Internet della Nuova Generazione: Protocolli e Prestazioni

1. Introduzione

1.1 Origini

Internet ha rivoluzionato il mondo dei computer e delle telecomunicazioni integrando le precedenti invenzioni del telegrafo, telefono, radio e computer in un unico mezzo.

Ci troviamo quindi di fronte ad uno strumento in grado di diffondere informazioni in tutto il mondo e un mezzo di collaborazione e interazione tra le persone e i loro computer.

L'evoluzione tecnologica di Internet affonda le sue radici nelle tecniche del packet switching e in ARPANET e in tutte le tecnologie ad essa collegate.

Nel 1962 J.C.R. Licklider allora direttore del progetto DARPA [ARPA (Advanced Research Projects Agency) poi DARPA (Defense Advanced Research Projects Agency) e i suoi collaboratori, si convinsero dell'importanza del concetto del networking.

Nel 1964 Leonard Kleinrock, che nel frattempo aveva pubblicato un lavoro sulla teoria del packet switching, convinse il ricercatore del MIT (Massachusetts Institute of Technology) Lawrence G. Roberts della flessibilità delle comunicazioni tramite packet switching.

Un altro punto chiave fu la prima connessione tra due computer avvenuta nel 1965 tra un TX-2 computer in Mass.E il Q-32 in California con una linea telefonica a bassa velocità, realizzando in effetti la prima WAN (Wide-area Network) mai costruita.

Il risultato principale di questo esperimento fu il costatare che Computer possono lavorare insieme, eseguendo programmi e condividendo dati su macchine remote, ma che il sistema del circuito commutato su linea telefonica era totalmente inadeguato.

Si rafforza la necessita dell'utilizzo del Packet switching e oltre a Roberts se ne occupano Donald Davies e Roger Scantlebury della NPL e altri al RAND (Research ANd Development) giungendo all'aumento della banda proposta per ARPANET da 2.4 kbps a 50 kbps.

Nell'Agosto del 1968 fu stabilita dal DARPA la RFQ (Request For Comment) per lo sviluppo del componente chiave, il packet switches chiamato IMP's (Interface Message Processors).

La realizzazione dell IMP's si ha nel dicembre 1968 da Frank Heart al BBN (Bolt Beranek and Newman) in collaborazione con Bob Kahn, la topologia della rete fu realizzata in collaborazione con Howard Frank e il suo team al Network Analysis Corporation e il sistema di misura della rete da Kleinrock alla UCLA.

Tutto questo lavoro teorico si concretizza nel settembre del 1969 quando BBN installo' il primo IMP alla UCLA e il primo host computer fu connesso e subito dopo, nell'ambito del progetto di Dough Engelbart "Augmantation of Human Intellect" (che addirittura comprendeva un tentativo di Ipertesto) allo Standford Information Institute (SRI), fu allestito un secondo nodo.

Un mese dopo fu inviato il primo messaggio host-to-host dal laboratorio di Kleinrock al SRI e furono aggiunti altri due nodi alla UC Santa Barbara e all'università dello Utah.

Siamo all'alba di INTERNET.

Negli anni successivi furono aggiunti numerosi computer e la ricerca si e' concentrata nello sviluppo di un protocollo Host-to-Host più funzionale e nella realizzazione di un adeguato software di rete.
Nel dicembre del 1970 il Network Working Group (NWG) sotto la supervisione di S. Crocker completo' il protocollo iniziale Host-to-Host di ARPANET,Network Control Protocol (NCP), permettendo agli utenti di iniziare lo sviluppo dei primi programmi applicativi.
Nell'Ottobre del 1972 Kahn organizza la prima grande dimostrazione pubblica di ARPANET durante l'International Computer Comunication Conference (ICCC).
Sempre nel 72, grazie al lavoro di Ray Tomlinson che scrisse il primo software in grado di leggere e scrivere messaggi, fa la sua comparsa un applicazione fondamentale, la posta elettronica.
Subito dopo Roberts aggiunse le funzioni di archiviazione, lista, lettura selettiva, inoltro e risposta ai messaggi realizzando un sistema di comunicazione che, da semplice meccanismo di coordinamento tra gli sviluppatori di ARPANET, si e' nel tempo trasformato nel World Wide Web che conosciamo oggi.

1.2 Le basi di Internet: dal packet radio al TCP/IP

Internet fu' basata sull'idea fondamentale che ci potevano essere reti multiple indipendenti con diverse strutture e a partire da ARPANET come rete iniziale basata sul packet switching si sarebbero presto incluse reti packet satellite e sistemi terrestri packet radio.

Il concetto base e' quello dell'architettura di rete aperta che interfaccia i vari nodi attraverso un meta-level "Internetworking Architecture".
Fino a questo punto c'era solo un modo per collegare tra loro i computer, il metodo tradizionale circuit switching dove le reti possono interconnettersi a livello di circuito passandosi bits a base sincrona su una porzione di circuito end-to-end.
In una rete ad architettura aperta le singole reti possono essere disegnate e sviluppate separatamente e avere interfacce per permettere la connessione ai singoli utenti o agli altri providers.
Questa idea fu introdotta da Kahn poco dopo il suo arrivo a DARPA nel 1972 partendo da un lavoro sul packet radio e arrivando alla realizzazione di un programma chiamato "Internetting". Al centro di tutto il lavoro era la ricerca di un protocollo affidabile che poteva mantenere l'effettiva comunicazione a dispetto di interferenze Jamming e di altro tipo o resistere a blackout intermittenti causati da una galleria o un ostacolo del terreno. Il NCP seguiva questa strada ma non era in grado di indirizzare reti (e macchine) a valle di un IMP di destinazione su ARPANET e non avendo un controllo di errore end-end se si verificava la perdita di un pacchetto il protocollo e di conseguenza le applicazioni che lo supportavano si bloccavano.
Riteniamo a questo punto necessario fornire una rapida visione delle caratteristiche salienti del protocollo X.25, dal quale nasce il protocollo AX.25, che viene utilizzato comunemente nelle trasmissioni Radioamatoriali in Packet.

1.2.1 Packet Radio

Il protocollo X.25 prevede, nel suo standard, tre differenti livelli di comunicazione: il livello fisico, il livello dei frames ed il livello dei pacchetti. Il

livello fisico si occupa del modo in cui i bit 0 e 1 sono rappresentati, come si stabilisce il contatto con la rete, i tempi di attesa e così via. Il livello dei frames si occupa della realizzazione della connessione tra due computer anche se questi sono connessi da una linea telefonica disturbata. Il livello dei pacchetti si occupa di gestire il formato ed il significato dei campi-dati contenuti nei singoli frames.
Il protocollo X.25, del quale ci occuperemo in questa sezione può essere definito come un protocollo “bit-oriented”, in contrapposizione con quelli che vengono definiti “character-oriented” e che hanno avuto un ruolo determinante agli albori del concetto di connessione in rete. Nei protocolli character-oriented, un frame, ovvero l'unità minima delle trasmissioni a pacchetto, è costituito da un numero intero di caratteri, scritti nel codice prescelto.
Il protocollo X.25, invece, è un protocollo “bit-oriented”, chiamato così in quanto i frames possono essere lunghi un numero arbitrario di bits. La dimensione del frame non deve essere necessariamente un multiplo intero di nessun carattere (in bit ovviamente).Ovviamente questo porta alla necessità di concepire un nuovo metodo per delimitare i frames. Ogni frames inizia e finisce con la stessa sequenza: 01111110, e questo tipo di successione (Flag1) non si potrà mai incontrare in nessuna altra parte del frame, in quanto il protocollo, quando incontra cinque bits 1 consecutivi, automaticamente inserisce un bit 0 al sesto posto (tecnica detta di bit stuffing). In modo del tutto analogo, quando il ricevitore incontra cinque bits 1 e poi uno 0, restituisce al bit 6 il valore di 1.
Così possiamo dire che tutti i protocolli "bit-oriented" utilizzano una struttura del frame come la seguente:

Figura 1.1 Frame del protocollo X25

dove, naturalmente, i campi flag1 e flag2 saranno proprio 01111110 .

Vediamo ora di chiarire il significato dei vari campi che compaiono nella figura.
ADDR: ovviamente è il campo address, di vitale importanza in quanto permette di identificare sia l'indirizzo di partenza, che quello di destinazione. Presenta una lunghezza di almeno un byte.
Control: può contenere tipi di informazioni diverse, a seconda del tipo di frame che è stato spedito e comunque presenta una lunghezza di almeno due bytes.
Data: può contenere delle informazioni del tutto arbitrarie e può avere una lunghezza qualsiasi, anche se l'efficienza della trasmissione può diminuire con l'aumentare della lunghezza di questo campo. Da notare il fatto che in packet la sua lunghezza massima è di 256 bytes.
FCS: ovvero "frame check sequence". E' il campo di sicuro più importante in quanto rappresenta una sorta di somma di tutti i bits contenuti nel frame dopo flag1 e viene generato dalla stazione trasmittente. Ovviamente anche la stazione ricevente esegue lo stesso controllo e se i dati coincidono il pacchetto viene accettato e confermato.
Flag2: niente altro che l'analogo di flag1 .
Come abbiamo già accennato parlando dei dati contenuti nel campo control, ci possono essere tre tipo di frames: Informazioni, Supervisione e Senza Numero (Information, Supervisory, Unnumbered).Nel primo caso il campo conterrà il numero di sequenza del frame ed altre informazioni utili per l'inoltro dei dati verso l'altra stazione (bits di poll/final). Nel caso di frame supervisore abbiamo di nuovo un'ulteriore suddivisione in frame di tipo "Receive Ready" (indica il successivo frame atteso), di tipo "Reject" (indica che si è verificato un errore di trasmissione), di tipo "Receive not Ready" (conferma tutti i frames a meno di quello indicato) , ed infine di tipo "Selective Reject" (chiede la ritrasmissione per un solo frame specificato).L'ultima classe di frames è quella dei frames Non Numerati, che sono utilizzati per scopi di controllo. Ovviamente anche i frames di controllo debbono essere confermati e per questo vi è uno speciale frame di controllo detto UA (Unnambered Acknowledgement-Conferma senza numero).

Tutti i protocolli "bit-oriented", quindi anche il X.25, prevedono un comando di disconnessione, che annuncia che il collegamento sta cessando, e molti altri ancora tra i quali si può citare il comando FRMR (Frame Reject), che indica l'avvenuta ricezione di un frame corretto con semantica impossibile.

1.2.1.1 Risorse attuali

Attualmente non è un problema, per chi voglia avventurarsi nelle trasmissioni digitali, comperare o autocostruire un TNC che possa funzionare a 1200 baud. Iniziano ad essere comuni anche delle velocità molto superiori, come 9600 baud o anche 38400 baud, velocità questa, utilizzata in prevalenza per il "forwarding" (ossia lo scambio di messaggi) tra diversi bbs (Bulletin board System). Tuttavia anche in Italia sono state realizzate delle Reti che utilizzano delle velocità superiori e che utilizzano in gran parte apparecchi e software autocostruiti. Un esempio per tutte può essere la rete ITANET, le cui tratte più veloci lavorano a 1.2 Mbit/s.

Purtroppo, nessun collegamento offre una affidabilità totale; dalla presenza di elementi di disturbo nel link ne consegue una perdita di pacchetti, ripetizioni (almeno nei protocolli che lo prevedono...) e sovrapposizioni di trasmissione. Si tratta quindi di studiare un protocollo che si occupi della rivelazione di errori nel trasferimento e ne determini la correzione. La tecnica più utilizzata nel ristabilire il link è quella della attesa, che consiste nello aspettare, per un tempo prestabilito, affinché sia il trasmettitore che il ricevitore di possano resincronizzare. Uno dei timer che maggiormente rallentano lo scambio dei dati è il cosiddetto RTD (Round Trip Delay). E' definito come il tempo medio che impiega un pacchetto di dati a percorrere il tratto tra i due computer in "andata e ritorno". Data la vulnerabilità del link, soprattutto nei sistemi radio, tale tempo

ha una variabilità del tutto casuale e comporta in breve tempo la perdita di sincronia tra i due computer. Un protocollo efficiente deve tuttavia prevedere uno Status Of Out Of Syncronization (SOS) tra le due parti. Lo SOS delimita la quantità dei dati che possono essere trasmessi prima che i due computer ritentino la sincronizzazione. Ogni protocollo ha dei valori di SOS (es. il maxframe del AX.25 o il windows del TCP/IP) che vengono determinati in modo da rendere affidabile o veloce il link.

1.2.2 Gli standard ISO/OSI, IEEE802.X

Per la realizzazione di una rete mondiale multiprotocollo il lavoro è enorme richiede un coordinamento a livello nazionale e internazionale e enti preposti a questo compito che sono:

PTT *(Post, Telegraph & Telephone)* è l'amministrazione che gestisce in una nazione i servizi trasmissivi (in Italia il Ministero delle Poste e delle Telecomunicazioni);

- CCITT *(Comité Consultatif International de Telegraphie et Telephonie)* è l'organismo internazionale che emette le specifiche tecniche che devono essere adottate dalle PTT. È recentemente entrato a far parte dell'ITU (*International Telecommunication Union*).
- ISO *(International Standard Organization)* è il principale ente di standardizzazione internazionale che si occupa anche di reti di calcolatori.
- ANSI *(American National Standards Institute)* è il rappresentante USA nell'ISO.

- UNINFO è il rappresentante italiano nell'ISO per le tematiche di reti di calcolatori.
- IEEE *(Institute of Electrical and Electronics Engineers)* è l'organizzazione professionale mondiale degli ingegneri elettrici ed elettronici con gruppi di standardizzazione sulle reti di calcolatori.

1.2.2.1 OSI (Open Systems Interconnections)

Alla fine degli anni settanta la ISO introduce OSI (Open Systems Interconnections) un progetto di ampio respiro con lo scopo di creare un modello di riferimento per le reti di calcolatori con un approccio a livelli (layers):
l'intero problema della comunicazione tra due applicazioni è stato spezzato in un insieme di sette livelli, ciascuno dei quali esegue funzioni ben specifiche. Nel processo di standardizzazione, OSI è partito dai livelli bassi (quelli più vicini all'hardware) ed è salito verso quelli alti (quelli più vicini all'uomo) ricevendo gradimento ed accettazione differenti. I livelli 1 (Fisico) e 2 (Data Link) di OSI sono oggi assolutamente standard e questo consente l'interoperabilità dei prodotti. Dal livello 3 al livello 7 i protocolli esistono da tempo, ma hanno difficoltà ad imporsi per l'alto impatto che la loro adozione ha sul software dei sistemi informativi stessi e sui dispositivi di instradamento (router). Solo la Digital Equipment Corp. ha deciso di abbandonare la propria architettura di rete proprietaria (DECnet fase IV) a favore di un'architettura totalmente OSI (DECnet fase V).

Per quanto riguarda i livelli 1 e 2 limitatamente alle reti locali e metropolitane è stato avviato, anch'esso tra la fine degli anni '70 e l'inizio degli anni '80, il

progetto IEEE 802 che ha portato ad una voluminosa serie di standard noti con sigle del tipo 802.X, oggi anche approvati dall'ISO. IEEE 802 è nato per razionalizzare i numerosi sforzi presenti in quegli anni per la creazione di nuove reti locali, spesso appositamente concepite - per ragioni commerciali - per essere incompatibili una con l'altra, ed ha ottenuto un notevole successo.
Per ridurre la complessità progettuale, OSI introduce *un'architettura a livelli (layered architecture)* i cui componenti principali sono:

- i livelli (*layers*);
- le entità (*entities*);
- i punti di accesso al servizio (SAP: *Service Access Points*);
- le connessioni (*connections*).

In una tale architettura, ciascun sistema è decomposto in un insieme ordinato di livelli, rappresentati per convenienza come una pila verticale. In figura sono rappresentati i livelli che compongono il modello di riferimento ISO-OSI.

Figura 1.2 Livelli del modello di riferimento ISO-OSI

Livelli adiacenti comunicano tramite la loro *interfaccia* (*interface*). Ogni livello è poi composto da una o più *entità*. Entità appartenenti allo stesso livello, su sistemi diversi, vengono dette *peer-entities*.

La trasmissione dei dati avviene quindi attraverso una serie di passaggi da livelli superiori a livelli inferiori in un primo sistema, quindi attraverso mezzi fisici di comunicazione, e poi attraverso un'altra serie di passaggi, questa volta da livelli inferiori a livelli superiori, in un secondo sistema

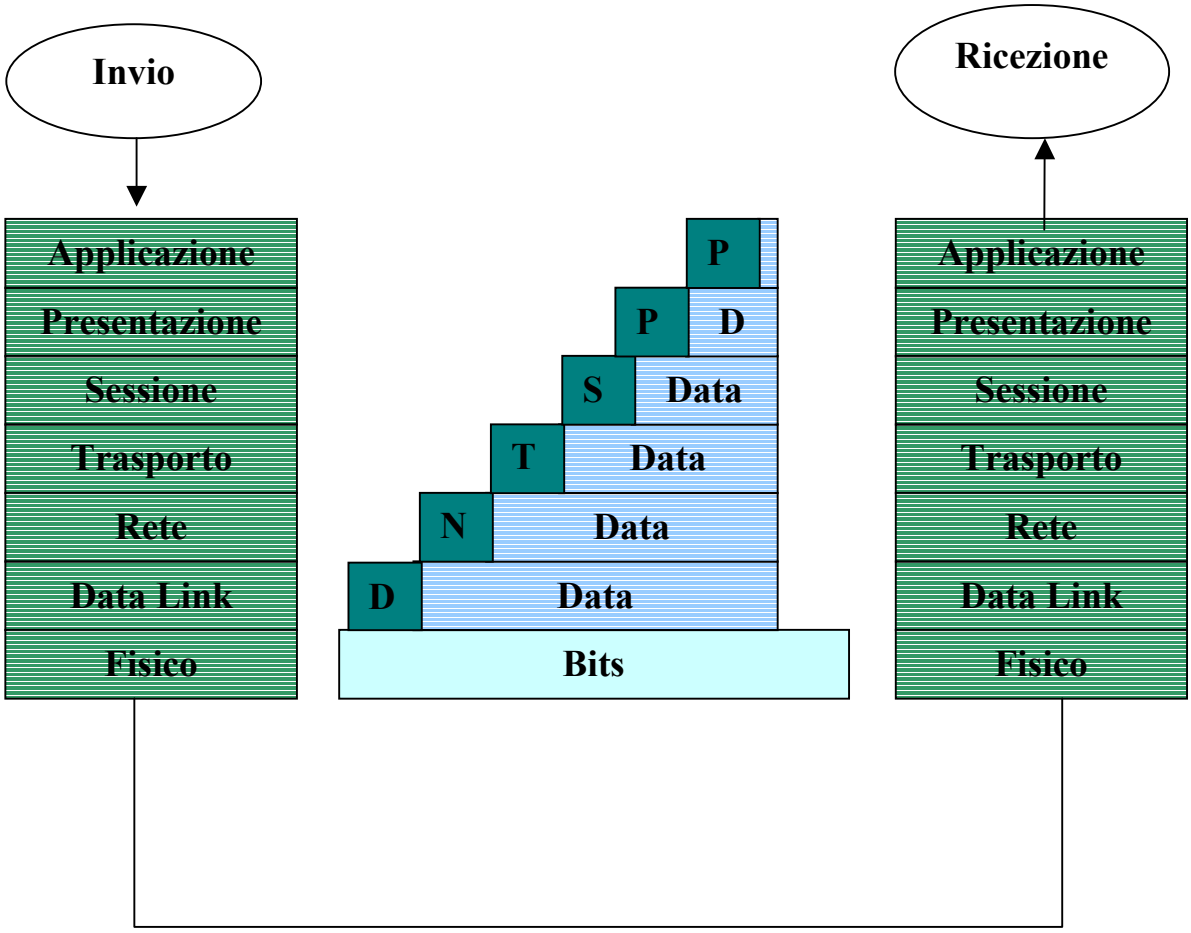

Figura 1.3 Trasmissione dati tra due sistemi

Vediamo ora di analizzare in dettaglio i vari livelli del modello ISO/OSI:

Livello 7 Applicazione

Il livello 7 è il livello *Applicazione*, cioè dei programmi applicativi (facenti parte del sistema operativo o scritti dagli utenti) attraverso i quali l'utente finale utilizza la rete; esempi di tali applicativi sono: VT (*Terminale Virtuale*), cioè connessione interattiva ad un elaboratore remoto, FTAM (*File Transfer and Access Management*), X.400 (la posta elettronica) e X.500 (*Directory Service*).

Livello 6 Presentazione

Il livello 6 è il livello *Presentazione*, che gestisce la sintassi dell'informazione da trasferire (ad esempio codifica ASCII o EBCDIC); a questo livello sono previste tre diverse sintassi: astratta (definizione formale dei dati che gli applicativi si scambiano, come in ISO 8824 o in ASN.1), concreta locale (come i dati sono rappresentati localmente) e di trasferimento (come i dati sono codificati durante il trasferimento).

Livello 5 Sessione

Il livello 5 è il livello *Sessione*, responsabile dell'organizzazione del dialogo tra due programmi applicativi e del conseguente scambio di dati; esso consente di aggiungere a connessioni end-to-end, cioè tra due entità collocate in ES (End System) servizi più avanzati, quali la gestione del dialogo (mono o bidirezionale), la gestione del token (per effettuare mutua esclusione nell'utilizzo di una risorsa condivisa) o la sincronizzazione (inserendo dei checkpoint in modo da ridurre la quantità di dati da ritrasmettere in caso di gravi malfunzionamenti).

Livello 4 Trasporto

Il livello 4 è il livello *Trasporto*, e fornisce trasferimento trasparente di informazione tra entità del livello sessione. In particolare, si occupa di fornire un trasferimento dati affidabile e di ottimizzare l'uso delle risorse di rete. Compiti del livello 4 saranno quindi tipicamente la frammentazione, la correzione degli errori e la prevenzione della congestione della rete. Il livello 4 è il più basso livello a trascurare la topologia della rete e la presenza di sistemi intermedi (IS) e quindi è il primo livello detto *end-to-end.*

Livello 3 Network

Il livello 3 è il livello *Network*, che gestisce l'instradamento dei messaggi; esso determina se e quali sistemi intermedi devono essere attraversati dal messaggio per giungere a destinazione, quindi deve gestire delle tabelle di instradamento e provvedere ad instradamenti alternativi in caso di guasti (fault tolerance).

Livello 2 Data Link

Il livello 2 è il livello *Data Link*, che ha come scopo la trasmissione sufficientemente affidabile di trame (*frame*); accetta come input dei pacchetti di livello 3 (tipicamente poche centinaia di bit) e li trasmette sequenzialmente. Esso verifica la presenza di errori aggiungendo delle FCS (*Frame Control Sequence*) e può gestire meccanismi di correzione di tali errori tramite ritrasmissione.

Livello 1 Fisico

Il livello 1 del modello OSI è il livello *Fisico*, che si occupa di trasmettere sequenze binarie sul canale di comunicazione; a questo livello si specificano, ad esempio, le tensioni che rappresentano 0 e 1 e le caratteristiche dei cavi e dei connettori.

Per tutti i livelli superiori al livello fisico sono definite due modalità operative: una modalità *connessa (CONS: Connection Oriented Network Service)* e una modalità non connessa *(CLNS: ConnectionLess Network Service)*. 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. Lo standard originale ISO 7498 prevedeva solo la modalità connessa ma, vista l'importanza della modalità non connessa, è stata aggiunta in seguito come emendamento allo standard stesso (ISO 7498/Addendum 1).
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 (i saluti finali e il posare il microtelefono).

Modalità connessa

Nella modalità connessa lo scambio di dati avviene tramite le tre fasi viste prima. Durante la fase di creazione della connessione (*initial setup*) due peer-entities concordano che trasferiranno delle PDU (Protocol Data Unit). Solo durante tale fase devono essere specificati gli indirizzi completi del mittente e del destinatario: successivamente le entità coinvolte specificheranno soltanto l'identificativo della connessione stabilito durante la prima fase. Un servizio connesso fornisce una modalità di trasferimento delle PDU affidabile e sequenziale. Per tutta la durata della connessione le PDU inviate sono ricevute correttamente nello stesso ordine. Se qualcosa non funziona correttamente, la connessione può essere riavviata (*reset*) o terminata (*released*). Per verificare che tutte le PDU inviate giungano a destinazione correttamente un servizio connesso utilizza degli schemi di numerazione dei pacchetti e di verifica dell'avvenuta corretta ricezione (ACK: acknowledgement). Quindi un protocollo connesso è in generale in grado non solo di rilevare la presenza di errori, ma anche di correggerli tramite ritrasmissioni.

Modalità non connessa

Con una modalità non connessa la comunicazione ha luogo in una fase singola: il pacchetto è inviato e deve contenere l'indirizzo completo del destinatario. Non essendo i pacchetti organizzati in una connessione, un pacchetto non può fare riferimento ad altri pacchetti trasmessi precedentemente o in seguito. Quindi un protocollo non connesso può solo rilevare la presenza di errori (scartando quindi le PDU errate), ma non correggerli in quanto non si possono realizzare meccanismi di ritrasmissione (in un pacchetto non è possibile fare riferimento ad altri pacchetti). Un protocollo non connesso è in generale più efficiente di un protocollo connesso, specialmente se bisogna trasferire piccole quantità di dati:

in quest'ultimo caso infatti l'overhead della creazione e distruzione della connessione è rilevante.
Un protocollo non connesso (detto anche *datagram*), non potendo garantire l'affidabilità del trasferimento dati, necessita che almeno un protocollo di livello superiore sia di tipo connesso.

1.2.2.2 Il progetto IEEE 802

Sempre nello stesso periodo (fine anni settanta) fanno la comparsa sul mercato le LAN (Local Area Network).
Una LAN è un sistema di comunicazione che permette ad apparecchiature indipendenti di comunicare tra di loro, entro un'area delimitata, utilizzando un canale fisico a velocità elevata e con basso tasso d'errore.
Fino ad ora il concetto dominante era quello master-slave
Il mainframe, o in generale l'elaboratore centrale, era il master della comunicazione, e i terminali, o le stazioni, gli slave. Tutti i sistemi connessi alla LAN diventano invece paritetici, cioè della stessa importanza.

L'idea di usare un solo canale fisico di trasmissione per realizzare una LAN può a prima vista sembrare restrittiva, ma così non è. Quando le LAN fecero la loro comparsa sul mercato, spinte dai costruttori di calcolatori, i costruttori di sistemi di telecomunicazione cercarono di ostacolarle, proponendo come alternativa i PBX (Private Branch eXchange) digitali. Questi sono dei centralini privati numerici in grado di commutare un grande numero di circuiti digitali a 64 Kbit/s (velocità standard per un canale telefonico numerico). Fu un clamoroso fallimento e la diffusione delle LAN crebbe sempre più velocemente. La causa di tale fallimento è da ricercarsi nella modalità operativa dell'utente di LAN.

Egli infatti, contrariamente a quanto si potrebbe pensare, per la maggior parte del tempo non utilizza la rete. Quando però la utilizza, chiede alla rete di avere prestazioni altissime. Tale modalità di utilizzo viene detta "a burst". Questo mal si accorda con il modello del PBX numerico che alloca permanentemente a ciascun utente 64 Kbit/s che sono inutilizzati per la maggior parte del tempo e di prestazioni troppo limitate quando l'utente decide di utilizzare la rete.

La trasmissione è sempre di tipo broadcast: un sistema trasmette e tutti gli altri ricevono. Tale organizzazione ha enormi vantaggi, ma impone anche alcune complicazioni: è necessaria la presenza di indirizzi per stabilire chi sono il reale destinatario e il mittente della trasmissione e occorre arbitrare l'accesso all'unico mezzo trasmissivo tra tutti i sistemi che hanno necessità di trasmettere.
Questo canale e' inoltre a basso tasso di errore permettendo quindi l'utilizzo di protocolli OSI di livello 2 connectionless ad alte prestazioni.
Gli attributi che deve possedere una LAN sono quelli classici delle reti di calcolatori e cioè:

- *affidabilità*: oggi la tecnologia delle LAN è assolutamente consolidata e consente di ottenere affidabilità elevatissime, tali da permettere a molti costruttori di produrre schede di rete locale con garanzia illimitata;
- *flessibilità*: oggi le LAN sono utilizzate per applicazioni molto disparate, dalle LAN di soli PC all'integrazione PC-mainframe, fungendo da supporto unificato per più architetture di rete, tra loro incompatibili ai livelli superiori del modello OSI;
- *modularità*: le LAN possono essere realizzate utilizzando componenti di molti costruttori diversi, perfettamente intercambiabili;
- *espansibilità*: le LAN sono strutture appositamente concepite per fornire una crescita graduale nel tempo, secondo le esigenze dell'utente;

- *gestibilità:* la maggior parte dei componenti delle LAN prodotti negli ultimi anni sono concepiti per essere gestiti mediante accessi remoti utilizzando il protocollo SNMP (*Simple Network Management Protocol*), che è un protocollo applicativo basato su UDP/IP

Quando le prime LAN cominciarono a diffondersi (ARC, Ethernet, Token Ring, ecc.), l'IEEE decise di costituire sei comitati per studiare il problema della standardizzazione

delle LAN e delle MAN (Metropolitan Area Network), complessivamente raccolti nel progetto IEEE 802. Tali comitati sono:

- 802.1 Overview, Architecture, Bridging and Management;
- 802.2 Logical Link Control;
- 802.3 CSMA/CD (*Carrier Sense, Multiple Access with Collision Detection*);
- 802.4 Token Bus;
- 802.5 Token Ring;
- 802.6 Metropolitan Area Networks - DQDB (Distributed Queue, Dual Bus).

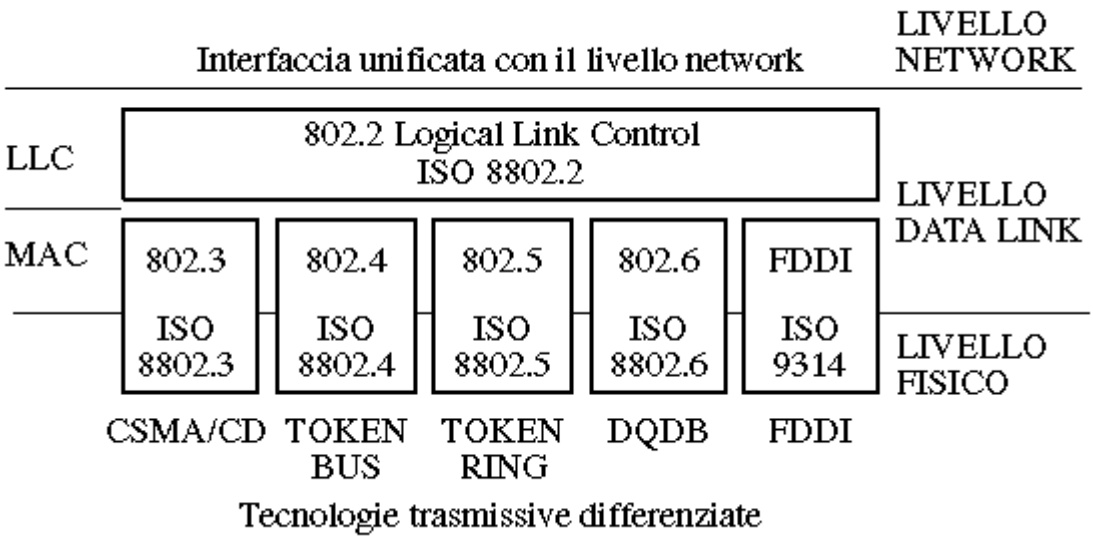

Figura 1.4 Tecnologie trasmissive differenziate

A tali comitati in seguito se ne sono aggiunti altri tra cui:

- 802.3u 100BaseT;
- 802.7 Broadband technical advisory group;
- 802.8 Fiber-optic technical advisory group;
- 802.9 Integrated data and voice networks;
- 802.10 Network security;
- 802.11 Wireless network;
- 802.12 100VG AnyLAN;
- 802.14 Cable-TV based broadband communication network.

IEEE 802 introduce l'idea che le LAN e le MAN devono fornire un'interfaccia unificata verso il livello Network (livello rete), pur utilizzando tecnologie trasmissive
differenziate. Per ottenere tale risultato, il progetto IEEE 802 suddivide il livello Data Link in due sottolivelli:

- LLC (*Logical Link Control*);
- MAC (*Media Access Control*).

Il sottolivello LLC è comune a tutte le LAN, mentre il MAC è peculiare di ciascuna LAN, così come il livello fisico al quale è strettamente associato.

Il sottolivello MAC è specifico di ogni LAN e risolve il problema della condivisione del mezzo trasmissivo. Esistono vari tipi di MAC, basati su principi diversi, quali la contesa, il token, la prenotazione e il round-robin. Il MAC è indispensabile in quanto a livello 2 (Data Link) le LAN implementano sempre una sottorete trasmissiva di tipo broadcast in cui ogni sistema riceve tutti

i frame inviati dagli altri. Trasmettere in broadcast, cioè far condividere un unico canale trasmissivo a tutti i sistemi, implica la soluzione di due problemi:

- in trasmissione, verificare che il canale sia libero prima di trasmettere e risolvere eventuali conflitti di più sistemi che vogliano utilizzare contemporaneamente il canale;
- in ricezione, determinare a quali sistemi è effettivamente destinato il messaggio e quale sistema lo ha generato.

La soluzione del secondo problema implica la presenza di indirizzi a livello MAC (quindi nella MAC-PDU) che trasformino trasmissioni broadcast in:

- trasmissioni punto-punto, se l'indirizzo di destinazione indica un singolo sistema;
- trasmissioni punto-gruppo, se l'indirizzo di destinazione indica un gruppo di sistemi;
- trasmissioni effettivamente broadcast, se l'indirizzo di destinazione indica tutti i sistemi.

Il MAC deve anche tenere conto della topologia della LAN, che implica leggere variazioni sulle possibili modalità di realizzazione del broadcast: con topologie a bus, è un broadcast a livello fisico (elettrico), mentre con topologie utilizzanti canali punto-punto, quali l'anello, è un broadcast di tipo logico.

Le reti locali hanno canali sufficientemente affidabili, quindi non è in genere necessario effettuare correzione degli errori. Se ciò fosse richiesto, sarebbe il sottolivello LLC ad occuparsene essendo il MAC sempre connectionless.

LLC ha lo scopo di fornire un'interfaccia unificata con il livello network, il più simile possibile a quella delle reti geografiche. Per queste ultime l'OSI ha accettato come standard i protocolli della famiglia HDLC (High-level Data Link Control) e quindi LLC è stato progettato come una variante di HDLC per le reti locali. La differenza principale tra LLC e HDLC è che, mentre HDLC si appoggia direttamente sul livello fisico e quindi deve occuparsi della

delimitazione delle trame e della trasparenza del campo dati, LLC si appoggia sul livello MAC cui viene demandata la soluzione di tali problemi.

Vediamo ora come sono costituite le intestazioni di Ethernet;
Al momento della creazione dello standard si voleva essere sicuri che ogni macchina avesse un differente indirizzo ethernet e che l'utente finale non avrebbe dovuto preoccuparsi di questo.
E' per questo motivo che ogni controller viene costruito con un suo personale indirizzo costituito da 48 bit che viene registrato dalle case produttrici presso un'autorità centrale.
Ethernet e' un mezzo che funziona in modalità "broadcast", quando si invia un pacchetto sulla rete ogni macchina lo vede e quindi sono necessarie delle procedure in grado di assicurare il recapito alla giusta macchina.
Ogni pacchetto ha quindi un'intestazione di 14-ottetti che include il source and destination ethernet address e un type code e tutte le macchine collegate presteranno attenzione solamente ai pacchetti che hanno il loro indirizzo nel campo destinazione.
Il type code permette a differenti famiglie di protocolli di essere usate nella stessa rete inserendo un diverso valore in questo campo.
Infine il checksum, posto alla fine:
viene calcolato dal controller che invia e da quello che riceve un certo pacchetto e nel caso siano differenti comporta l'eliminazione di detto pacchetto che viene considerato affetto da errori di trasmissione.

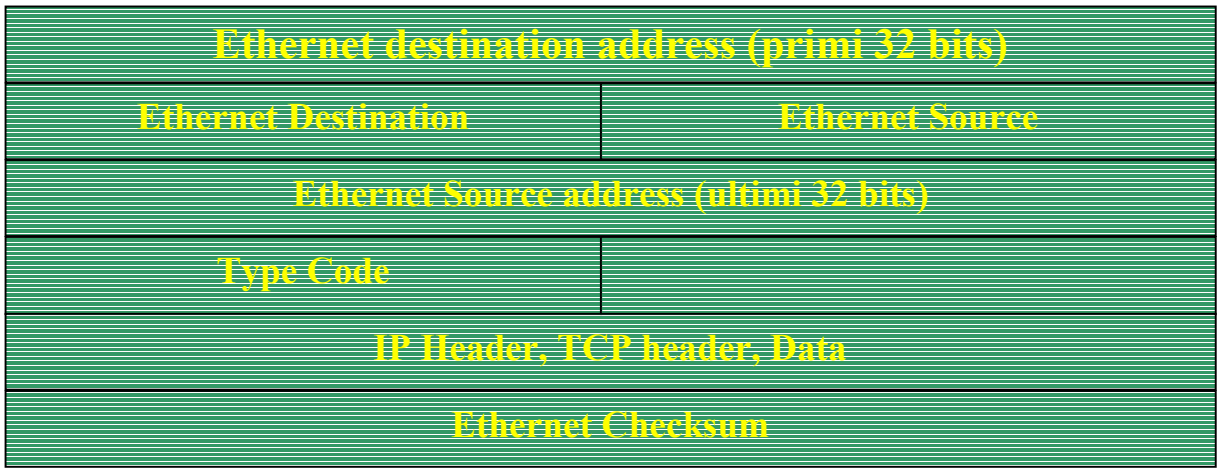

Figura 1.5 Formato di un pacchetto Ethernet

Quando questi pacchetti sono ricevuti naturalmente le intestazioni sono rimosse, si guarda poi al type code per passare il datagramma ai protocolli di livello superiore (in genere IP e TCP)

2 Il TCP/IP

Verso la fine degli anni '70, tale sforzo portò al completamento dell'*Internet Protocol Suite,* di cui i due protocolli più noti sono il TCP (*Transmission Control Protocol*) e l'IP (*Internet Protocol*).

Questi protocolli furono utilizzati da un gruppo di ricercatori per la rete ARPAnet e ottennero un elevato successo, anche perché posti sin dall'inizio nel dominio

pubblico e quindi utilizzabili gratuitamente da tutti. Il nome più accurato per l'architettura di rete rimane quello di Internet Protocol Suite, anche se comunemente si fa riferimento ad essa con la sigla TCP/IP o IP/TCP. Questo può portare ad alcune ambiguità: ad esempio è comune sentir parlare di NFS come un servizio basato su TCP/IP, anche se NFS (Network File System) non usa il protocollo TCP, ma un protocollo alternativo detto UDP (User Datagram Protocol) appartenete all'Internet Protocol Suite. Visto l'uso estremamente comune della sigla TCP/IP, essa verrà adottata anche in questo libro in luogo del termine più corretto, quando non crei confusione.

TCP/IP è l'architettura adottata dalla rete Internet che, con le sue decine di milioni di calcolatori e il suo tasso di crescita del 5% al mese, è la più grande rete di calcolatori al mondo.

I protocolli appartenenti a questa architettura sono specificati tramite standard che si chiamano RFC (*Request For Comments*) facilmente reperibili sulla rete Internet.

Famoso è lo RFC 791 Internet Protocol, redatto da Jon Postel e datato 1981, che specifica appunto il protocollo IP.

L'architettura TCP/IP ha dei componenti, quali l'IP, indubbiamente datati, ma non obsoleti: il grande successo di TCP/IP è quotidiano. Negli anni 1990, gli anni della maturità dell'ISO/OSI, l'unica architettura di rete che sembra interessare il mercato è quasi paradossalmente TCP/IP.

Anche gli enti di standardizzazione nazionali e internazionali hanno dovuto arrendersi davanti alla massiccia diffusione di TCP/IP e dargli la stessa dignità di ISO/OSI.

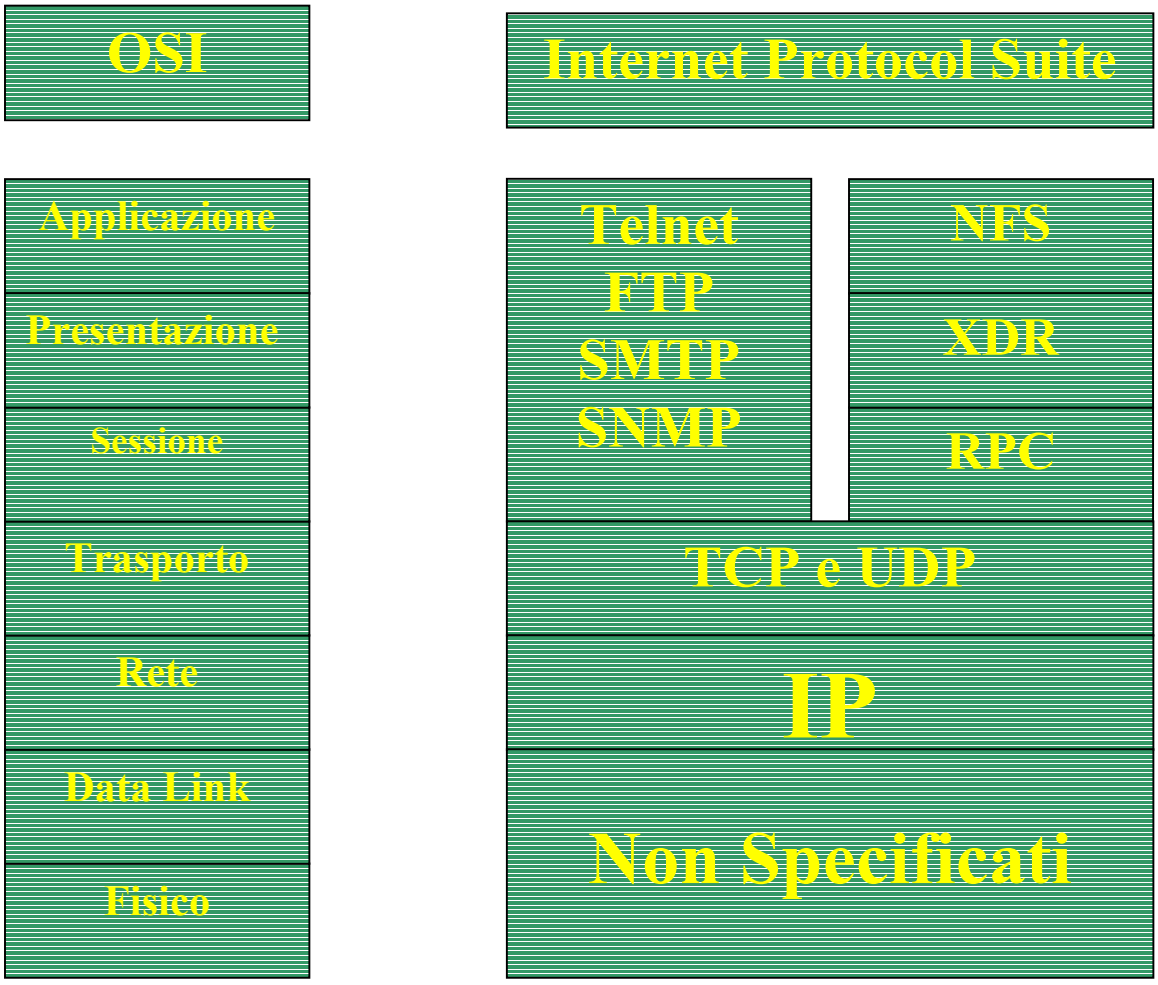

Figura 2.1 Relazione tra i livelli di TCP/IP e di OSI

L'architettura di rete TCP/IP non specifica i livelli 1 e 2 della rete, ma utilizza quelli normalmente disponibili e conformi agli standard. Ad esempio, nell'ambito delle reti locali opera su Ethernet/IEEE802.3, Token-Ring e FDDI; nell'ambito delle reti geografiche su HDLC, PPP, SLIP, X.25, Frame Relay, SMDS e ATM.

Esistono anche realizzazioni per reti molto strane, spesso diffuse solo all'interno di certe comunità, ad esempio AIX.25, una rete packet switched dei radioamatori.

Se un pacchetto non arrivava a destinazione veniva subito ritrasmesso dalla sorgente rendendo necessaria la costruzione di black box, poi chiamate gateway e routers, per svolgere le funzioni di interpretazione delle intestazioni degli IP.

Il progetto era quello di creare una rete che sfruttando le conoscenze acquisite con le tecniche del packet switching riuscisse a realizzare la connessione fra host con diversi sistemi operativi senza la necessita' di un controllo globale delle trasmissioni.

Era invece diventato indispensabile un sistema di indirizzamento globale e venne usato un IP a 32 bit in cui i primi 8 bits identificano la rete e i restanti 24 l'host su quella rete.

A questo punto il TCP era già in grado di fornire una serie di servizi di trasporto e inoltro necessari al funzionamento di Internet dalla consegna dei dati in successione (Virtual Circuit model) al Datagram service nel quale le applicazioni sono costruite appositamente per sfruttare la struttura della rete (prevedendo quindi occasionali pacchetti persi, corrotti o riordinati).

Anche se in grado di funzionare perfettamente nei circuiti virtuali per funzioni di trasferimento file e accesso remoto il TCP aveva difficoltà con le applicazioni di rete avanzate (in particolare packet voice) e quindi venne diviso in due protocolli separati.

Il semplice IP, che si occupa dell'indirizzamento e l'inoltro dei singoli pacchetti, e il separato TCP con caratteristiche di servizio come il controllo di flusso (flow control) e il recupero dei pacchetti persi.

Per quelle applicazioni che non usufruivano del TCP, fu' introdotto il User Datagram Protocol (UDP) per fornire l'accesso diretto ai servizi fondamentali dell'IP.

In entrambi i casi i dati da inviare vengono spezzettati in datagrammi e riassemblati una volta ricevuti, mentre ne viene chiesto il rinvio se si sono persi per strada. IP procede poi all'instradamento degli stessi. Da notare che ogni livello di protocollo ignora completamente l'altro; e di fatti ogni livello di protocollo aggiunge delle intestazioni ai datagrammi di 20 bytes (160 bit), cosicché questi sono univoci ed imperdibili.

Il TCP è un protocollo di transport di tipo connection-oriented che fornisce un servizio di tipo full-duplex (bidirezionale-contemporaneo), con acknowledge (conferma) e controllo di flusso. Il TCP è utilizzato dalle applicazioni di rete che richiedono una trasmissione affidabile dell'informazione. Le applicazioni si connettono alle porte TCP e ad alcune applicazioni principali sono associate delle *well know port* cioè delle porte che hanno lo stesso numero su tutti i calcolatori (ad esempio all'applicazione telnet è associata la porta 23).

Il TCP segmenta e riassembla i dati secondo le sue necessità: ad esempio se un'applicazione fa cinque scritture su una porta TCP, l'applicazione destinataria può dover effettuare 10 letture per ottenere tutti i dati, oppure ottenerli tutti in una sola lettura. Il TCP è un protocollo a sliding window (finestre) con meccanismi di time-out e ritrasmissione. La ricezione dei dati deve essere confermata dall'applicazione remota.

Si può quindi pensare a TCP come ad una libreria di routines che le applicazioni usano quando vogliono comunicare con un altro computer e che a sua volta richiama i servizi forniti da IP che raccoglie appunto una serie di servizi utilizzabili sia da TCP che da altre applicazioni che non usano TCP.

Questa strategia di costruire diversi livelli di protocolli e' detta "layering", si considerano applicazioni come mail, TCP e IP come livelli separati ognuno dei quali chiama quello sottostante.

Generalmente TCP/IP utilizza 4 livelli:

- Un livello applicazione (come ad esempio mail)
- Un protocollo come TCP che fornisce i servizi necessari a molte applicazioni
- IP che provvede al servizio fondamentale di recapito dei datagrammi a destinazione
- Il protocollo necessario a gestire il mezzo fisico (Ethernet o un collegamento punto-punto)

TCP/IP e' basato sul "catanet model" che consiste nel considerare un grande numero di reti indipendenti collegate tra loro per mezzo di gateways (attualmente chiamati routers).
L'utente e' in grado di accedere ai computer o a altre risorse di ognuna di queste reti e i datagrammi attraverseranno numerosi sistemi diversi prima di giungere a destinazione.
E' bene precisare che un indirizzo internet e' un numero di 32 bit normalmente scritto come la sequenza di 4 numeri decimali separati da un punto che rappresentano 8 bits.(Il termine ottetto e' usato in internet quando si parla di blocchi di 8 bit al posto di byte in quanto TCP/IP e' supportato anche da alcuni computer che hanno dimensioni diverse dei bytes)
Normalmente comunque ci riferiamo ai sistemi usando dei nomi piuttosto che gli indirizzi.
Questo è possibile in quanto un software controlla in un database la corrispondenza tra nomi e numeri.

TCP/IP quindi trasferisce le informazioni attraverso una sequenza di datagrammi.

Un datagramma e' un insieme di dati che sono inviati come un singolo messaggio.

Le informazioni sono spezzate in datagrammi e ognuno di questi e' inviato attraverso la rete individualmente.

Supponiamo di voler trasferire un file di 15000 ottetti, tenendo presente che molte reti non sono in grado di gestire datagrammi di tali dimensioni.

Così il protocollo lo spezza in , ad esempio, 30 datagrammi da 500 ottetti che verranno inviati e ricostruiti nel file originale.

Ovviamente, mentre i datagrammi sono in transito, la rete non sa che essi sono parte di un unico file e può tranquillamente succedere che il datagramma 53 arrivi prima del 52 o che alcuni vadano persi (vedremo in seguito come e' gestita la ritrasmissione dei datagrammi).

Il termini datagramma e pacchetto spesso sembrano essere la stessa cosa anche se tecnicamente datagramma e' la giusta parola da usare quando si parla di TCP/IP.

Un datagramma e' appunto un'unita di dati usata dal protocollo per negoziare informazioni mentre un pacchetto e' una quantità fisica utilizzata da Ethernet e da altre connessioni via cavo.

Generalmente un pacchetto semplicemente contiene un datagramma ma in alcuni casi come ad esempio quando TCP/IP è usato con reti X25 che spezzano il datagramma in pacchetti da 128 ottetti ne sono necessari diversi.

Questo e' invisibile al protocollo in quanto i pacchetti sono riassemblati nel datagramma originale prima di essere processati da TCP/IP.

2.1 TCP

Passiamo ora ad analizzare più in dettaglio come si comporta il livello TCP:

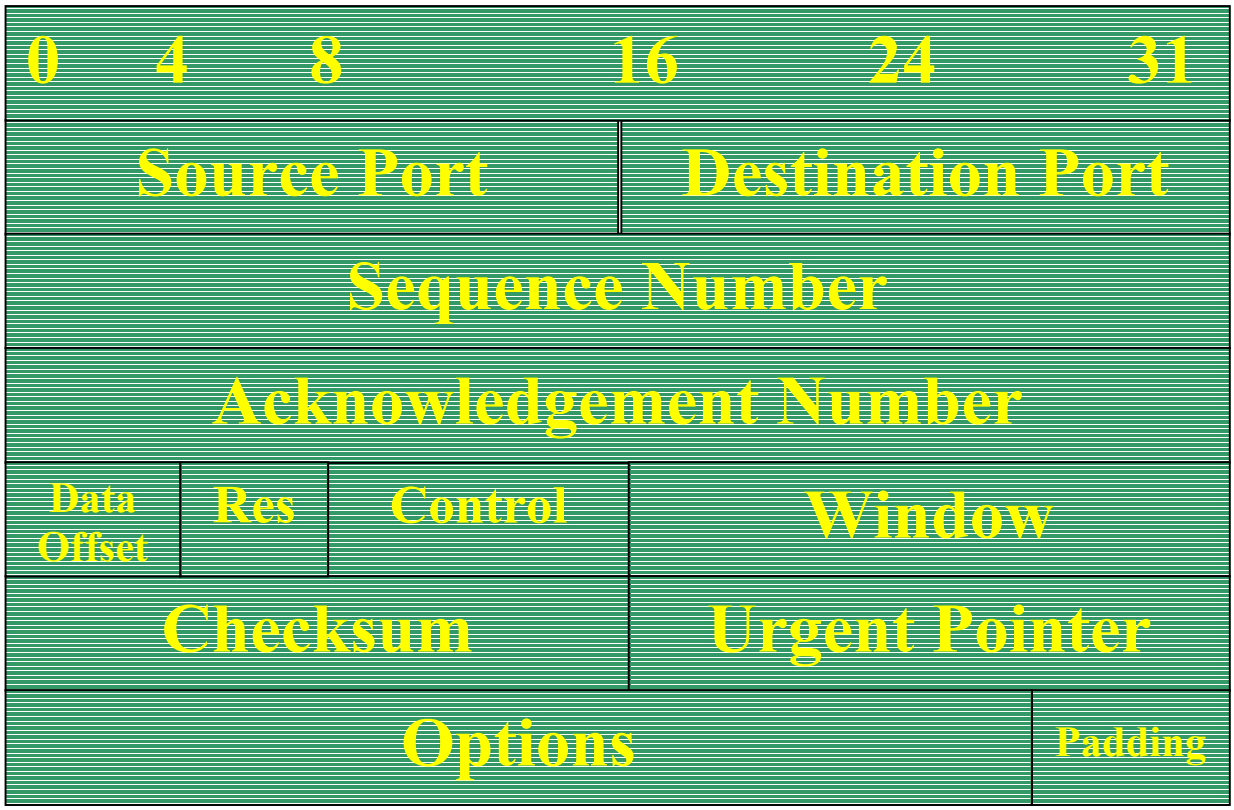

Figura 2.2 Formato di un pacchetto TCP

- **SOURCE PORT** e' l'interfaccia che si occupa del transito del file che viene inviato.
- **DESTINATION PORT** e', ovviamente, l'equivalente che viene raggiunto nell'altro computer. Il significato di queste due ripartizioni e' da ricercare nel

fatto che ogni sistema può essere provvisto di più porte e può cosi' connettere contemporaneamente più di un utente. Source Port e Destination Port si interscambiano quando l'utente contattato manda a sua volta indietro dei dati.

- **SEQUENCE NUMBER**, specifico per ogni datagramma, permette al corrispondente di assicurarsi della corretta ricostruzione e quindi ricezione perfetta del file. Come ci si accorge che il ricevente ha effettivamente ricevuto il datagramma e nella giusta sequenza? Il destinatario (target) invia al mittente (sender) un **ACKNOWLEDGEMENT NUMBER**, che andrà a formare il Sequence Number dei dati ricevuti dal corrispondente.
- **WINDOW** e' un campo che serve a controllare quanti dati possono essere inviati. Poiché il corrispondente deve avere il tempo di immagazzinare la serie di datagrammi in arrivo (che varia in dipendenza da molti fattori) rimanda di volta in volta al mittente, tramite il campo Window, quanti dati può ancora ricevere; per cui via via che vengono scambiati i dati, il campo varia continuamente. Questo campo risulta vitale per informare il mittente quando deve sospendere l'invio dei dati e quando riprendere la trasmissione.
- **CHECKSUM** presenta la somma di tutti i bytes contenuti nel datagramma che deve risultare identica nel datagramma di partenza e nel datagramma di arrivo. E' anche questa una informazione di controllo; infatti se durante il trasferimento dei dati qualcosa va storto i dati di checksum non saranno identici e verrà richiesto di ritrasmettere il datagramma errato.
- **URGENT** permette ad uno dei due sistemi di segnalare all'altro la necessita' di passare a trasmettere un determinato ottetto. Con tale sistema si forza, ad esempio, l'interruzione nella trasmissione.

Il Trasmission Control Protocol e’ responsabile della rottura del messaggio in datagrammi, della ricostruzione all’altro capo della rete, del rinvio di qualsiasi cosa sia andata persa e del riordino dei datagrammi mentre IP si occupa dell’instradamento dei singoli datagrammi.

L’interfaccia tra TCP e IP e’ molto semplice, TCP passa ad IP un datagramma con la destinazione e non si preoccupa di sapere come questo si relaziona con gli altri.

Figura 2.3 Flusso di dati da inviare

Ipotizzando di avere un singolo flusso di dati da inviare, TCP lo spezza in datagrammi di dimensione variabile a seconda della capacita delle reti coinvolte nel trasferimento e inserisce, nelle intestazioni, le porte (sorgente e destinazione) e il numero sequenziale.

Figura 2.4 Datagrammi creati da TCP

Ogni datagramma ha un numero sequenziale che è usato per essere sicuri di ricostruire esattamente i dati inviati.
In realtà non vengono numerati i datagrammi ma gli ottetti.
Se ad esempio inseriamo 500 ottetti ogni datagramma, il primo sarà numerato 0 il secondo 500 ecc.
Viene anche calcolato il checksum sommando tutti gli ottetti e il risultato viene inserito nelle intestazioni per poter essere confrontato con quello calcolato all’altro estremo della rete.
Se per qualche motivo sono differenti il datagramma viene considerato errato, viene eliminato e ne viene chiesta la ritrasmissione.

Nella trasmissione interviene anche un importante campo delle intestazioni, acknowledgement, che assicura la corretta ricezione dei datagrammi.
Ad esempio l'invio di un " acknowledgement number" 1500 significa che tutti i dati fino all'ottetto 1500 sono stati ricevuti correttamente.
Viceversa se la sorgente non riceve un acknowledgement in un tempo ragionevole, invia di nuovo i dati.
Il campo window e' usato per controllare la quantità di dati che può transitare contemporaneamente, non essendo pratico aspettare un acknowledgement per ogni datagramma prima di spedire il successivo.
Nello stesso tempo non si possono inviare dati indiscriminatamente rischiando di saturare la capacita' di ricezione dell'altro computer e quindi si inserisce nel campo window il numero di ottetti indicante la capacità residua di assorbire dati.
Come il computer riceve dati il valore dello spazio nel campo windows diminuisce, fino a quando giungendo a zero comporta il bloccaggio dell'invio dei dati.Come il ricevitore processa i dati incrementa il campo window e indica che e' in grado di accettare altri dati.
Esiste anche un campo urgent che permette bloccare l'elaborazione di un particolare ottetto.

2.2 Il Livello IP

I datagrammi formati dal TCP vengono passati al set di istruzioni IP. Il compito di IP, noncurante della struttura del file, sarà quello di trovare il percorso da far compiere al datagramma per consegnarlo al destinatario.
IP non si occupa di quello che contiene il datagramma o l'intestazione TCP, il suo lavoro consiste nel ricevere da TCP l'indirizzo internet del destinatario e inserire una sua intestazione che permetta ai router o agli altri sistemi intermedi di inoltrare il datagramma.

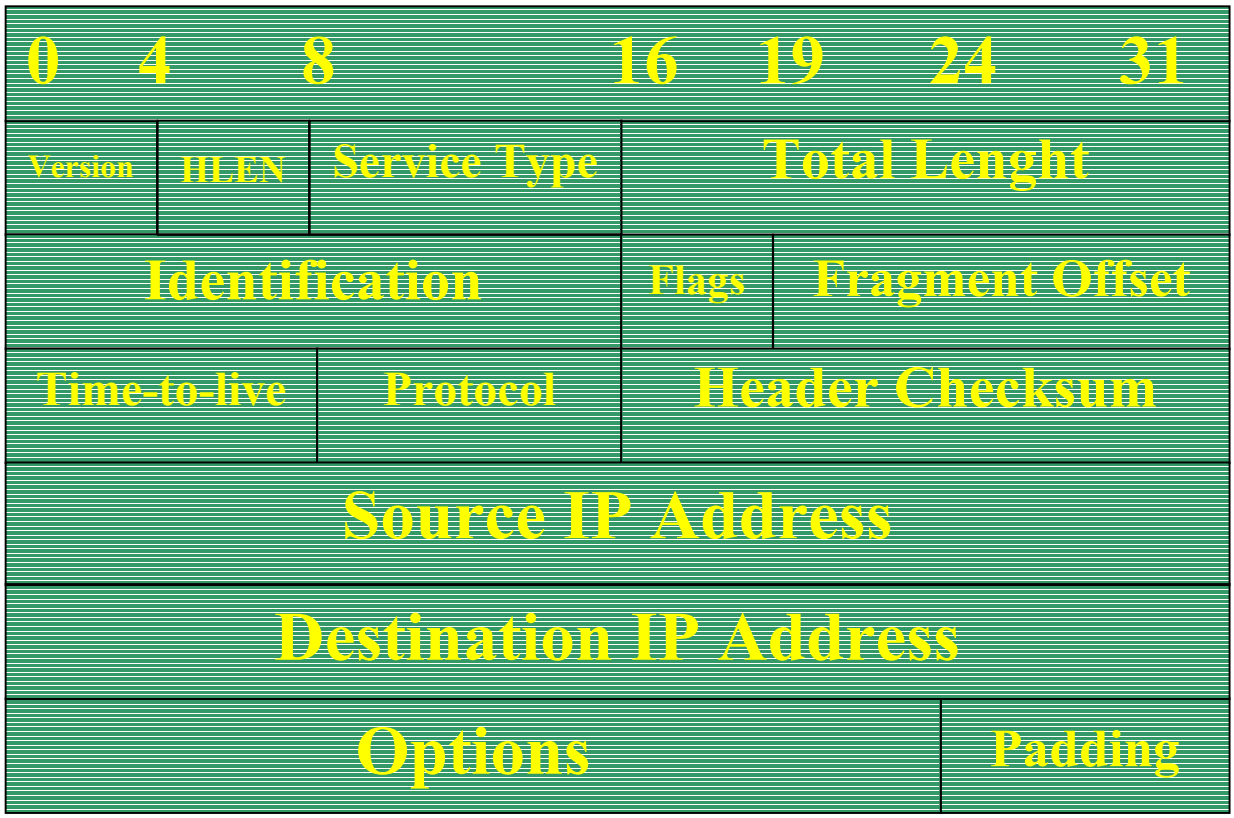

Figura 2.5 Formato di un pacchetto IP

- **Source Internet Address** : Si tratta semplicemente dell'indirizzo della macchina di partenza. Come tutti gli indirizzi Internet è costituito da un numero a 32 bit. Serve al computer che riceve i dati per identificarne la provenienza.
- **Destination Internet Address** : Un altro numero a 32 bit che caratterizza il destinatario a cui viene inviato il Mail (nel caso del nostro esempio precedente e sul quale abbiamo fondato la trattazione).
- **Protocol Number** : Identifica il tipo di protocollo usato e determina il corretto trattamento dal lato della ricezione.
- **Checksum** : Permette all'IP in ricezione di verificare che la testata IP non sia stata corrotta durante il trasferimento. Questo Checksum riguarda solo la parte header e non i dati e la parte TCP che hanno un altro Checksum.

- **Flags -- Fragment Offset** : Vengono usati per ricomporre i datagrammi nel caso in cui essi vengano divisi.
- **Time To Live** : Si tratta di un numero che viene ridotto ogni volta che il datagramma passa attraverso un sistema. Quando diventa zero, il datagramma viene automaticamente cancellato. Questo è molto utile nei casi in cui si crei un anello chiuso dal quale il datagramma non esce. In teoria questo sarebbe non possibile ma un errore umano può portare a questo tipo di alterazioni. In questi casi il TTL impedisce al sistema di collassare.
- **Service type:** specifica come un protocollo di livello superiore vuole che il pacchetto sia trattato; è possibile assegnare vari livelli di priorità utilizzando
- questo campo;
- **Total length:** è la lunghezza del pacchetto IP (header più dati) in byte;
- **Identification:** questo campo contiene un numero intero che identifica il pacchetto; è usato per permettere il riassemblaggio di un pacchetto frammentato;
- **Protocol:** identifica il protocollo di livello superiore contenuto nel campo dati del pacchetto. In appendice A, paragrafo A.7, sono riportati i codici dei
- protocolli che possono essere contenuti nel campo dati di IP;
- **Option:** è un campo usato dall'IP per fornire varie opzioni, quali la sicurezza e il source routing, che può essere di tipo loose o strict.

I campi più importanti in questa intestazione sono il source e destination internet address, il protocol number e un altro checksum.
I primi due campi sono, ovviamente, gli indirizzi delle macchine di partenza e di arrivo mentre il numero di protocollo specifica come IP deve passare il datagramma al livello successivo anche se la maggior parte del traffico utilizza TCP.

Abbiamo infine un ulteriore checksum che ci mette al sicuro da errori avvenuti nella trasmissione e ci garantisce la corretta ricezione di un pacchetto IP prima di inviarlo a TCP.
A questo punto i nostri dati hanno assunto questa forma:

Figura 2.6 Formato dati al livello IP

Se il computer e' collegato attraverso una linea telefonica diretta a quello di destinazione, o ad un router, si possono semplicemente inviare con un protocollo sincrono come l'HDLC (High Level Data Link Control un protocollo standard ITU-TSS per collegamenti punto punto o multipunto).
Comunque la maggior parte delle reti oggi usa Ethernet che quindi inserisce le sue intestazioni e il suo checksum:

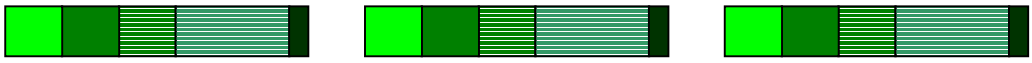

Figura 2.7 Formato dati al livello Ethernet

Quando un pacchetto cosi strutturato raggiunge la sua destinazione tutte le intestazioni sono rimosse.
Ethernet rimuove la sua intestazione e il checksum, guarda al type code e passa il datagramma, generalmente a IP.
IP rimuove le sue intestazioni e controlla il campo protocollo passando il datagramma a TCP (Eventualmente a UDP o altri protocolli) che guarda il numero sequenziale e altre informazioni per ricostruire il file originale.

L'indirizzamento IP è parte integrante del processo di instradamento dei messaggi sulla rete. Gli indirizzi IP, che devono essere univoci sulla rete, sono lunghi 32 bit (quattro byte) e sono espressi scrivendo i valori decimali di ciascun byte separati dal carattere punto.

Esempi di IP sono:

90.0.0.1 193.205.128.28 212.193.201.123 193.205.130.149

Gli indirizzi IP comprendono due o tre parti. La prima parte indica l'indirizzo della rete (network), la seconda (se presente) quello della sottorete (subnet) e la terza quello dell'host.

Occorre subito evidenziare che non sono i nodi ad avere un indirizzo IP, bensì le interfacce. Quindi se un nodo ha tre interfacce, esso ha tre indirizzi IP. Poiché la maggior parte dei nodi ha una sola interfaccia, è comune parlare dell'indirizzo IP di un nodo. Questo tuttavia è senza dubbio sbagliato nel caso dei router che hanno, per definizione, più di una interfaccia.
Gli indirizzi IP sono assegnati da un'unica autorità e quindi sono garantiti univoci a livello mondiale. Essi vengono assegnati a gruppi come dettagliato nel seguito.

Gli indirizzi IP vengono suddivisi in cinque classi, come schematizzato in figura-2.8:

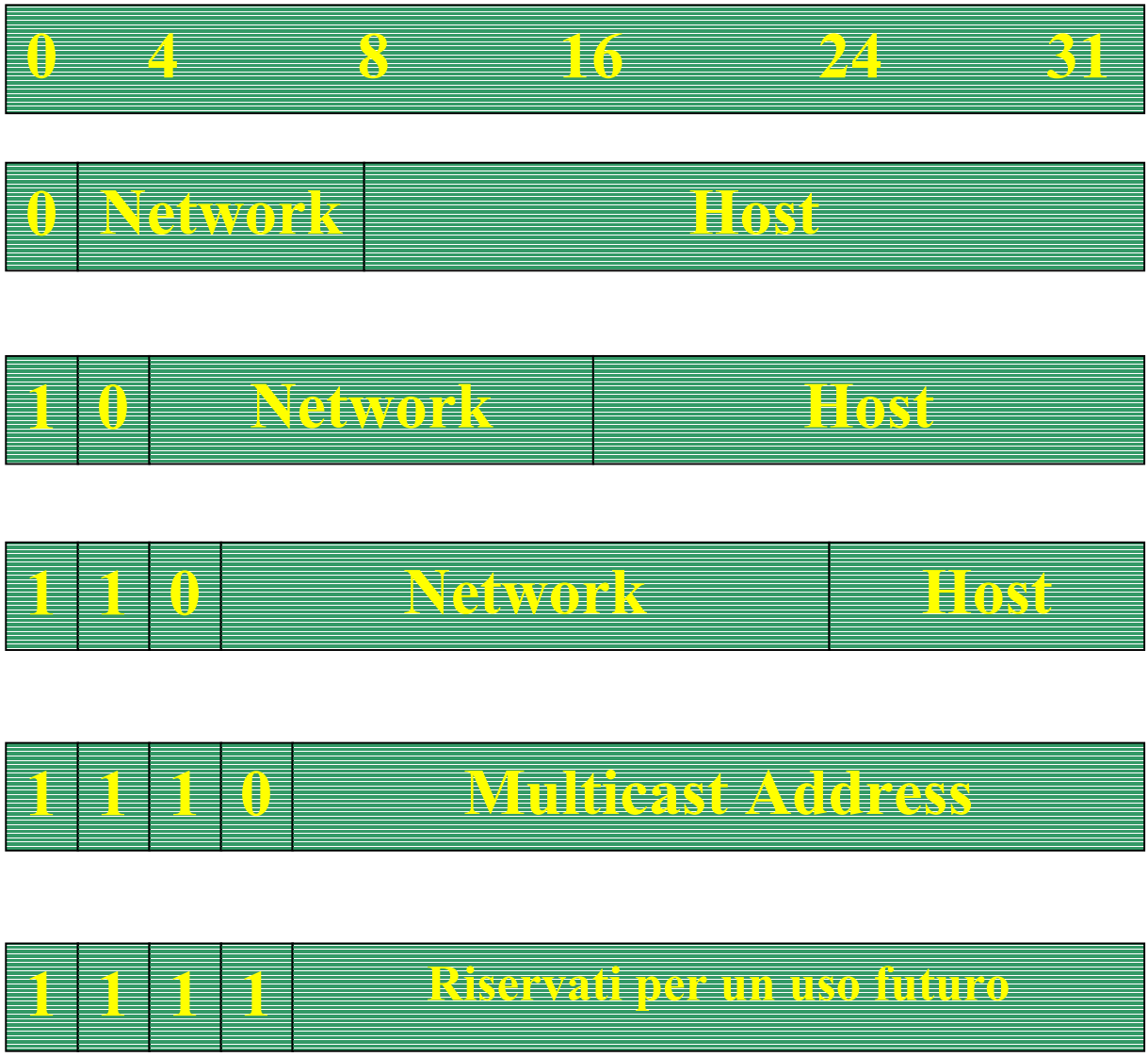

Figura 2.8 Classi di indirizzi IP

- **Classe A.** Sono concepiti per poche reti di dimensioni molto grandi. I bit che indicano la rete sono 7 e quelli che indicano l'host 24. Quindi si possono avere al massimo 128 reti di classe A, ciascuna con una dimensione massima di circa 16 milioni di indirizzi. Gli indirizzi di classe A sono riconoscibili in quanto il primo campo dell'indirizzo è compreso tra 0 e 127.

- **Classe B.** Sono concepiti per un numero medio reti di dimensioni medio-grandi. I bit che indicano la rete sono 14 e quelli che indicano l'host 16. Quindi si possono avere al massimo circa 16000 reti di classe B, ciascuna con una dimensione massima di circa 64000 indirizzi. Gli indirizzi di classe B sono riconoscibili in quanto il primo campo dell'indirizzo è compreso tra 128 e 191.
- **Classe C.** Sono concepiti per moltissime reti di dimensioni piccole. I bit che indicano la rete sono 21 e quelli che indicano l'host 8. Quindi si possono avere al massimo 2 milioni di reti di classe C, ciascuna con una dimensione massimadi 256 indirizzi. Gli indirizzi di classe C sono riconoscibili in quanto il primo campo dell'indirizzo è compreso tra 192 e 223.
- **Classe D**. Sono riservati ad applicazioni di multicast secondo quanto descritto nel RFC 1112. Gli indirizzi di classe D sono riconoscibili in quanto il primo campo dell'indirizzo è compreso tra 224 e 239.
- **Classe E**. Questi indirizzi sono riservati per usi futuri. Gli indirizzi di classe E sono riconoscibili in quanto il primo campo dell'indirizzo è compreso tra 240 e 255.

La parte host di un indirizzo di classe A, B o C può essere ulteriormente divisa in due parti: la subnet e l'host. La figura 16.4 mostra un indirizzo di classe B prima e dopo il subnetting. Il subnetting è discusso nel RFC 950. L'ampiezza dei campi subnet e host può essere definita in modo molto flessibile tramite un parametro detto *netmask*. La netmask contiene bit a uno in corrispondenza dei campi network e subnet, e a zero in corrispondenza del campo host. Ad esempio una netmask 11111111111111111111111100000000, più comunemente scritta come indirizzo IP 255.255.255.0 o in esadecimale ffffff00, indica che il campo host coincide con l'ultimo byte dell'indirizzo.

All'interno di una network IP la netmask deve essere univoca, in quanto il partizionamento della network in subnet deve essere univoco. La netmask viene

messa in AND bit a bit con gli indirizzi IP per estrarre la parte network e subnet. Tramite questo procedimento è possibile verificare se due indirizzi appartengono alla stessa subnet. Ad esempio si supponga di aver una netmask 255.255.254.0 e i due indirizzi 128.155.4.77 e 128.155.5.75.Mettendo in AND bit a bit gli indirizzi con la netmask si ottiene in entrambi i casi 128.155.4.0 e quindi gli indirizzi appartengono alla stessa subnet. Un caso di due indirizzi simili ai precedenti, ma appartenenti a subnet diverse è 128.155.5.75 e 128.155.6.77, in quanto i due AND rendono rispettivamente i valori 128.155.4.0 e 128.155.6.0.

L'importanza di comprendere se due indirizzi appartengono o no alla stessa subnet è fondamentale in quanto il primo livello di routing è implicito nella corrispondenza fissata in TCP/IP tra reti fisiche e subnet: *una rete fisica deve coincidere con una subnet IP*.

La regola che impone una corrispondenza biunivoca tra subnet IP e reti fisiche è stata ultimamente leggermente rilassata per le LAN, dove è ammesso dalle implementazioni più recenti di TCP/IP che ad una LAN possano essere associate più subnet IP. Continua a non valere il viceversa. Il concetto di subnet introduce un livello di gerarchia nelle reti TCP/IP. Il routing diventa un routing all'interno della subnet e tra subnet. Il routing all'interno della subnet è banale in quanto la subnet coincide con una rete fisica che garantisce la raggiungibilità diretta delle stazioni ad essa collegate. L'unico problema che si può incontrare è quello di mappare gli indirizzi IP nei corrispondenti indirizzi di livello 2.Questo mappaggio è oggi quasi sempre gestito in modo automatico, tramite i protocolli ARP e RARP descritti nel seguito. Il routing tra le subnet è gestito dagli IP router che originariamente erano stati definiti gateway. Tale definizione è infelice in quanto gli IP gateway sono quelli che OSI chiama router e i gateway OSI non hanno un corrispettivo nel mondo TCP/IP. Nel seguito del lavoro si useranno i termini IP router e gateway come sinonimi preferendo comunque la più moderna router.

2.2.1 Intestazioni IPv6

Vediamo di chiarire le specifiche proposte dal protocollo Ipv6.
La terminologia usata da IPv6 è la stessa di IPv4:

Nodo - dispositivo che implementa IPv6
Router - Un nodo in grado di inoltrare pacchetti Ipv6
Host - un nodo che non e' un router
Link - Capacità di trasmissione o mezzo trasmissivo attraverso il quale i nodi possono comunicare, ad esempio Ethernet; PPP links; reti X.25, Frame Relay o ATM

IP versione 6 (IPv6) una nuova versione dell'internet protocol è stata pensata per sostituire IP versione 4 [RFC-791]
I cambiamenti possono essere riassunti nelle seguenti categorie:

- Aumento della capacità di indirizzamento: IPv6 Aumenta la dimensione del campo indirizzo da 32 a 128 bit, per supportare più livelli di gerarchie, un maggior numero di nodi e una configurazione più semplice degli indirizzi. Significativa l'introduzione del campo "scope" per gli indirizzi multicast. Inoltre è stato inserito un nuovo tipo di indirizzo chiamato "anycast address", usato per inviare un pacchetto ad un gruppo di nodi.
- Semplificazione nel formato dell'intestazione del pacchetto: Alcuni campi presenti nell'IPv4 sono stati eliminati o resi opzionali, per ridurre il carico nel processare un pacchetto e la banda occupata dall'intera intestazione.
- Miglioramento delle funzioni dei campi Extension e Opzions: I cambiamenti permettono una maggiore efficienza nell'inoltrare i pacchetti e una grande flessibilità nei confronti di futuri cambiamenti.

- Capacità di classificazione dei flussi IP: Questa è una nuova funzione che permette di classificare i differenti pacchetti in base a richieste variabili di QoS o all'esigenza di servizi "real time".
- Miglioramento della procedura di autenticazione e della privacy.

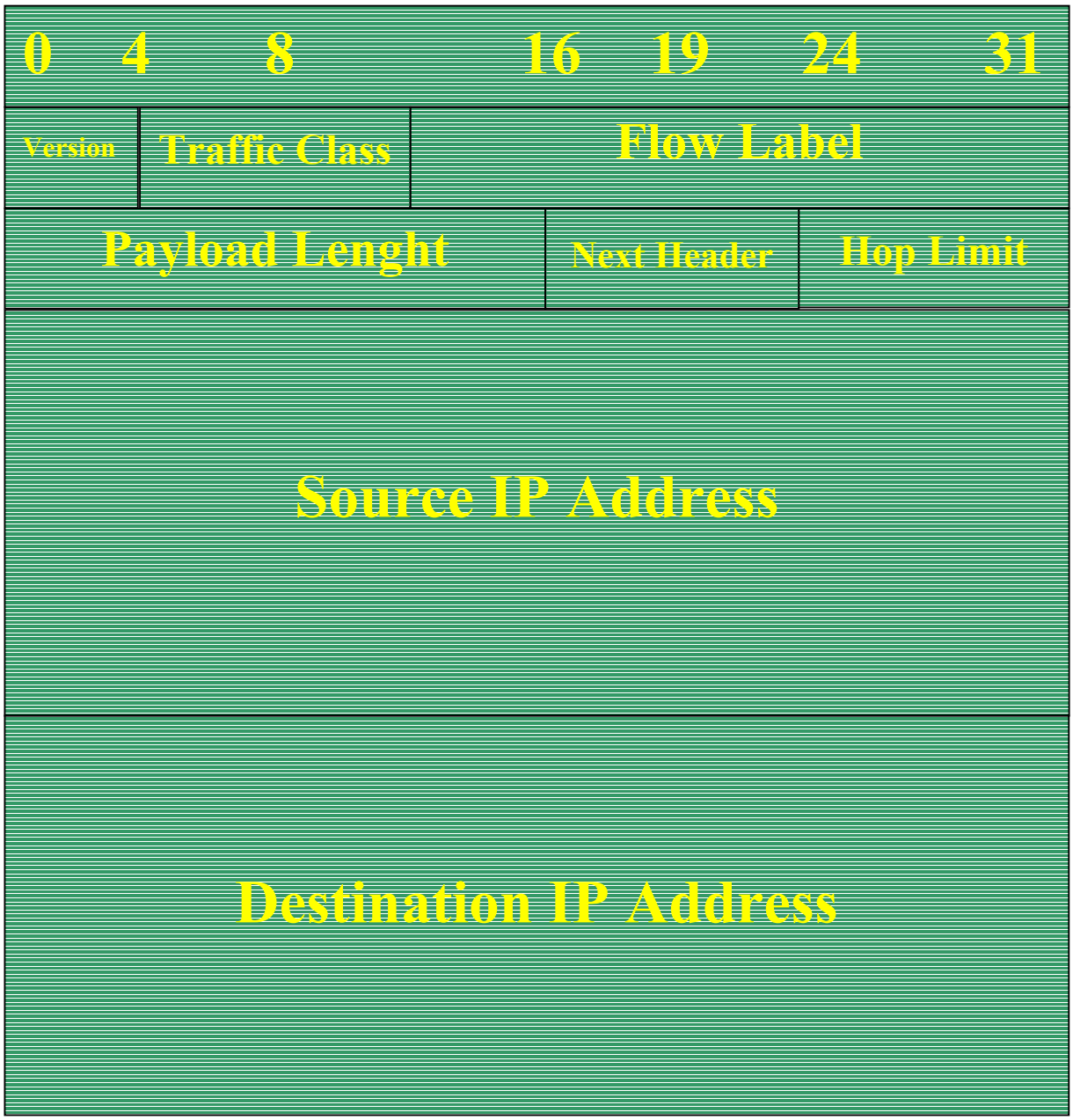

Figura 2.9 Formato di un pacchetto IPv6

Dove:

- **Version**: indica la versione del protocollo IP ed è di 4 bit.
- **Traffic Class**: campo che serve ai nodi e ai router per identificare e distinguere le classi e le priorità dei pacchetti IPv6. Contiene 8 bit.
- **Flow Label**: il campo da 20 bit dell'intestazione IP che può essere utilizzato dalla sorgente dei pacchetti per i quali è richiesto un trattamento speciale da parte dei router IPv6(QoS e servizi "real time").
- **Payload Lenght**: un intero senza segno di 16 bit che in ottetti segnala la lunghezza del payload.
- **Next Header**: identifica il tipo dell'intestazione immediatamente seguente a quella IP. Contiene 8 bit.
- **Hop Limit**: Un numero intero senza segno di 8 bit che viene decrementato di 1 ogni da ogni nodo che inoltra il pacchetto. Il pacchetto viene eliminato se si raggiunge lo zero.
- **Source Address**: 128 bit per l'indirizzo della sorgente.
- **Destination Address**: 128 bit per l'indirizzo del destinatario.

2.2.1.1 Extension header IPv6

In IPv6, alcune informazioni opzionali sono contenute in una intestazione separata che si trova tra quella IPv6 e quella del livello superiore (TCP nel nostro caso).

Hop-by-Hop: Usato per trasportare informazioni addizionali che devono essere esaminate da tutti i nodi attraversati dal pacchetto.
E' identificato da uno 0 nel campo Next Header dell'intestazione IPv6 ed ha il seguente formato:

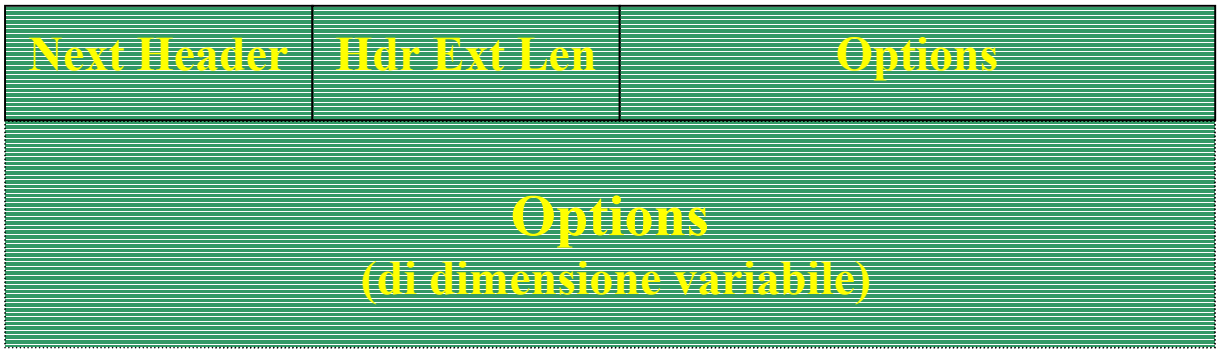

Figura 2.10 Formato dell'intestazione hop-by-hop

Next header, di 8 bit, identifica il tipo dell'intestazione immediatamente successiva all'hop-by-hop options header.

Hdr Ext Len è un intero di 8 bit. Rappresenta la lunghezza dell'hop-by-hop options header in numero di ottetti.

Options e' invece un campo di lunghezza variabile multiplo intero di 8 bit.

Destination: usato per trasportare informazioni che devono essere esaminate solo dal nodo di destinazione del pacchetto.
Ha lo stesso formato e gli stessi campi dell'intestazione hop-by-hop.

Routing header: usato da una sorgente IPv6 per elencare uno o più nodi intermedi che sono stati "visitati" dal pacchetto prima di raggiungere la destinazione.

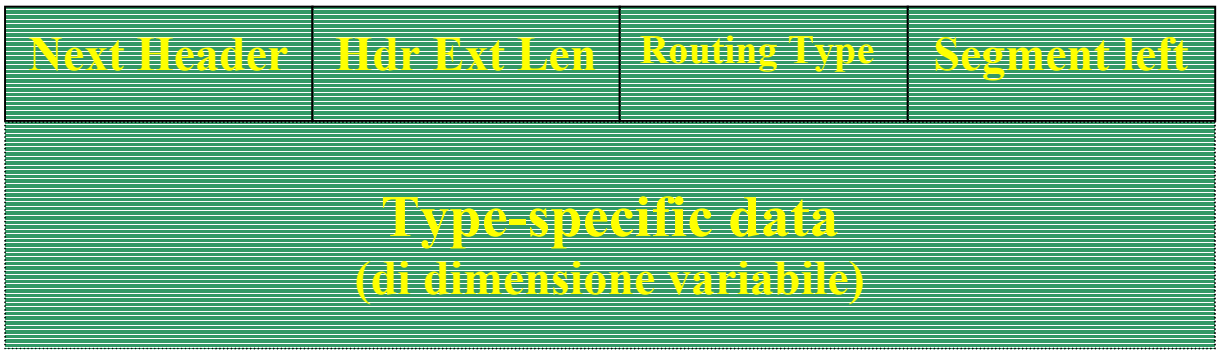

Figura 2.11 Formato dell'intestazione routing header

Next header, di 8 bit, identifica il tipo dell'header seguente al routing header.
Hdr Ext Len, intero di 8 bit, indica la lunghezza dell'intestazione considerata.
Routing Type, identifica il tipo di routing. (8 bit)
Segment Left è un intero di 8 bit che indica quanti nodi sono rimasti prima di raggiungere la destinazione.
Type-specific data e' un campo di lunghezza variabile, multiplo di 8 bit, con differente formato a seconda del routing type.

Fragment header: usato da una sorgente IPv6 per inviare un pacchetto più largo della path MTU della destinazione (inquesto caso, a differenza del precedente IPv4 la frammentazione viene attuata solo dal nodo sorgente e non dai router incontrati durante il percorso).

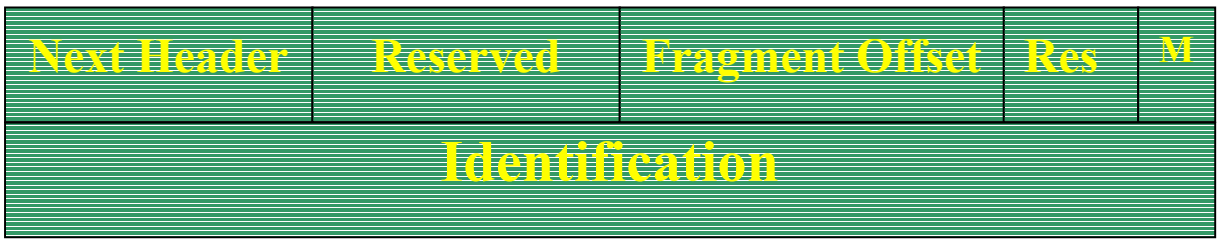

Figura 2.12 Formato dell'intestazione fragment header

Next header, di 8 bit, identifica il tipo di intestazione della parte frammentabile del pacchetto originale.
Reserved e' un campo riservato.
Fragment offset, un numero intero di 13 bit, è l'equivalente in ottetti della lunghezza dei dati che seguono questa intestazione.
Res è un campo di 2 bit a zero in trasmissione e ignorato in ricezione.
M è un flag che se settato ad uno indica la presenza di ulteriori frammenti.
Identification, di 32 bit, e' un campo che contiene informazioni per identificare i diversi frammenti

2.2.1.2 Architettura degli indirizzi IPv6

Gli indirizzi IP versione 6 sono degli identificatori a 128 bit per interfaccie o set di interfacce. Possono essere classificati in tre categorie:

- Unicast: Un identificatore per una singola interfaccia. Un pacchetto inviato ad un indirizzo unicast viene consegnato all'interfaccia che possiede tale indirizzo.

- Anycast: Identifica un set di interfaccie (in genere apparteneti a nodi differenti). Un pacchetto inviato ad un indirizzo anycast viene consegnato ad una delle interfaccie, in genere la più "vicina" (decisa in base ai protocolli di routing).
- Multicast: Indirizzo che appartiene ad un set di interfaccie appartenenti in genere a nodi differenti. Ogni pacchetto inviato a questo tipo di indirizzo viene recapitato a tutte le interfaccie.

Da notare che in IPv6 non esistono indirizzi broadcast, le cui funzioni vengono sostituite dagli indirizzi multicast e che gli indirizzi IPv6, come del resto quelli IPv4, sono assegnati alle interfaccie e non ai nodi.
Ci sono tre differenti metodi per rappresentare un indirizzo con una stringa di testo:

- quella preferita è x:x:x:x:x:x:x:x, dove x è il valore esadecimale degli 8 blocchi da 16 bit che compongono l'indirizzo. Alcuni esempi possono essere
  FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
  1080:0:0:0:8:800:200C:417A
  3FFE:B00:C18:1FFF:0:0:0:7 (indirizzo IPv6 assegnato al router del laboratorio gateway.tlc.ee.unian.it per sperimentare un tunnelling di pacchetti IPv6 su rete IPv4 .
- dati i metodi di allocazione degli indirizzi IPv6 è tipico avere lunghe stringhe di bit a zero. Per renderli più semplici e leggibili è stata studiata una particolare sintassi che riduce il numero degli zeri.
  La stringa è "::" e può essere usata una sola volta in un indirizzo.
  Ad esempio gli indirizzi:

1080:0:0:0:8:800:200C:417A	indirizzo unicast
FF01:0:0:0:0:0:0:101	indirizzo multicast
0:0:0:0:0:0:0:1	indirizzo del loopback
0:0:0:0:0:0:0:0	indirizzo non specificato

Sono rappresentati come:

1080::8:800:200C:417A	indirizzo unicast
FF01::101	indirizzo multicast
::1	indirizzo del loopback
::	indirizzo non specificato

- un alternativa, conveniente quando il nodo è misto IPv6 IPv4, è dalla struttura x:x:x:x:x:x:x:d.d.d.d dove con "x" indichiamo il valore esadecimale dei 6 blocchi da 16 bit di ordine superiore che compongono l'indirizzo mentre con "d" indichiamo il valore decimale dei 4 blocchi da 8 bit di ordine inferiore dell'indirizzo (rappresentazione standard di IPv4).
  Esempi sono:
  0:0:0:0:0:0:13.1.68.3
  0:0:0:0:0:FFFF:129.144.52.38
  o in forma compressa:
  ::13.1.68.3
  ::FFFF:129.144.52.38

2.3 Il livello Applicazione

Fino a questo punto abbiamo descritto come un flusso di dati viene spezzato, inviato e ricostruito, ma abbiamo bisogno di qualcosa di più pratico.

Ci deve essere un modo per aprire una connessione ad un computer specifico, effettuare il log in , comunicare a quale file vogliamo accedere, e controllarne la trasmissione.

Tutto questo e' realizzato dai protocolli di applicazione.

Questi lavorano sopra al TCP/IP, che si occupa della effettiva consegna del messaggio, trattando i dati come un semplice flusso di una connessione terminale o di una linea telefonica.

Molti sistemi hanno programmi separati per gestire il trasferimento dei file, le connessioni remote tipo terminale, la posta ecc.

Quando ci connettiamo ad uno specifico indirizzo dobbiamo specificare con quale tipo di Server vogliamo comunicare e su quale porta.

I programmi di rete usano queste porte, inserite nelle intestazioni TCP, in maniera più o meno casuale anche se esistono porte riservate a particolari applicazioni che risiedono in attesa di una richiesta di connessione.

Una di queste e' la porta numero 21 riservata al server FTP (File transfer protocol) che rimane in attesa di una connessione da un client che può invece usare una qualsiasi porta.

Una connessione e' descritta da un set di 4 numeri:

L'indirizzo internet e le porte TCP di entrambi gli estremi.

Quello che viene inviato e' una combinazione di dati e comandi che devono essere interpretati dall'idonea applicazione preposta a questo compito.

Ad esempio se parliamo del protocollo mail (la posta elettronica) il funzionamento e' il seguente:

Il nostro programma mail apre una connessione con il server e gli fornisce il nome della nostra macchina, il mittente e il destinatario e infine invia un comando che indica l'inizio della trasmissione dei dati.
A questo punto il server smette di ricevere comandi e inizia ad accettare il messaggio che terminerà con un carattere speciale (generalmente un punto sulla prima colonna) che segnalerà appunto la fine dei dati.
Il trasferimento file (FTP) e' qualcosa di leggermente più complesso e presuppone due distinte connessioni.
All'inizio si comporta come il programma mail inviando comandi del tipo "connettimi come utente", "questa e' la mia password", "inviami questo file" ecc..
Una volta inviati i comandi, viene aperta una seconda connessione per il trasferimento dei dati stessi.
Ovviamente si poteva usare una sola connessione, come avviene per mail, ma considerando che l'operazione di ftp può richiedere molto tempo si e' pensato comodo dividere la connessione dati da quella per i comandi per essere in grado, in qualsiasi momenti, di interrogare il server o interrompere il collegamento.
La connessione remota (Telnet) utilizza ancora un altro meccanismo:
si ha una sola connessione che normalmente invia dati ma che, se necessario, manda seguito un carattere speciale che indica che quello successivo è un comando.
Supponiamo di avere un computer chiamato gulliver.unian.it che vuole inviare il seguente messaggio:

Date: Sat, 9 Jan 1999 15:13:28 +0100
From: capriott@gulliver.unian.it
To: chifra@popcsi.unian.it
Subject : Tesi

Consegnerò il lavoro oggi pomeriggio.

Innanzitutto dobbiamo precisare che il messaggio deve essere trasmesso con lo standard net ASCII e che dovrà essere costituito da un gruppo di linee di intestazione, una linea bianca e quindi il corpo vero e proprio.
Il nostro programma a questo punto pone una serie di domande per essere in grado di recapitare il messaggio.
Per prima cosa deve sapere quale computer gestisce la posta per popcsi.unian.it (in questo caso appunto popcsi.unian.it), ne ricava l'indirizzo che è 193.205.128.3 e quindi apre una connessione TCP alla porta 25 (la porta riservata a mail) e inizia ad inviare i comandi:

POP	220 popcsi.unian.it SMTP Service at Sat, 9 Jan 1999 15:13:28 +0100
GUL	HELO gulliver.unian.it
POP	250 popcsi.unian.it – Hello, gulliver.unian.it
GUL	MAIL From:<capriott@gulliver.unian.it>
POP	250 MAIL accepted
GUL	RCPT To:<chifra@popcsi.unian.it>
POP	250 Recipient accepted
GUL	DATA
POP	354 Start mail input, end with <CRLF>.<CRLF>
GUL	Date: Sat, 9 Jan 1999 15:13:28 +0100
GUL	From: capriott@gulliver.unian.it
GUL	To: chifra@popcsi.unian.it
GUL	Subject : Tesi
GUL	
GUL	Consegnerò il lavoro oggi pomeriggio.
GUL	.

POP 250 OK
GUL QUIT
POP 221 popcsi.unian.it Service closing trasmission channel

Ogni sessione deve iniziare con un HELO, che da il nome del sistema che inizia la connessione. Poi vengono specificati mittente e destinatario (eventualmente più di uno) e inviati i dati che terminano con un punto sulla prima colonna (se questo fa parte del messaggio viene raddoppiato).
Dopo che il messaggio è stato accettato, ne possiamo inviare un altro o terminare la connessione come nel nostro esempio.
In generale, le risposte che cominciano con 2 indicano un successo, quelle con 3 che sono necessarie ulteriori azioni, 4 indica errori temporanei (disco pieno) e 5 errori permanenti (destinatario non esistente).
Nel caso di errore il messaggio viene rispedito al mittente con il relativo codice di errore.
Si nota subito che ogni risposta inizia con un numero, tipico per i protocolli internet, al quale e' associata una significato univoco.
Il testo che segue non ha nessun effetto e serve solo all'operatore per interpretare le operazioni che si stanno compiendo e per controllare eventuali errori.

Come i comandi per e-mail, FTP e telnet furono standardizzati, divenne molto più semplice per tutti utilizzare la rete e il numero di utilizzatori e di risorse iniziò la sua crescita esponenziale che continua ancora oggi.
Parallelamente divenne sempre più difficile rintracciare all'interno della rete il materiale che ci era necessario.
I primi sforzi in questo senso furono compiuti nel 1989 da Peter Deutsch presso la McGill University di Montreal dove fu creato il primo archivio per siti FTP chiamato Archie.

Questo software raggiunge periodicamente tutti i siti ftp disponibili, lista i loro files e costruisce un indice all'interno del quale si può effettuare una ricerca utilizzando comandi Unix.
Contemporaneamente Brewster Kahle al Thinking Machines Corp sviluppava il suo Wide Area Information Server (WAIS), che nel momento di massimo sviluppo conteneva l'indice di oltre 600 database sparsi nel mondo. (archiviava le FAQ di Usenet e la documentazione che sviluppava gli standard di internet).
Nel 1991 fu sviluppata dalla Università del Minnesota una interfaccia a internet molto semplice chiamata Gopher (il nome della loro mascot) ed era un sistema a menu per accedere ai file alle informazioni e alle risorse del campus attraverso la loro rete locale.
Nel giro di pochi anni vi furono oltre 10000 gopher in tutto il mondo visto che non richiedeva nessuna conoscenza dello UNIX o di architettura dei computer per poterlo utilizzare.
Bastava digitare un numero o cliccare e si ottenevano le informazioni richieste.
La versatilità di gopher fu poi incrementata quando l'università del Nevada sviluppo VERONICA (Very Easy Rodent-Oriented Netwide Index to Computerized Archives) un indice indirizzato dei menù gopher.
Nel frattempo, nel 1989, un altro evento significativo stava rendendo la rete veramente molto semplice da utilizzare.
Tim-Berners-Lee al CERN proposero un nuovo protocollo per la distribuzione delle informazioni basato su un sistema di ipertesto nel quale venivano inseriti i collegamenti ad altri documenti che potevano essere selezionati mentre si leggeva il testo.
Lo sviluppo nel 1993 del browser grafico di Mark Andreessen e del suo team al National Center For Supercomputing Applications (NCSA) diede al protocollo lo slancio per lo sviluppo futuro.
Stiamo ovviamente parlando di HTML (Hyper Text Markup Language) e del famosissimo Netscape Navigator.

Da questo momento in poi si afferma un principio che rivoluzionerà il modo di utilizzare i computer nel mondo.
Collegare tra loro due macchine diventava sempre più economico che duplicarle e applicazioni come il file transfert (FTP) e l'accesso remoto (Telnet) si sono diffuse rapidamente in tutto il mondo.
La più grande innovazione fu' comunque l'introduzione della posta elettronica che ha modificato completamente il concetto di collaborazione e comunicazione, inizialmente all'interno di internet stessa (fu un mezzo indispensabile per la comunicazione fra i vari sviluppatori) e poi in tutta la società moderna.
Grazie alla versatilità del TCP/IP, che forniva l'infrastruttura sulla quale potevano essere concepite infinite applicazioni, lo sviluppo di Internet ha seguito una crescita esponenziale che sta continuando ancora e che mette al centro della ricerca odierna il riuscire a soddisfare un ovvio aumento, anch'esso esponenziale, della Banda richiesta a livello mondiale.

2.3.1 Telnet - Collegamento remoto

Telnet è un protocollo che permette ad un utente di collegarsi, tramite l'elaboratore locale, ad un qualsiasi altro elaboratore remoto connesso alla rete.
La connessione viene attivata facendo seguire al comando telnet il nome del calcolatore remoto o il suo indirizzo. Da quel momento in poi, tutti i caratteri battuti sulla tastiera sono inviati all'elaboratore remoto e le risposte da questo generate sono
visualizzate sullo schermo locale. Il calcolatore locale è reso trasparente dal programma telnet e si opera come se si fosse direttamente connessi all'elaboratore remoto. Quando ci si scollega dall'elaboratore remoto, il programma telnet termina e ci si trova

nuovamente a dialogare con il sistema operativo dell'elaboratore locale.
Il telnet e' un comando che si effettua direttamente dal prompt in ambiente unix ma ne esistono versioni per ogni SO.

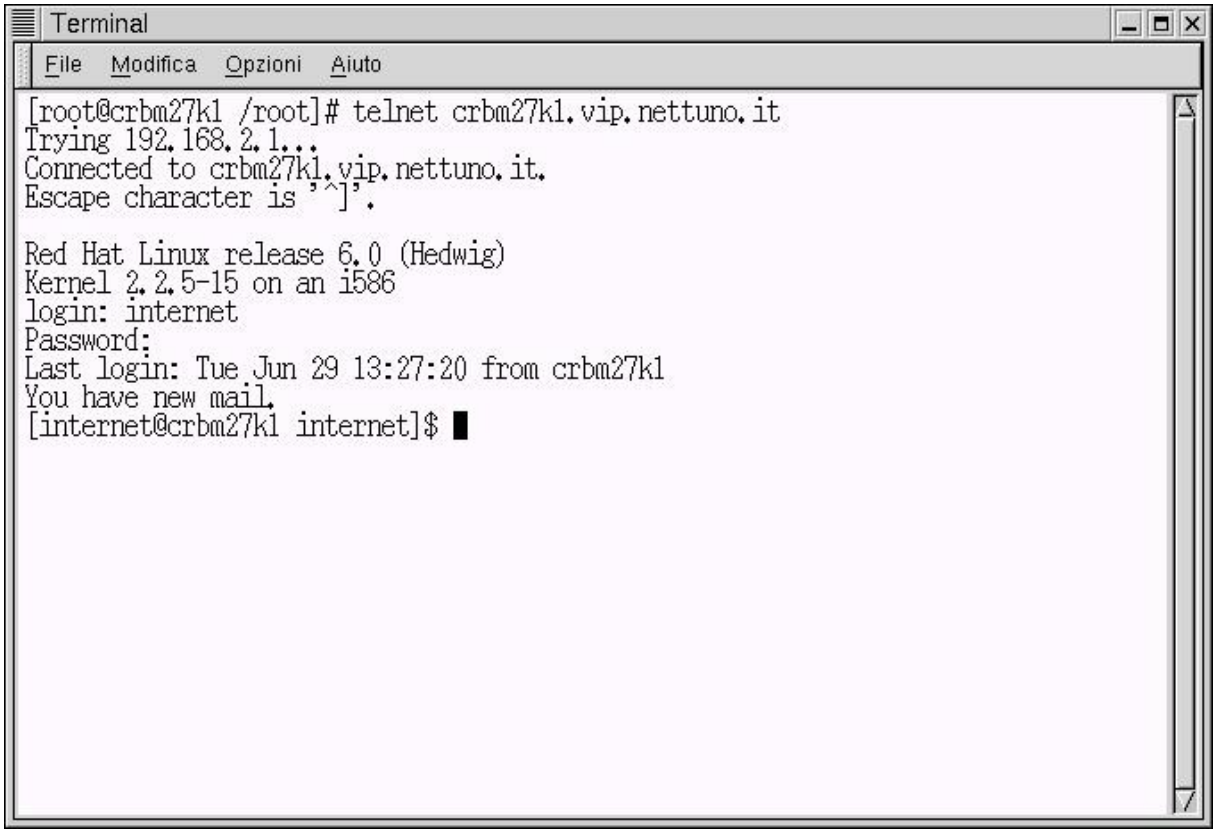

Figura 2.13 Telnet

2.3.2 SMTP (Simple Mail Transfer Protocol) - Posta Elettronica

Il Simple Mail Transfer Protocol (SMTP) è probabilmente l'applicativo più importante del TCP/IP. Esso permette di inviare posta elettronica agli utenti della rete. Ogni utente è identificato dalla sintassi Utente@Elaboratore.dominio e non è richiesta alcuna

autorizzazione per poter inviare un messaggio di posta elettronica. Il procedimento di invio avviene in batch, riprovando più volte sino a quando l'elaboratore remoto non diventa raggiungibile. L'utente remoto viene avvisato dell'arrivo di un nuovo messaggio.

I principali RFC che si occupano di posta elettronica sono lo RFC 821 e lo RFC 822.Nell'ottica di rete e condivisione delle risorse tipica dell'ambiente unix. La lettura di un messaggio ricevuto coincide con il decodificare opportunamente il contenuto del file mail presente nella directory di ogni utente abilitato al servizio. Si sono immediatamente create interfacce utente che semplificano ed automatizzano questo procedimento, dal più ostico e spartano Elm

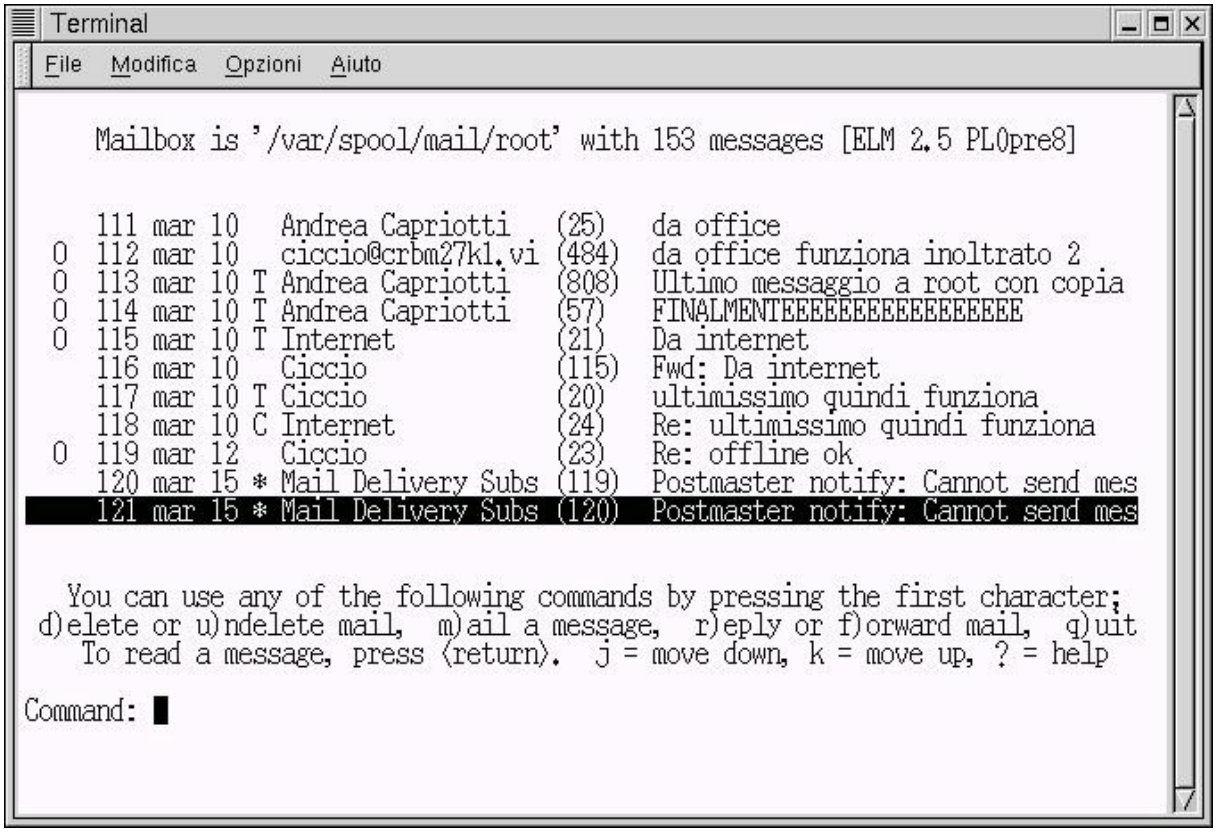

Figura 2.14 Elm

al diffusissimo pine mostrato nella figura 2.11:

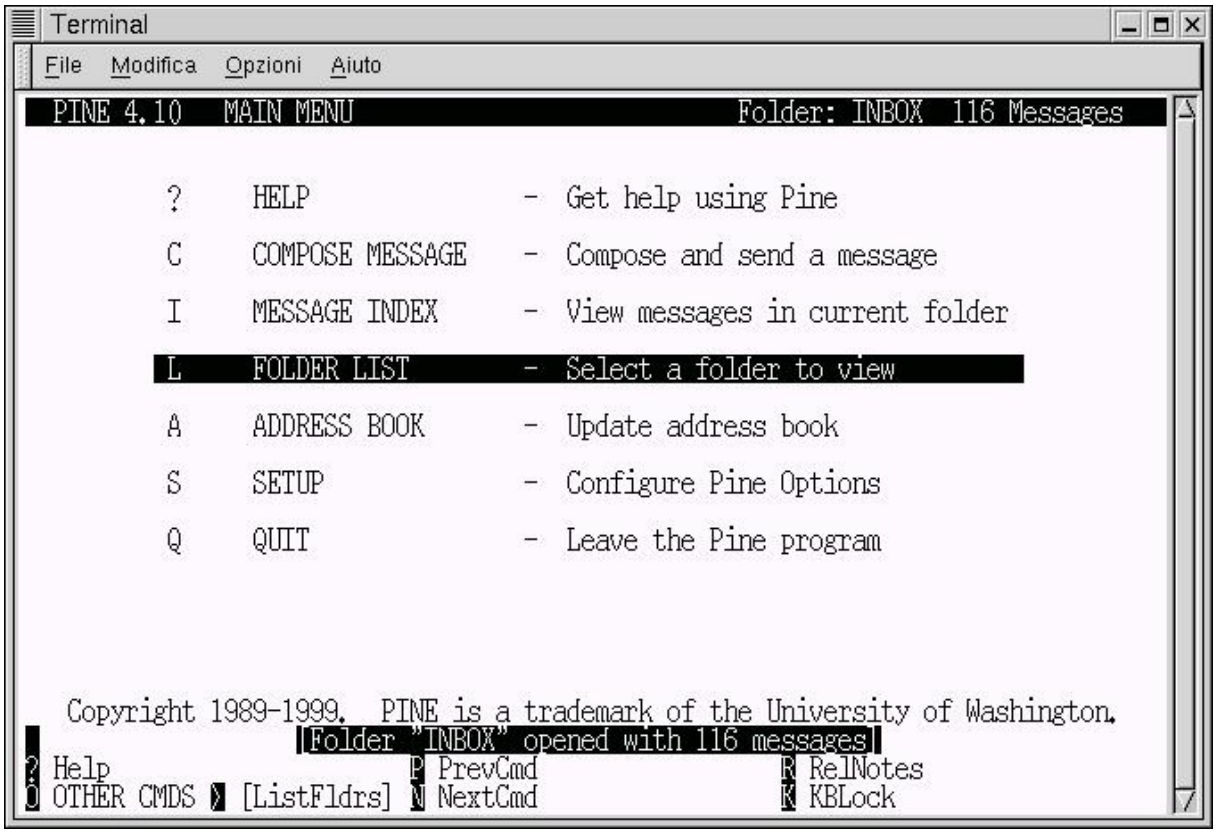

Figura 2.15 Pine

fino ai più recenti e grafici client mail.

L'introduzione della grafica ha permesso l'aggiunta di funzioni come l'invio in allegato di file o di materiale multimediale e la formattazione in html che rende i messaggi delle vere e proprie pagine web, come il messaggio pubblicitario mostrato in figura 2.16.

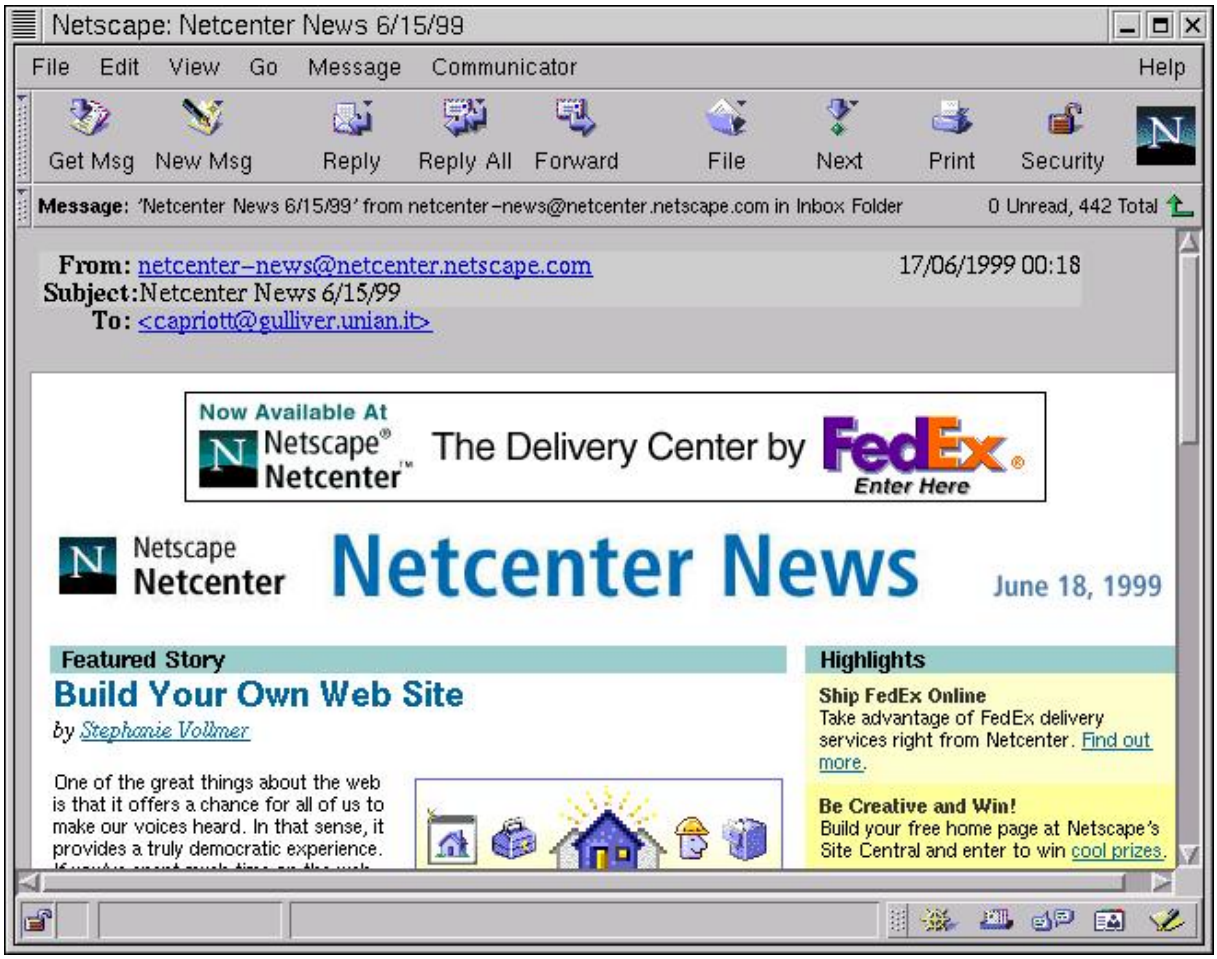

Figura 2.16 Netscape mail

2.3.3 FTP (File Trasnfer Protocol) - Trasferimento file

Il File Transfer Protocol (FTP) è un applicativo che permette ad un utente collegato ad un elaboratore di trasferire file da e verso un altro elaboratore. La sicurezza è gestita chiedendo all'utente di fornire uno username e una password validi sull'elaboratore remoto. FTP gestisce anche la conversione automatica di file di testo tra elaboratori con codifiche dei caratteri diverse. FTP è specificato

nel RFC 959. Il trasferimento file è un elemento di collaborazione e cooperazione che permette l'ottimizzazione delle risorse di un gruppo di lavoro, di una LAN/WAN o di tutta la comunità internet (e possibile concedere un accesso selettivamente ad alcuni utenti e a determinate zone dell'archivio o un accesso anonimo a zone pubbliche).Il primo ad essere introdotto è stato il semplice ftp o programmi come ncftp che ha inserito alcune funzioni aggiuntive utili all'utente, ma poi ovviamente si sono sviluppate interfacce grafiche di ogni tipo che rendono il trasferimento file da sito remoto un'operazione banale come la copia locale.

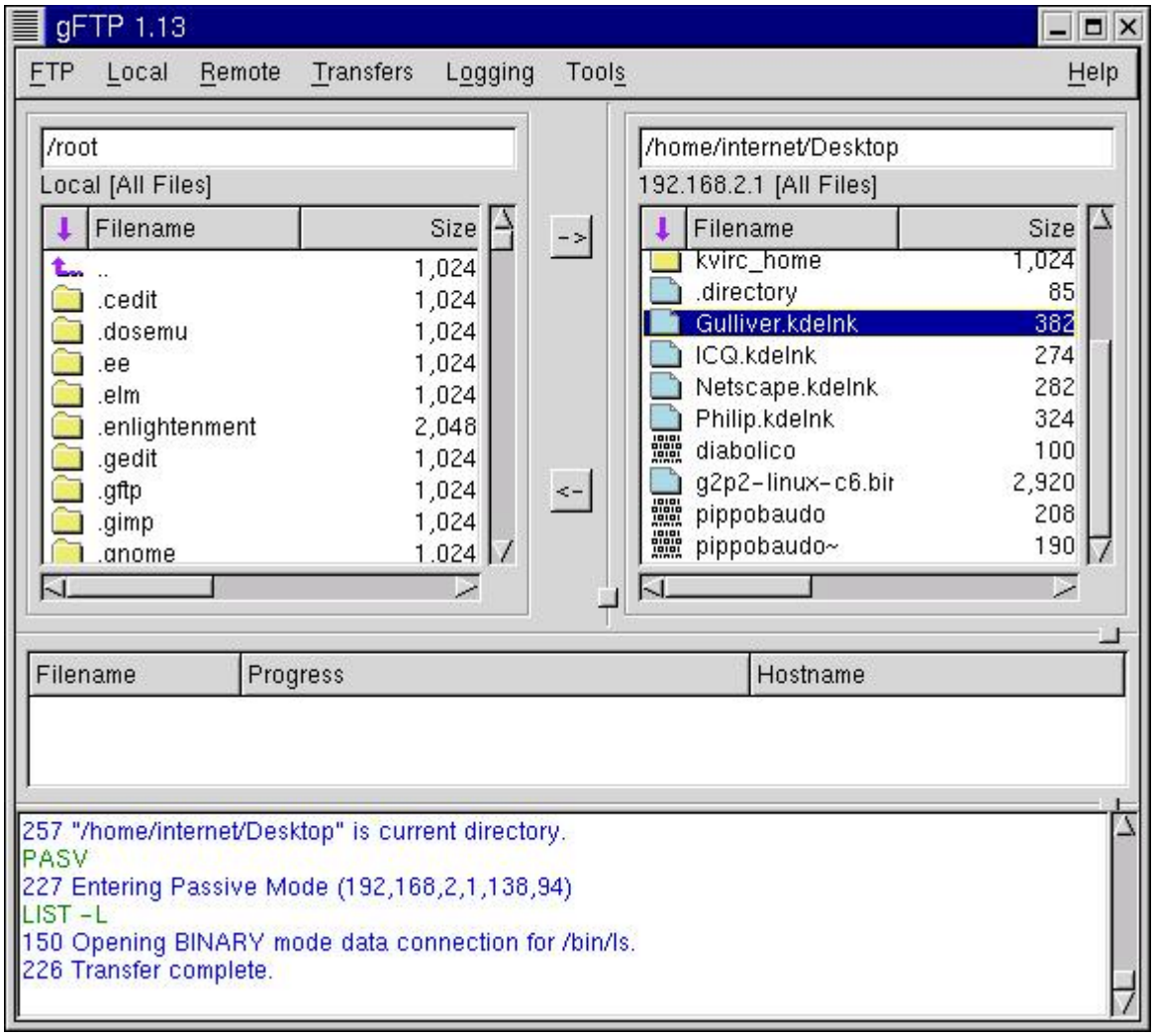

Figura 2.17 Gftp

2.3.4 RSH, REXEC e RWHO - Esecuzione remota

Le applicazioni rsh e rexec permettono di richiedere che un file di comandi o un programma eseguibile siano eseguiti su un elaboratore remoto invece che sull'elaboratore locale. L'applicazione rwho permette di verificare quali utenti siano connessi da un elaboratore remoto.

2.3.5 NFS e Netbios

Il Network File System (NFS) è un applicativo di sistema che permette a più elaboratori client di condividere un file system, messo a disposizione da un elaboratore server. Il tipo di network file system più noto è NFS proposto dalla SUN Microsystems ed adottato su tutti gli elaboratori con sistema operativo Unix.

SUN/NFS permette di avere molti server sulla rete e ad ogni elaboratore di fungere contemporaneamente da server e da client, per porzioni diversi del file system. Si appoggia su XDR (eXternal Data Representation), un pacchetto con scopi

simili al livello Presentation OSI, e questo su RPC (Remote Procedural Call) e quindi su UDP e IP.

SUN/NFS richiede una gestione coordinata della sicurezza degli elaboratori coinvolti nel file system distribuito che normalmente è realizzata con l'applicazione di sistema Yellow Pages (YP).

Un altro tipo di file system distribuito molto utilizzato in ambito personal computer si basa su Netbios ed è trattato negli RFC 1001 e 1002. Interessante il livello di condivisione tra reti basate su Netbios (Microsoft) e reti basate su NFS (unix-Linux) che si realizza con l'applicativo SAMBA.

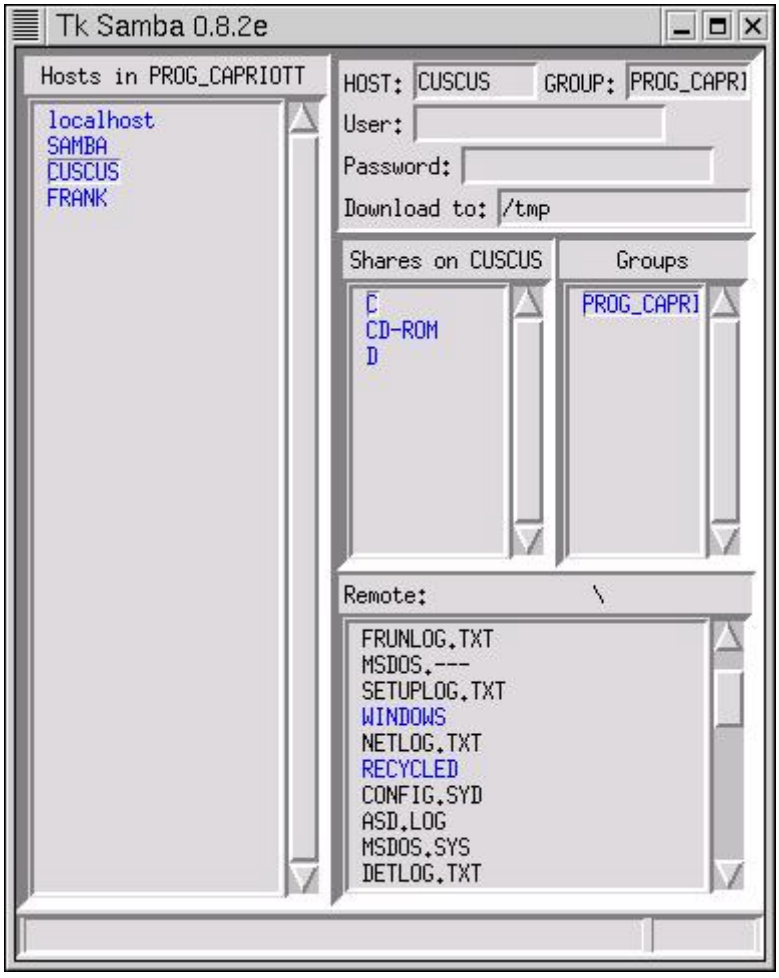

Figura 2.18 Tksmb (Interfaccia grafica del SAMBA)

2.3.6 SNMP (Simple Network Menagemant Protocol)

Il Simple Network Management Protocol (SNMP) è un protocollo per la gestione degli apparati, basato su UDP/IP. SNMP è stato progettato per inviare dati sullo stato della rete provenienti dagli apparati ad un centro di gestione che li interpreti in modo opportuno. Con SNMP è anche possibile modificare alcuni parametri degli apparati di rete.

2.3.7 X-Window

X-Window è un software di rete client-server che permette ad un programma client di visualizzare dati grafici del display di un altro elaboratore che funge da server grafico.
Nato nell'ambito del progetto MIT Athena, X-window si è diffuso su tutti gli elaboratori e su tutti i protocolli, tra cui anche TCP/IP.
Un utilizzo molto interessante di Xwindow e' quella di permettere con una semplice riga di comando, se si parla di sistemi unix, e con un programma shareware, se si e' in ambiente windows, il display di un programma che viene elaborato su di un pc remoto.
Questa funzione e' stata testata nel nostro laboratorio mostrando i limiti della tecnologia LAN tradizionale.

2.3.8 HTML - Il Web

Questo è lo strumento più grande che la rete mette a disposizione della comunicazione e della ricerca di informazioni.
La possibilità di, come per mail e ftp, di condividere spazio del file system con utenti remoti, l'esistenza di programmi come il diffusissimo Web Server Apache, la semplicità del linguaggio HTML hanno in brevissimo tempo permesso la proliferazione di siti web di ogni tipo e su qualsiasi argomento.
Sono nati, sul modello dei precedenti WAIS, GOPHER e VERONICA, dei veri e propri motori di ricerca che reindirizzano l'utente al sito che contiene l'informazione richiesta.
Inoltre la possibilità di compilare form di dati, rende questo servizio interattivo permettendo all'utente di ricercare o inserire dati per mezzo di programmi evoluti di gestione Database come PostgreSQL o MySQL,

Sono già operativi moltissimi servizi come l'iscrizione a esami o la possibilità di avere l'orario di partenza di un treno o un numero di telefono, ma le prospettive sono molto interessanti e da approfondire.

Un altro elemento degno di nota è l'integrazione di materiale multimediale all'ipertesto che oltre al lato puramente estetico e comunicativo rivela anche uno sforzo creativo verso un vero e proprio linguaggio con un suo vocabolario e una sua grammatica.