantita l' identità del mittente (solo la chiave pubblica del mittente può decodificare il messaggio) ma non la segretezza, in quanto tutti hanno a disposizione la chiave pubblica.

Garanzia della segretezza
Il mittente codifica il messaggio con la chiave pubblica del destinatario e lo invia. Solo chi riceve il messaggio può decodificarlo con la sua chiave privata. In questo caso è garantita la segretezza ma non l'identita del mittente (il messaggio viene codificato con la chiave pubblica del destinatario a disposizione di tutti).

Garanzia dell'identita e della segretezza
E' una combinazione dei due precedenti metodi. Il mittente codifica 2 volte in messaggio, prima con la sua chiave privata poi con quella pubblica del destinatario. Il ricevente decodifica il messaggio prima con la sua chiave privata e poi con quella pubblica del mittente.

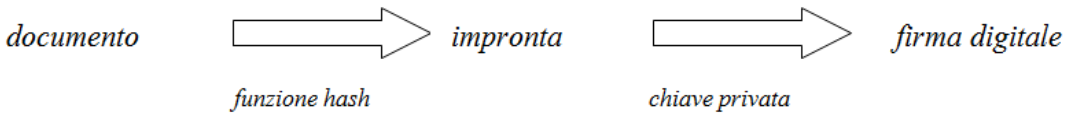
La firma digitale

E' un metodo che permete ad una persona di apporre un segno distintivo personale a dei documenti in formato elettronico. Tramite la firma digitale si possono garantire :

autenticità la certezza che il messaggio sia stato scritto da una determinata persona
integrita che il documento non sia stato modificato durante la trasmissione
non ripudiabilità che la persona che ha trasmesso il messaggio non possa disconoscerlo

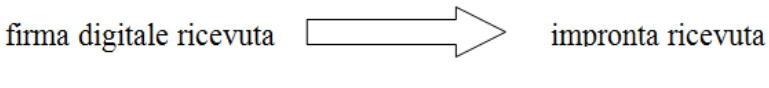
La firma digitale viene realizzata utilizzando il sistema di cifratura a chiave asimmetrica, solo che le due chiavi vengono utilizzate in modo differente, si firma con la privata, si controlla con la pubblica.

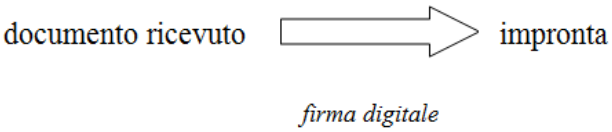
Procedura per generare la firma digitale



La funzione hash è tale da generare una sequenza di 128 o 160 bit (impronta) UNICA per ciascun documento, cioè ad una semplice modifica del documento originale causa la generazione di un'impronta diversa. Al documento viene accodata la firma digitale e quindi il tutto può essere inviato al destinatario.

Procedura verificare la firma digitale
Il destinatario che vuol verificare la firma dovrà fare due operazioni:





Se impronta ricevuta è uguale all'impronta calcolata dal ricevente la firma è corretta.

La coppia di chiavi per avere un valore legale deve essere rilasciata da un Ente Certificatore che la rilascia solo dopo aver verificato con certezza l'identità del richiedente. Esempi di enti sono : Aruba Posta Elettronica Certificata S.p.A. , Banca d'Italia,Intesa Sanpaolo S.p.A ed altri elencati dall'Agenzia delle Entrate all'indirizzo *http://www.digitpa.gov.it/firma-digitale/certificatori-accreditati/certificatori-attivi*
Un esempio di programma per generare le chiavi sotto licenza GNU GPL è *gpg* (GNU Privacy Guard).
MD5 (Message Digest) è un programma che genera un'impronta da 128 bit.

Tecniche di sicurezza in IP

Tecniche di Tunneling

Il tunneling “incapsula” pacchetti di un generico protocollo in un’ altro trasportabile in internet, prima dell’incapsulamento generalmente i pacchetti vengono crittografati (in questo caso si parla di **VPN** Virtual Private Network). Per le reti IP si ha l’incapsulamento di pacchetti non IP (o IP con caratteristiche particolari) all’interno di un datagramma IP.
Con il tunneling si ottengono numerosi vantaggi :

- sicurezza : tunneling + cifratura permettono di ottenere collegamenti sicuri su reti pubbliche.
- attraversamento di segmenti IP da parte di pacchetti di protocolli diversi
- mobile IP

NETWORK ADDRESS TRASLATION

Il NAT è una tecnica che ci offre la possibilità di modificare gli indirizzi IP, presenti nei pacchetti, in transito su una specifica interfaccia di un router.

TERMINOLOGIA

- **Inside Local Address**: è l'indirizzo con il quale un **host nella rete** si presente al router prima di essere tradotto. Tipicamente sono gli indirizzi della rete interna, univoci solo all'interno della rete stessa.

- **Inside Global Address**: è l'indirizzo con il quale un **host nella rete** è visto da internet dopo essere stato tradotto.

- **Outside Local Address**: è l'indirizzo con il quale un **host remoto** si presenta sul router prima di essere tradotto. Tipicamente è l'indirizzo stesso dell'host a meno che anche in remoto non sia attivato il NAT

- **Outside Global Address**: è l'indirizzo con il quale un **host remoto** si presenta all'interno della rete dopo essere stato tradotto.

CONFIGURAZIONE

Per la configurazione del NAT sui router Cisco è necessario definire la posizione delle interfacce interessate al NAT:

- **Inside**: solitamente l'interfaccia interna
- **Outside**: solitamente l'interfaccia esterna che punta verso internet

Per la configurazione delle interfacce useremo i seguenti comandi:

Router(config)#interface FastEthernet 0/0	interfaccia verso l’ inside network
Router(config-it)#ip nat inside	
Router(config)#interface FastEthernet 0/1	interfaccia verso l' outside network
Router(config-it)#ip nat outside	

Occorre poi definire quale tipologia di NAT si vuole utilizzare, sui router Cisco è possibile implementare NAT di tipo statico, dinamico e con overloading.

NAT STATICO (ONE TO ONE)

Il NAT statico definisce una traduzione degli indirizzi uno a uno.

Se vogliamo pubblicare l'indirizzo IP interno (inside) 192.168.10.85 rendendolo raggiungibile tramite l'indirizzo IP esterno (inside global) 1.2.3.4 useremo il comando

router(config)#ip nat inside source static 192.168.10.85 1.2.3.4

Così facendo il router Cisco tradurrà l'indirizzo IP interno (inside) 192.168.10.85 nell'indirizzo IP esterno 1.2.3.4 e di conseguenza tutte le richiesta che perverranno dall'esterno sull'indirizzo io 1.2.3.4 verranno inoltrate all'host interno 192.168.10.85

Se vogliamo pubblicare internamente l'indirizzo IP esterno (outside) 1.2.3.4 sull'indirizzo IP interno (outside local) 192.168.10.85 useremo il comando

router(config)#ip nat outside source static 1.2.3.4 192.1681.10.85

Così facendo il router Cisco tradurrà l'indirizzo IP esterno (outside) 1.2.3.4 con l'indirizzo IP interno (outside local) 192.168.10.85 e potremo lavorare sull'host remoto come se questo fosse connesso direttamente alla nostra rete interna.

NAT DINAMICO (MANY TO MANY)

Il NAT dinamico può essere considerato come una variante del NAT statico in quanto la traduzione viene sempre eseguita in modalità uno a uno ma la sua particolarità sta nel fatto che l'indirizzo IP associato viene scelto dinamicamente prendendolo da un pool di indirizzi IP.

Se vogliamo fare in modo che ogni host della nostra rete interna (inside) raggiunga internet utilizzando un indirizzo IP univoco (inside global) per host, e abbiamo a disposizione un range di 19 indirizzi IP pubblici dovremo:

1. Creare una ACL che permetta il traffico a tutti gli host della rete;
2. Creare un pool di indirizzi esterni che rappresenti il range di indirizzi IP pubblici che ci sono stati assegnati;
3. Configurare il NAT dinamico affichè esponga gli indirizzi IP interni come indirizzi IP esterni.

Per questo tipo di configurazione utilizzeremo i comandi

router(config)#access-list 98 permit 192.168.10.0 0.0.0.255

router(config)#ip nat pool POOL_INDIRIZZI 1.2.3.1 1.2.3.19 netmask 255.255.255.0

router(config)# ip nat inside source list 98 pool POOL_INDIRIZZI

Con questo particolare tipo di NAT va posta attenzione poiché gli indirizzi IP esterni (inside global) rimangono associati agli indirizzi IP interni (inside) finché non scade il tempo di timeout del NAT e inoltre, qualora si presentasse un ulteriore host e i 19 indirizzi IP pubblici (inside global) risultassero già occupati, quest'ultimo non riuscirà a collegarsi ad internet finché un indirizzo IP

pubblico non sarà rilasciato.

OVERLOADING NAT (MANY TO ONE)

Il NAT overloading permette l'associazione di tradurre più indirizzi IP interni (inside) con un unico indirizzo IP esterno (inside global).

L'esempio classico si verifica quando tutti gli host di una rete (inside) si presentano esternamente con un unico indirizzo IP (inside global). Se vogliamo creare un NAT overloading useremo il comando

Router(config)#access-list 98 permit 192.168.10.0 0.0.0.255

Router(config)#ip nat inside source list 98 interface FastEthernet 0/1 overload

Così facendo il router Cisco tradurrà tutti gli indirizzi IP interni (inside) che raggiungeranno l'esterno tramite l'interfaccia FastEthernet0/1 e tutti gli host della rete interna si presenteranno all'esterno con un solo indirizzo IP (inside global).

VERIFICHE NAT

In qualsiasi momento possiamo eseguire il debug sulle configurazioni di NAT presenti sul router Cisco utilizzando i seguenti comandi

router-Cisco#show ip nat translations

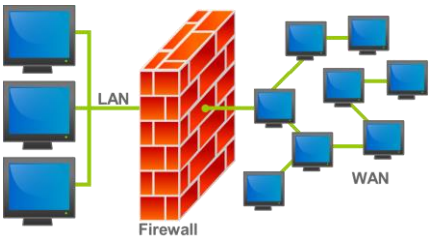
router-Cisco#show ip nat statistics

router-Cisco#debug ip nat detailed

*router-Cisco#clear ip nat translation **

Firewall

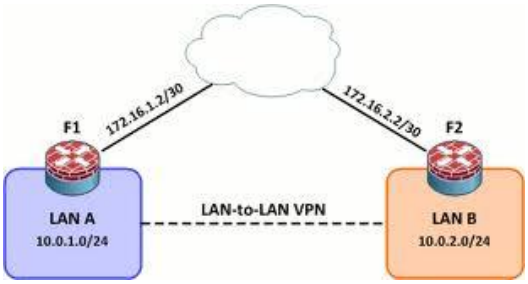
Un firewall è uno degli strumenti di sicurezza più efficaci per proteggere gli utenti della rete interna da minacce esterne. Un firewall, interposto tra due o più reti controlla il traffico e può anche prevenire accessi non autorizzati. I firewall possono essere classificati in base al tipo di regole che applicano per filtrare i pacchetti:
Packet Filtering - Impedisce o consente l'accesso sulla base di indirizzi IP o MAC (livelli 2 e 3).
Application Filtering - Impedisce o consente l'accesso in base ai numeri di porta (levello 4).
Filtri URL - Impedisce/consente l'accesso in base all' *URL* o *parole specifiche* presenti nel testo.
Stateful Packet Inspection (SPI) - Impedisce l'ingresso ai pacchetti *che non sono risposte* a richieste di host interni alla rete. L'amministatore può comunque attraverso politiche specifiche gestire casi particolari che consentano l'accesso anche a pacchetti che non sono risposta a richieste interne alla rete. SPI può includere anche la capacità di riconoscere e filtrare determinati tipi di attacchi come il DoS.
Misti – un firewall che include più funzioni viste in precedenza



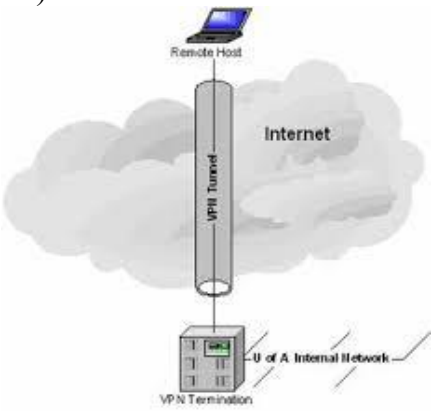
VPN – Virtual Private Network

Le VPN sono reti virtuali private all'interno della rete pubblica Internet,e si può considerare come l'estensione di una rete aziendale attraverso Internet, che stabilisce una connessione sicura, attraverso il tunelling e la cifratura dei pacchetti. LaVPN può essere :

LAN to LAN connessione sicura fra due LAN remote (es. due reti di due sedi distaccate di un'azienda)



HOST to LAN connessione sicura fra un host e una rete (es. rappresentante che si connete alla sede centrale per inoltrare un'ordine)



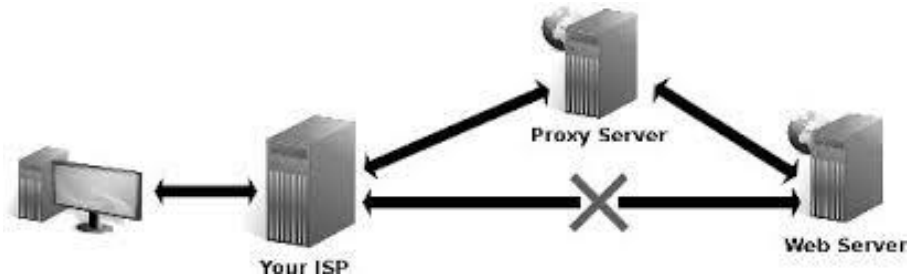
Le caratteristiche di una VPN sono:

- **Autenticazione** dell'origine dei dati
- **Impossibilita del mittente di disconoscere** i dati inviati
- **Integrità** dei dati che non possono esser modificati durante il trasporto
- **Riservatezza**, la garantisce la crittografia
- **Protezione** nessun attacco può intercettare i datagrammi
- **Gestione delle policy** implementazione semplice delle politiche di sicurezza
- **Scalabilità** la VPN può cresce col crescere dell'azienda in modo semplice

Proxy server

E' un applicativo che si colloca tra un client ed un server facendo da tramite tra due host ovvero inoltrando le richieste e le risposte dall'uno all'altro. Il client si collega al proxy invece che al server,

e gli invia delle richieste. Il proxy a sua volta si collega al server e inoltra la richiesta del client, riceve la risposta e la inoltra al client.



A differenza di bridge e router, che lavorano ad un livello ISO/OSI più basso,i proxy generalmente lavorano a livello applicativo gestendo un numero limitato di protocolli. Per questo motivo si parla di proxy http, proxy ftp ecc..ecc..

Proxy cache

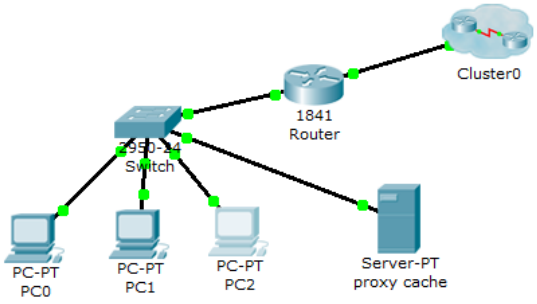
Un cache proxy è un servizio di memorizzazione locale degli URL richiesti più frequentemente nella rete.

L'utilizzo di un proxy offre due vantaggi principali:

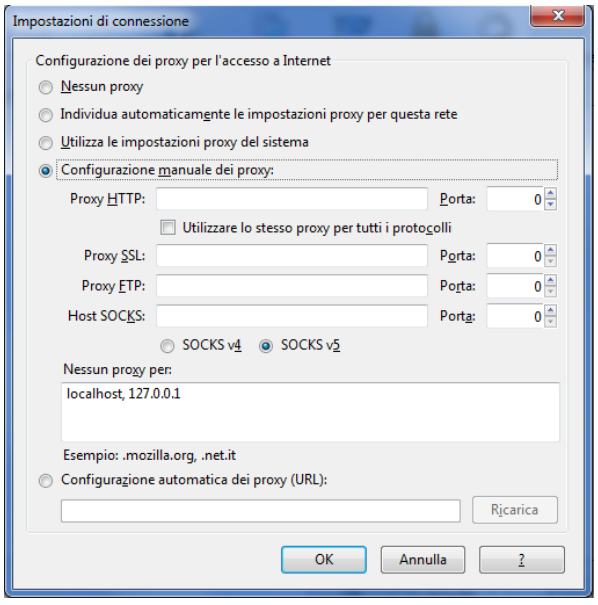
- accesso rapido a risorse presenti nella memoria cache
- riduzione del traffico nella rete che precede il proxy stesso.

Il servizio di cache proxy può essere collocato in posizioni differenti nella rete, a seconda delle esigenze o delle particolarità delle situazioni.

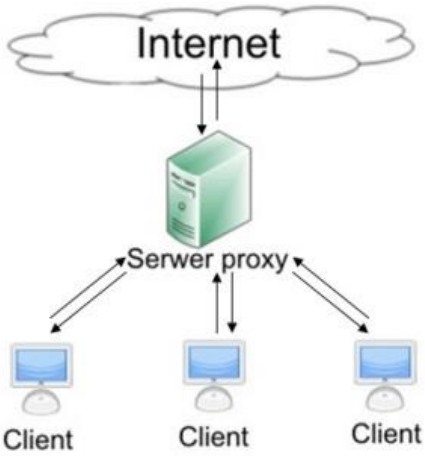
Una prima possibilità è il proxy che serve una rete LAN collegata ad Internet.



In questo caso il client dovrà esser configurato per far uso del proxy. Ad esempio in Firefox: strumenti / opzioni / avanzate rete / impostazioni



Nel secondo caso il proxy è utilizzato come filtro. Il server è connesso, attraverso due interfacce di rete, a due reti differenti ad esempio una LAN e Internet, quindi un client per accedere ad Internet dovrà necessariamente attraversare il proxy che potrà quindi filtrare il traffico. Questo tipo di configurazione accesso si limita ai protocolli gestiti dal proxy; spesso si tratta solo di HTTP e FTP.



In questo caso si parla di *proxy trasparente*, cioè non è necessario configurare i client come nel caso precedente.

Proxy http

Un caso in cui viene spesso usato un proxy è la navigazione web. Un proxy può essere usato diversi motivi che potremmo sintetizzare:

privacy: un proxy può garantire un maggiore livello di privacy mascherando il vero indirizzo IP del client in modo che il server non venga a conoscenza di chi ha effettuato la richiesta.

connettività: per permettere ad una rete privata di accedere all'esterno è possibile configurare un computer in modo che faccia da proxy tra gli altri computer e Internet, in modo da mantenere un unico computer connesso all'esterno, ma permettere a tutti di accedere. In questa situazione, solitamente il proxy viene usato anche come firewall.

caching: come già detto è un servizio di memorizzazione di URL richiesti dagli host della rete, e permette

- miglioramento delle prestazioni
- riduzione del consumo di ampiezza di banda.

monitoraggio: un proxy può permettere di tenere traccia di tutte le operazioni effettuate (ad esempio, tutte le pagine web visitate), consentendo :

statistiche

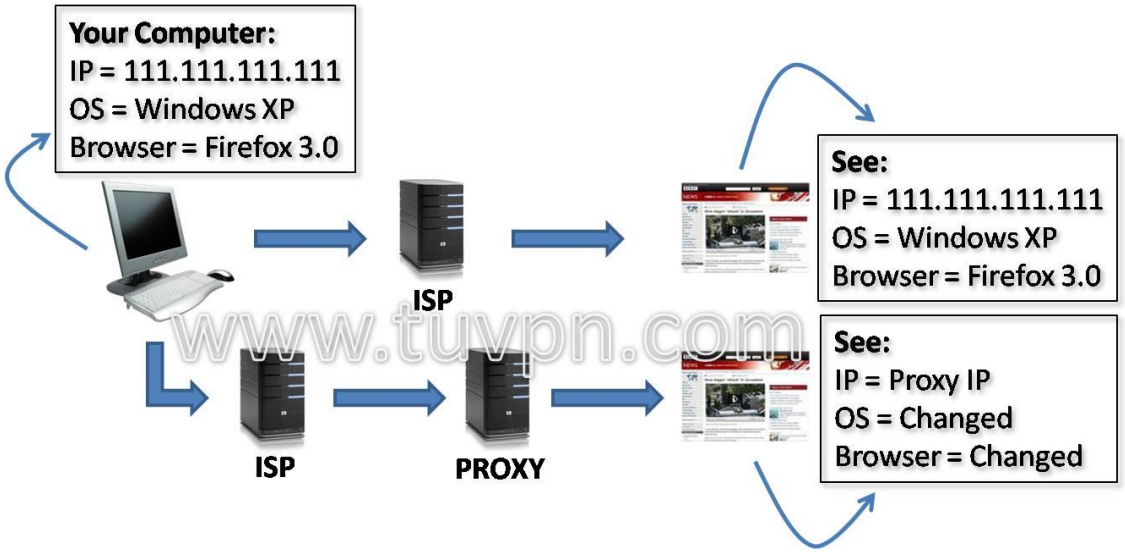
controllo del traffico allo scopo di verificare utilizzi delle rete che possano violare la privacy degli utenti.

controllo: un proxy può applicare regole definite dall'amministratore di sistema per determinare quali richieste inoltrare e quali rifiutare, oppure limitare l'Ampiezza di banda utilizzata dai client, oppure filtrare le pagine Web in transito, ad esempio bloccando quelle il cui contenuto è ritenuto offensivo in base a determinate regole.

Proxy HTTP e anonimato

I server esterni a cui si collega il client quando si utilizza un proxy vedranno generalmente l'indirizzo IP del proxy (e non quello del client). Se l'uso di un proxy garantisce una relativa privacy del client (il server esterno, o chi analizzi il traffico diretto ad esso, non potrà infatti conoscere l'indirizzo IP del client), può impedire la connessione a quei siti che utilizzino l'indirizzo IP del client per scopi di autenticazione o di riconoscimento delle sessioni (come ad esempio nei collegamenti agli sportelli bancari on-line).

Il protocollo HTTP prevede però che un proxy possa inserire nelle richieste che inoltra al server degli header standardizzati, che permettono di riconoscere che la richiesta è stata inoltrata da un proxy, e possono contenere anche l'indirizzo IP del client, che in questo modo può essere noto ad un server opportunamente configurato. Quando viene usata questa funzionalità, il server web "si fida" dell'indirizzo del client inviatogli dal proxy, e non può in alcun modo verificare questa informazione. L'amministratore di un server proxy può decidere se inviare o meno questi header determinando quindi il livello di anonimato del proxy.
I proxy HTTP, a seconda dell'anonimato che riescono a fornire, possono essere suddivisi in:



- **NOA** (non anonymous proxy) proxy non anonimi (o "trasparenti"): Modificano alcuni header trasmessi dal browser e ne aggiungono altri, mostrano anche l'indirizzo IP reale del richiedente.
- **ANM** (anonymous proxy server) proxy anonimi: non trasmettono l'IP del richiedente, ma modificano o aggiungono alcuni header. Sono pertanto facilmente riconoscibili.

- **HIA** (high anonymous proxy) proxy altamente anonimi: non trasmettono l'IP del richiedente e non modificano gli header della richiesta. Sono difficili da riconoscere attraverso i normali controlli.
- **Proxy distorcenti**: trasmettono un IP casuale, diverso da quello del richiedente e modificano o aggiungono alcuni header. Solitamente vengono scambiati per proxy Anonimi, ma offrono una protezione maggiore, in quanto il server web vede le richieste di un utente provenienti da indirizzi IP diversi.

Nonostante un proxy anonimo i cookies possono permettere al server di ottenere informazioni sul client anche se questi si collega da reti diverse
Per avere conferma se il proxy server consente una navigazione anonima, ossia se non rivela l'IP del client a nessun altro server della rete, è bene effettuare un controllo accedendo ad uno dei numerosi siti che restituiscono l'indirizzo di IP se esso e diverso da quello dell'host (cioè quello del proxy) il test di anonimizzazione è positivo.

Approfondimento

ACCESS CONTROL LIST

Le ACL (Access Control List) sono una lista di istruzioni da applicare alle interfacce di un router allo scopo di gestire il traffico, filtrando i pacchetti in entrata e in uscita.

Le ACL vengono usate per:
- **Implementare sicurezza** restringendo, ad esempio, gli accessi a una determinata rete o sottorete;
- **Limitare il traffico e aumentare la performance della rete**: si può, infatti, decidere che alcuni pacchetti vengano processati prima di altri.
- **Decidere il tipo di traffico** può essere trasmesso: si può permettere l'invio di e-mail e impedire allo stesso tempo il Telnet.

Una ACL prevede due elementi :
una **condizione** **c**he il pacchetto può soddisfare o meno
un' **azione** di *accept* o *deny* sul pacchetto intercettato dalla condizione

Le *ACL vengono elaborate dal router in maniera sequenziale in base all'ordine in cui sono state inserite.* Appena un pacchetto soddisfa una delle condizioni, la valutazione s'interrompe e il resto delle ACL non viene preso in considerazione. Il pacchetto viene quindi inoltrato o eliminato secondo l'istruzione eseguita. Se il pacchetto non soddisfa nessuna delle condizioni viene scartato (si considera cha alla fine di un ACL non vuota ci sia l'istruzione deny any ovvero nega tutto).

Posizionamento dell ACL
Uno degli aspetti fondamentali da ricordare è il *posizionamento delle ACL*. Infatti il cattivo posizionamento delle ACL può aver un impatto negativo sulla rete in termini di :
- maggiori risorse utilizzate dal router nel processare i pacchetti;
- decadimento delle prestazioni della rete;
- collasso della rete stessa.

Due sono gli aspetti fondamentali da tenere in considerazione nel posizionamento di un'ACL:
- I pacchetti scartati o accettati non andranno a utilizzare risorse del router, quindi inserire prima le ACL che si riferiscono a pattern molto frequenti. ad esempio prima i pattern relativi alla porta 53 o 80 rispetto a quelli che si riferiscono alla porta 110 o 20-23 ;
- Processare le ACL richiede risorse aggiuntive per il router (quindi non bisogna abusare delle ACL).

Cisco consiglia che le

ACL *standard* (che specificano solo la sorgente del traffico) siano posizionate più vicino alla *destinazione* per almeno due motivi:
- Per non bloccare in partenza anche il traffico verso destinazioni autorizzate.
- Non sapendo il percorso fatto dai pacchetti, conviene proteggere la rete di destinazione mettendosi al suo ingresso.

ACL *estese* siano posizionate più vicino alla *sorgente*, ricucendo il traffico inutile in rete.

ACCESS CONTROL LIST standard

Le ACL standad sono quelle con *access-list number* compreso fra 1 e 99 che operano il controllo esclusivamente sull'indirizzo sorgente.

Vengono utilizzate per bloccare o permettere il traffico da una rete o da un host specifico o per negare una insieme di protocolli. La sintassi del comando Cisco è

R1(config)# **access-list** *access-list number* {**permit | deny**} *indirizzo-sorgente* [*wildcard*] [**log**]
dove

access-list-number	numero della ACL da 1 a 99
permit/deny	permette / nega l'accesso se le condizioni sono verificate
indirizzo-sorgente	del pacchetto
wildcard	wildcard mask che deve essere applicata all'indirizzo sorgente (opzionale)
log	(Opzionale) Attiva i messaggi di log. Questi comprendono l'indirizzo sorgente, il numero di pacchetti e l'esito del controllo (permit o deny). I log vengono generati a intervalli di 5 minuti.

Una volta definite le condizioni si deve applicare la ACL all'interfaccia desiderata, secondo la seguente sintassi:

R1(config-if)# **ip access-group** *access-list-number* { **in | out** }

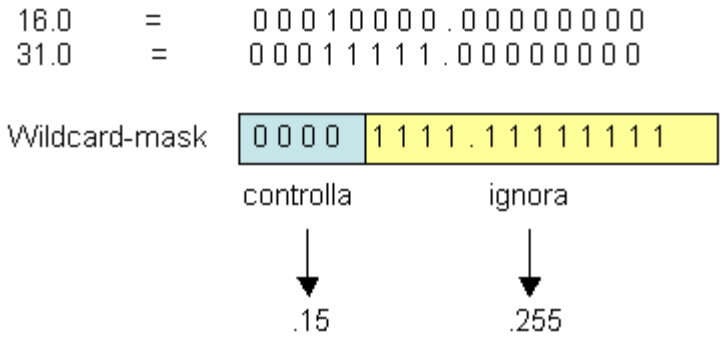
access-list-number indica il numero della ACL che deve essere legata all'interfaccia.

in|out Specifica se l'ACL deve essere applicata ai pacchetti in entrata all'interfaccia (in) o applicata ai pacchetti in uscita all'interfaccia (out). Se non è specificato, per *default è out*.
Per eliminare una ACL bisogna utilizzare il comando

R1(config)# **no access-list** *access-list number*
R1(config-if**)# no ip access-group** *access-list number*

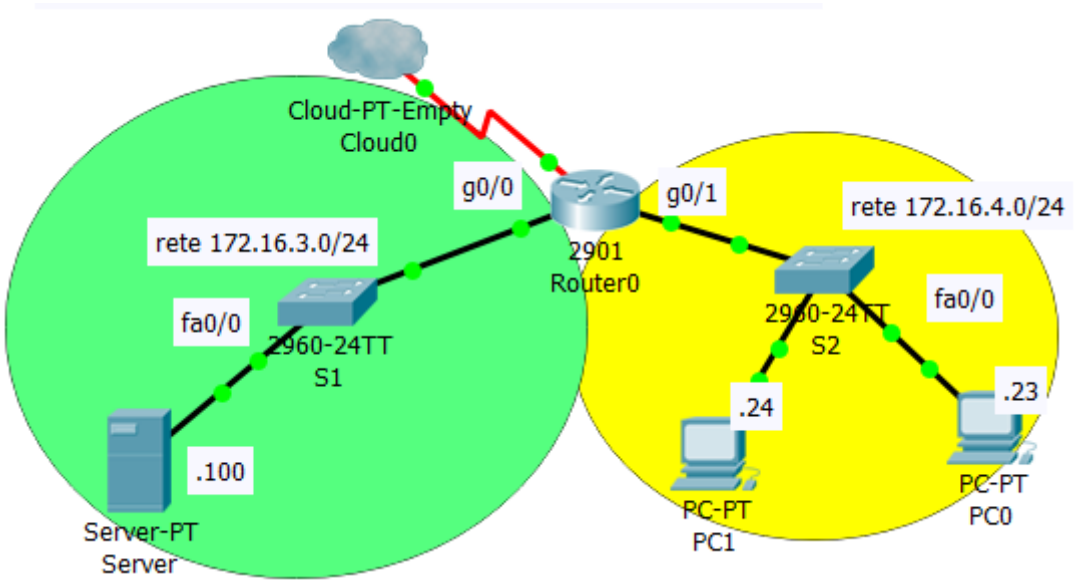
La wildcard mask

La wildcard mask è un numero di 32-bit diviso in 4 ottetti che indica quali bit di un indirizzo IP devono essere controllati in un'ACL. Se il bit è:
0 il bit corrispondente deve essere controllato,
1 che deve essere ignorato.
Esempio.
Si supponga di voler testare gli indirizzi IP da 172.50.16.0 a 172.50.31.0.
In questo caso conviene analizzare esclusivamente gli ultimi due ottetti:



Per cui, in questo caso, la wildcard-mask avrà la forma 0.0.15.255.
Nel caso si vogliano controllare tutti gli indirizzi (Cisco li indica con l'IP 0.0.0.0) si può utilizzare il termine **any**; per un singolo indirizzo si utilizza il termine **host**.

R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255
any
R1(config)# access-list 1 permit 172.30.24.12 0.0.0.0
host 172.30.24.12



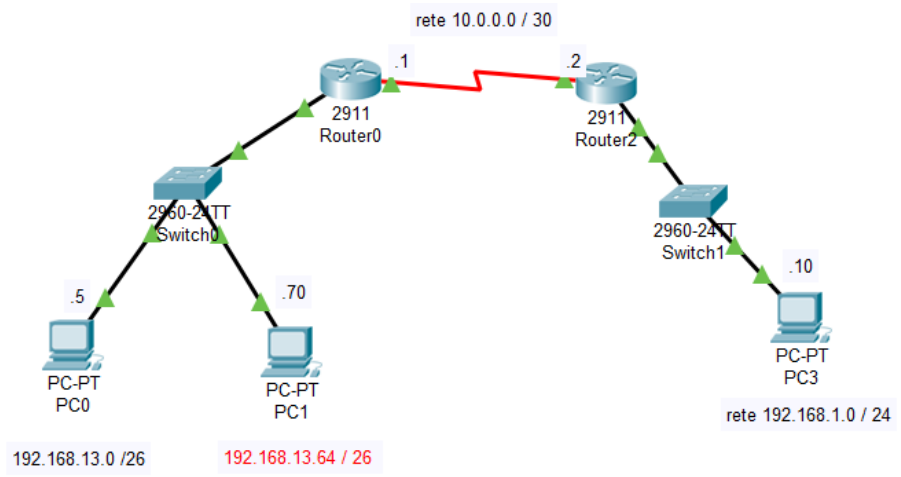
Esempio:

Bloccare il traffico che parte dall'host 172.16.4.23 verso la rete 172.17.3.0 /24.

R1(config)# ***access-list 2 deny*** *172.16.4.23 0.0.0.0*
R1(config)# ***access-list 2 permit*** *0.0.0.0 255.255.255.255*
(*access-list 2 deny any* è implicita quindi non è necessario scriverla)
.....
R1(config-if)# ***interface*** *fa0/0*
R1(config-if)# ***ip access-group 2 out***

Esercizi sulle Access list standard

Esercizio 1
Attraverso l'uso di ACL standard si vuole impedire alla rete 192.168.13.0 / 26 di accedere alla 192.168.1.0/24.



Esercizio 2

Con riferimento allo scenario dell'esercizio 1, impedire agli host 192.168.13.10 / 26 e 192.168.13.69 / 26 di accedere alla 192.168.1.0/24

ACCESS CONTROL LIST estese

Le ACL estese possono effettuare il controllo più esteso

- indirizzo del mittente e destinatario
- su uno specifico protocollo
- sul numero di porta
- su altri parametri.

Configurazione di un'ACL Estesa
Il comando per definire un'ACL Estesa sui router Cisco è:
Router(config)# **access-list** access-list-number {**deny | permit**}
protocol
source wildcard
destination wildcard
[**operator** operand] [**established**] [**precedence** precedence] [**log**]

Parametri	Descrizione
Access-list-number	Numero dell'ACL. Ne indica il nome e il tipo (es. da 100 a 199 e da 2000 a 2699
Permit / Deny	Permette / nega l'accesso se le condizioni sono soddisfatte
Protocol	Il protocollo di comunicazione; es. IP, TCP, UDP, ICMP, IGRP, …
Source e Destination	Indirizzo del mittente e del destinatario.
wildcard	wildcard mask applicata al sorgente e al destinatario
operand	Un operatore logico: **lt, gt, eq, neq, range (***less than, greater than, equal, not equal, range***)** e il numero o il nome della porta TCP o UDP Nel caso dell'operatore range occorre inserire due valori operand (es. range 21 25)
Established (opzionale)	Solo per TCP: indica una "*estabilished connection*". Il controllo viene effettuato solo se il segmento ha settato il bit di ACK o RST, mentre non ci sono controlli sul pacchetto iniziale per stabilire la connessione.
Precedence (opzionale)	Indica un numero da 0 a 7, che specifica la precedenza del pacchetto rispetto a un altro (Queuing).
Log (opzionale)	Attiva i messaggi di log. Questi comprendono l'indirizzo sorgente, il numero di pacchetti e l'esito del controllo (permit o deny). I log vengono generati a intervalli di 5 minuti.

Una volta definite le condizioni si deve applicare la ACL all'interfaccia desiderata, secondo la seguente sintassi:

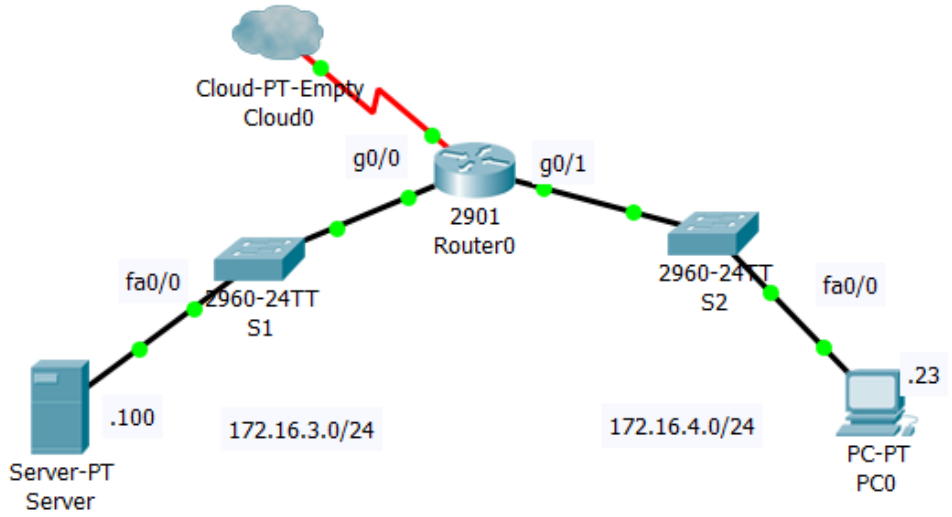
R1(config-if)# **ip access-group** *access-list number* { **in|out** }

Access-list-number: Indica il numero della ACL che deve essere legata all'interfaccia.

in|out Specifica se l'ACL deve essere applicata ai pacchetti in entrata all'interfaccia (in) o applicata ai pacchetti in uscita all'interfaccia (out). Se non è specificato, per *default è out*.

In caso di cancellazione della ACL rimuoverla prima dall'interfaccia/e a cui e associata e poi cancellarla.

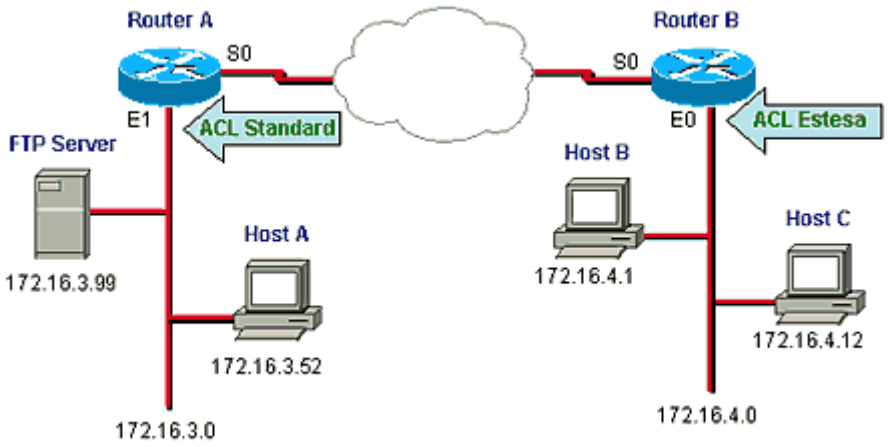
Porta	Keyword	Descrizione	TCP/UDP
20	FTP-DATA	FTP (data)	TCP
21	FTP	FTP	TCP
23	TELNET	Terminal connection	TCP
25	SMTP	SMTP	TCP
42	NAMESERVER	Host Name Server	UDP
53	DOMAIN	DNS	TCP/UDP
69	TFTP	TFTP	UDP
70		Gopher	TCP/IP
80		WWW	TCP



Esempio 1
Vogliamo bloccare il traffico FTP proveniente dalla rete 172.16.4.0 / 16.
R1(config)# ***access-list 101 deny tcp*** *172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255* ***eq*** *21*
R1(config)# ***access-list 101 permit ip*** *172.16.4.0 0.0.0.255* ***any***
(access-list 101 ip deny any any è implicita si omette)
.....
R1(config-if)# ***interface GigabitEthernet0/1***
R1(config-if)# ***ip access-group 101 in***

Esempio 2
In questo esempio viene bloccato sull'interfaccia f0/0 tutto il traffico a esclusione della posta elettronica.
R1(config)# **access-list** *102* **permit tcp** *172.16.3.0 0.0.0.255* **any eq 25** (o **eq smtp**)
(l' acl access-list 102 deny any any è implicita si omette)
.....
R1(config-if)# **interface** *fa 0/0*
R1(config-if)# **ip access-group** *102* **out**

Esempio 3
Si vuole negare all'host B l'accesso al Server FTP e allo stesso tempo negare all'host C qualsiasi accesso alla rete 172.16.3.0. Inoltre si vuole bloccare il servizio FTP per l'host 172.16.4.15.



RouterA(config)#**access-list** 1 **deny host** 172.16.4.12
RouterA(config)#**access-list** 1 **permit any**

RouterA(config)#**interface** ethernet 1
RouterA(config-if)#**ip access-group** 1

RouterB(config)#**access-list** 101 **deny tcp host** 172.16.4.15 172.16.3.0 0.0.0.255 **eq ftp**
RouterB(config)#**access-list** 101 **permit ip** 172.16.4.0 0.0.0.255 **any**

RouterB(config)#**interface** ethernet 0
RouterB(config-if)#**ip access-group** 101

Verificare le ACL
Per visualizzare le informazioni sulle ACL si possono utilizzare i seguenti comandi:

Router> **show access-list** [*access-list number*]
mostra il contenuto di tutte le ACL caricate sul router (utilizzando l'opzione *access-list number* vengono elencate solo le condizioni di una determinata ACL);

Router> **show ip interface** [*interface-type number*]
mostra le informazioni sulle interfacce IP e quindi anche la presenza di un'eventuale ACL collegata all'interfaccia (le opzioni interface-type e number permettono di indicare una determinata interfaccia).

Regole fondamentali

- Quale scopo che si vuole ottenere con le ACL;
- Inserire le clausole più restrittive e quelle più utilizzate all'inizio della lista;
- E' consigliabile posizionare le ACL *standard* sull'interfaccia più vicine alla *destinazione*
- E' consigliabile posizionare le ACL *estese* più vicine alla *sorgente*.
- Scrivere le ACL con un editor di testo e poi esportarle sul router;
- Ricordare che le ACL, "rubano" tempo di CPU;

Nota: le access-list si applicano ai pacchetti che viaggiano attraverso un router. Non sono progettate per bloccare i pacchetti originati all'interno del router. Per impostazione predefinita, un ACL in uscita non impedisce le connessioni di accesso remoto avviate dal router.

Named ACLs

Le ACLs viste finora sono dette **numbered** poiché la singola ACL è identificata da un numero. Oltre ad esse vi sono le ACLs basate sul nome, dette **named ACLs**. Le ACLs named possono essere sia standard che estese e il tipo va specificato in sede di configurazione nel seguente modo:

Router(config)# **ip access-list** {**standard | extended**} *nome-acl*

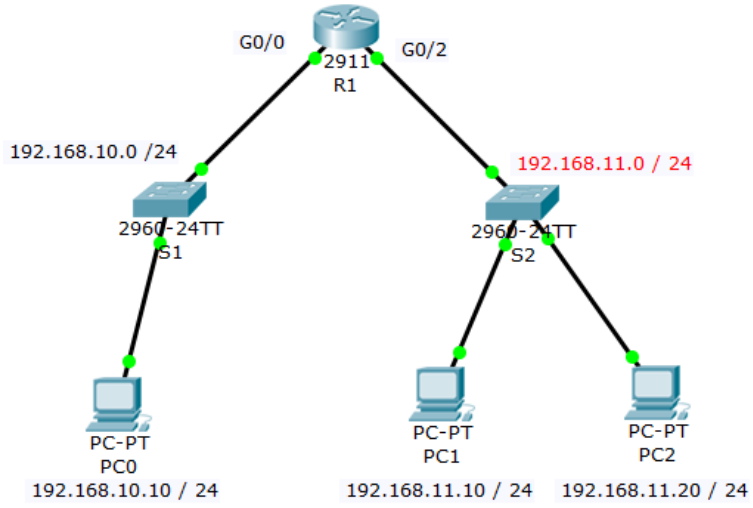
Nelle named ACLs il comando access-list è preceduto dal comando **ip**
Dopo la dichiarazione della ACL named si entra in modalità di configurazione della ACL

Router(config-std-nacl)# [**permit|deny|remark**] {sorgente[wildcard]}[**log**]

Naturalmente come nelle ACL numerate si dovrà applicare ad una specifica interfaccia la ACL con
Router (config) # **interface** numero.interfaccia
Router (config-if) # **ip access.group** nome-acl [**in|out**]

Esempio

Definire una ACL di nome NO-ACCESS che nega l'accesso alla rete 192.168.10.0 all'host 192.168.11.10



Router (config) # **ip access-list standard** NO_ACCESS
Router (config-std-nacl) # **deny host** 192.168.11.10
Router (config-std-nacl) # **permit any**
Router (config-std-nacl) # **exit**
Router (config) #**interface g0/0**
Router (config-if) # **ip access-group** NO_ACCESS **out**
Per verificare
Router#**sh access-lists**

Standard IP access list NO_ACCESS
10 deny host 192.168.11.10
20 permit any
E' da notare che le linee digitate sono numerate di 10 in 10 quindi volendo editare l'access-list dovremo:
Per aggiungere
R1(config)#**ip access-list standard** NO_ACCESS

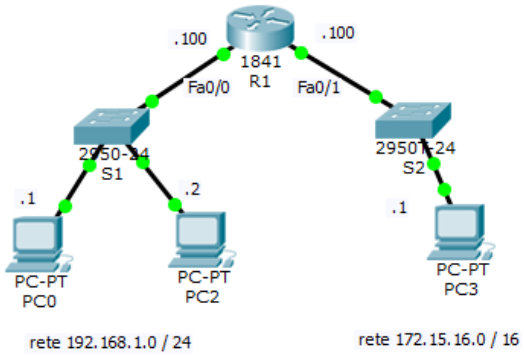
R1(config-std-nacl)#**15 deny any**
R1(config-std-nacl)#**exit**
R1(config)#**do sh access-lists**

Standard IP access list NO_ACCESS
10 deny host 192.168.11.10
15 deny any
20 permit any
La linea 15 è stata aggiunta nella giusta posizione
Per cancellare
R1(config)#**ip access-list standard** NO_ACCESS
R1(config-std-nacl)#**no 15**
R1(config-std-nacl)#**exit**
R1(config)#**do sh access-lists**
Standard IP access list NO_ACCESS
10 deny host 192.168.11.10
20 permit any
La linea 15 è stata cancellata.

Dall'esempio risultano evidenti i vantaggi nell'uso delle ACL con nome:

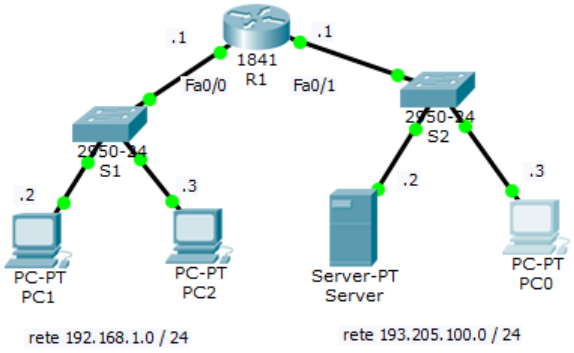
- possibilità di utilizzare caratteri alfanumerici per identificare l'ACL (più facili da ricordare);
- non vi è alcun limite al numero di named ACLs;
- possono essere modificate senza cancellare completamente l'ACL e riconfigurarla.

Esercizio 1



Realizzare una ACL che consenta il ping dall'host 192.168.1.1 verso tutti ma lo impedisca dalle altre reti verso la rete 192.168.1.0 / 24

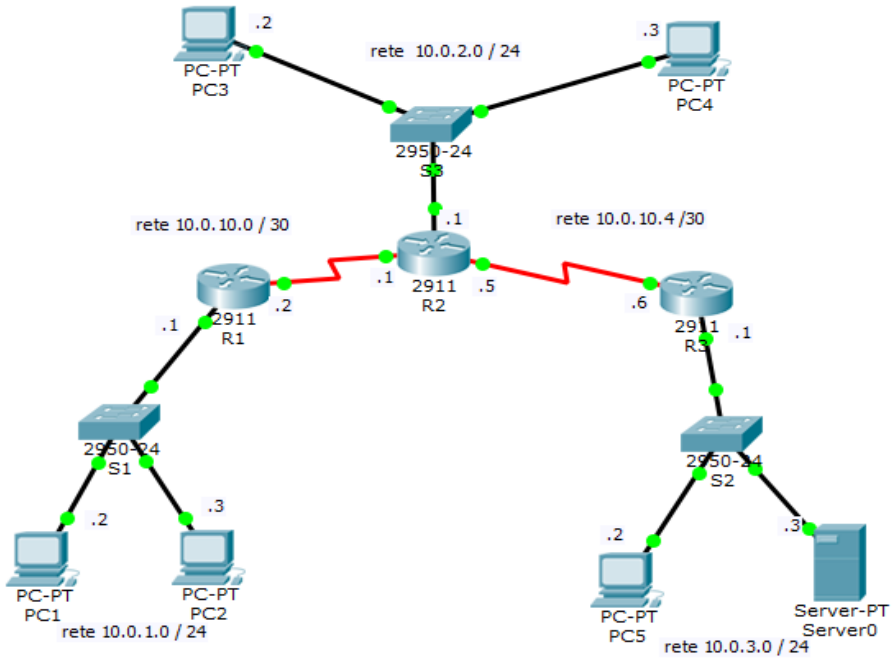
Esercizio 2



Realizzare su R1 una Access List che:

1) consenta ai soli pacchetti provenienti dal PC1 l' uscita verso la rete 193.205.100.0/24.
2) consenta a tutti l'accesso al server WEB sulla macchina PC 3.

Esercizio 3



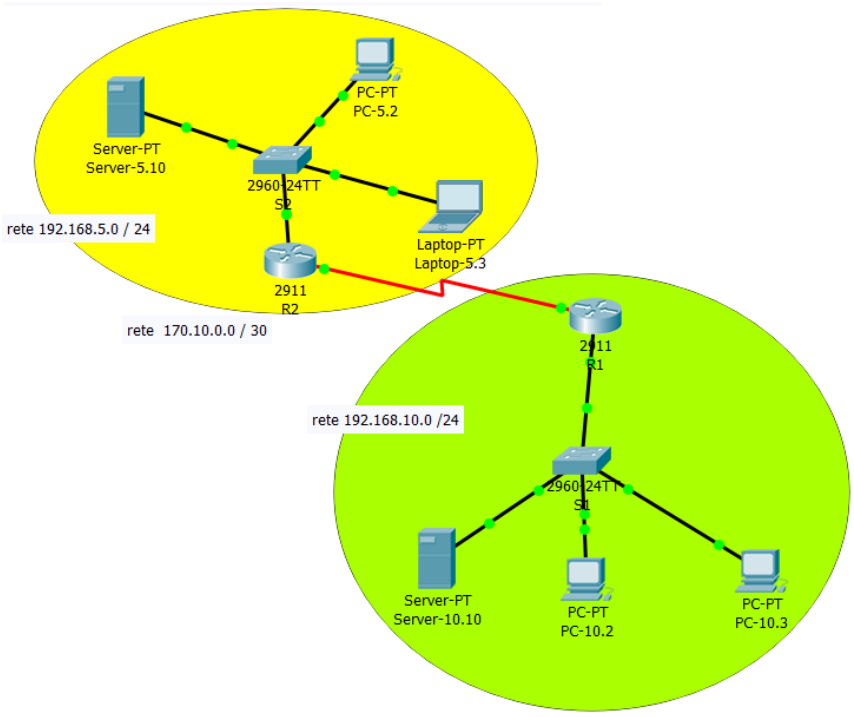
Realizzare una Access List che impedisca alla lan 10.0.1.0/24 di accedere alla lan 10.0.3.0/24

Esercizio 4

Utilizzando lo scenario dell' esercizio precedente realizzare una Access List che impedisca il traffico FTP dalla lan 10.0.1.0/24 alla lan10.0.3.0/24

Esercizio 5

Si vuole consentire il SOLO traffico http e https verso il server della rete 192.168.10.0/24



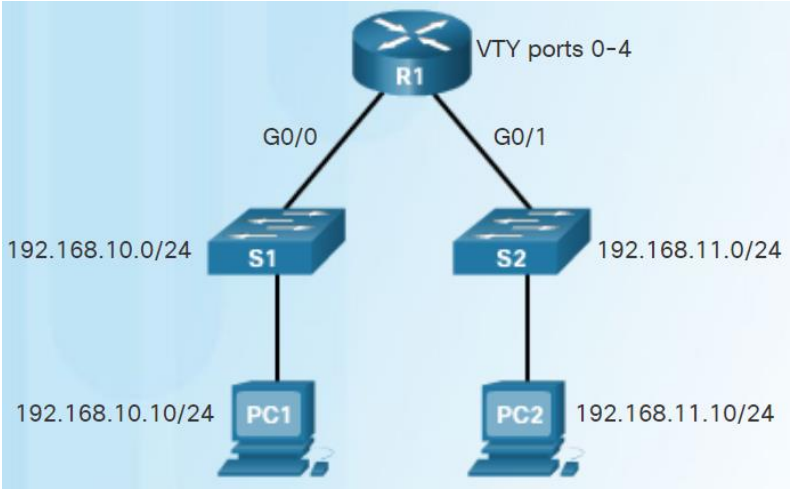
Sicurezza delle linee VTY

È possibile migliorare la sicurezza delle linee amministrative limitando l'accesso VTY. Questa tecnica consente di definire gli indirizzi IP consentiti per l'accesso in remoto al router. È possibile specificare a quali indirizzi IP è consentito l'accesso remoto al router con un ACL e un'istruzione di access-class configurata sulle linee VTY. Utilizzando questa tecnica in unione con l' SSH viene ad aumentare ulteriormente la sicurezza di accesso per amministare il device.
La sintassi del comando del comando access-class è:

router(config-line) # **access-class** access-list-number **{ in | out }**

in limita le connessioni in entrata tra gli indirizzi nell'access list e il dispositivo Cisco.
out limita le connessioni in uscita tra un particolare dispositivo Cisco e gli indirizzi nell'access list.

Un esempio che consente ad un intervallo di indirizzi di accedere alle linee VTY 0 - 4 è mostrato in figura . L'ACL è configurata per consentire alla rete 192.168.10.0 di accedere alle linee VTY 0 - 4 ma negare tutte le altre reti.



R1(config) # **line vty 0 4**
R1(config-line) # **login local**
R1(config-line) # **transport input ssh**
R1(config-line) # **access-class 21 in**
R1(config-line) # **exit**
R1(config) # **access-list 21 permit 192.168.19.0 0.0.0.255**
R1(config) # **access-list 21 permit 192.168.19.0 0.0.0.255 (non necessaria)**

Quando si configura una access-class è necessario considerare :
- per le access-class possono essere usate sia le ACL numerate che quelle named
- le restrizioni devono essere uguali per tutte le VTY.