Capitolo 4: I Protocolli IP e TCP

4.1. Introduzione

La rete Internet utilizza per il trasferimento dei dati un insieme di protocolli, di cui i più noti sono il **Trasmission Control Protocol (TCP)** e l'**Internet Protocol (IP)**. L'insieme di questi protocolli consente di trasferire le informazioni attraverso un insieme di reti interconnesse ed è adatto per qualunque tipo di rete, per cui può essere utilizzato sia in reti locali, sia in reti geografiche. I protocolli utilizzati nella rete Internet sono specificati mediante standard noti con il nome di **RFC (Request For Comments)**.

Il modello TCP/IP può essere pensato, da un punto di vista concettuale, suddiviso in tre classi di servizi, come mostrato nella Figura 4.1.

Il livello inferiore fornisce il servizio di consegna dei dati senza connessione e senza riscontro, per cui non garantisce la corretta ricezione dei messaggi. *La rete è perciò inaffidabile* ed il corretto trasferimento dei dati è demandato al livello superiore. Il protocollo che definisce questo servizio è il **protocollo IP**; esso svolge le seguenti funzioni:

- definisce il **formato** dei dati che vengono trasmessi all'interno della rete.
- realizza la funzione di **routing**, ovvero il meccanismo con cui si sceglie il percorso per la trasmissione dei dati.
- prevede una serie di regole che determinano come devono essere processati i pacchetti, come e quando devono essere generati i messaggi di errore e le condizioni per le quali un pacchetto deve essere scartato.

Figura 4.1 – Struttura a livelli dei servizi in TCP/IP.

Il secondo livello in Figura 4.1 fornisce un *trasporto sicuro dell'informazione*, adottando una serie di regole e protocolli che consentono di recuperare eventuali malfunzionamenti o perdite di informazioni verificatosi nella rete. Il protocollo che realizza questo servizio è il **protocollo TCP**. Le principali funzioni svolte dal protocollo TPC/IP sono:

- controllare il flusso dell'informazione nella rete;
- fornire un trasporto sicuro dell'informazione assicurando che i dati arrivino senza errori e nell'ordine corretto;
- dividere i dati provenienti dal livello superiore in pacchetti.

Il controllo della correttezza delle informazioni è perciò realizzato **end-to-end** e non link-per-link, come in altre reti.

L'ultimo livello mostrato nella Figura 4.1 contiene i programmi applicativi (Telnet, FTP,…) che possono essere utilizzati su una rete TPC/IP. Ciascun applicativo sceglie la modalità di trasporto dei dati, che può essere nella forma di messaggi individuali o di uno stream di bit.

Un aspetto fondamentale di questa architettura, che rispecchia perfettamente la filosofia di Internet, è rappresentato dal fatto che i tre tipi di servizi vengono realizzati con protocolli software nettamente separati tra loro; in questo modo è possibile sostituire un servizio senza che gli altri ne risentano.

La struttura dei dati nei vari livelli del modello TPC/IP è mostrata nella Figura 4.2. Il livello applicativo contiene messaggi o stream continui di dati. Il livello di trasporto suddivide i dati in **pacchetti**, il livello IP in **datagram**. Le informazioni nei livelli inferiori sono strutturate secondo la rete utilizzata per il trasporto.

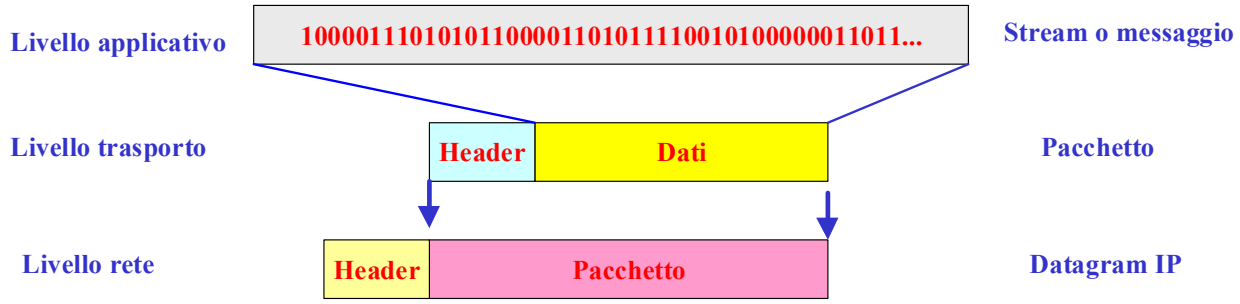

Figura 4.2 – Struttura dei dati nel protocollo TCP/IP.

4.2. Confronto tra il modello OSI e il modello TPC/IP

Il modello TPC/IP è distribuito su cinque livelli, che per la maggior parte corrispondono ai primi cinque livelli del modello OSI, come mostrato schematicamente nella Figura 4.3. Sono invece completamente assenti gli ultimi due livelli del modello OSI.

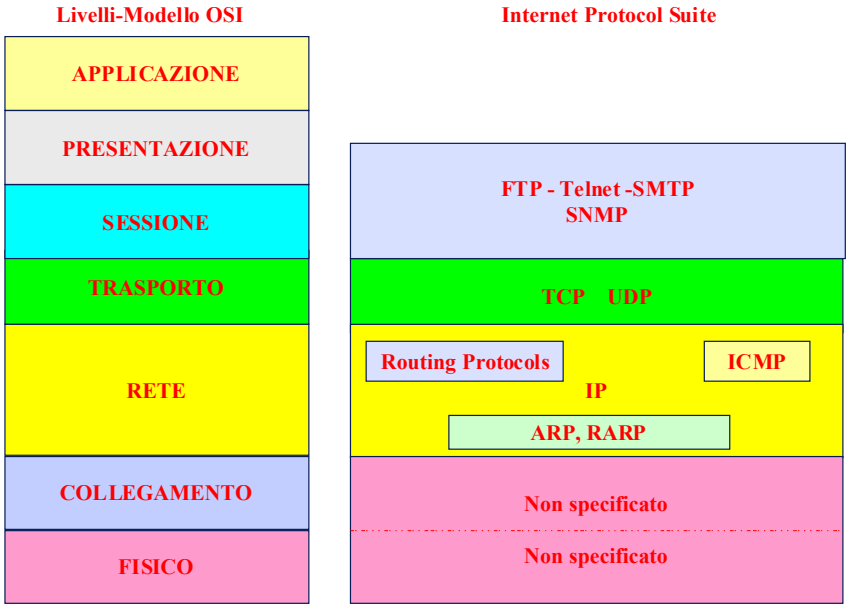

Figura 4.3 - Struttura a livelli del TCP/IP e del modello OSI.

L'architettura di rete TPC/IP non specifica il livello fisico e di collegamento, ma utilizza quelli disponibili sulla rete utilizzata. Ad esempio, nel caso di una rete LAN può utilizzare Ethernet, Token Ring, FDDI,…, mentre a livello geografico può essere utilizzata una qualunque rete geografica quale ATM, Frame Relay, X.25,….

4.3. Il protocollo IP

IP è il protocollo principale del livello di rete nell'architettura TCP/IP. Esso è stato specificato in RFC 791 e, come detto in precedenza, è di tipo *datagram*. Sono state sviluppate varie versioni del protocollo IP, che sono indicate generalmente con la sigla IPvN, dove N rappresenta il numero della versione. Attualmente è utilizzata la versione IPv4, mentre nei prossimi anni sarà introdotta la nuova versione IPv6, che fornirà nuove ed interessanti funzionalità. Nel seguito per protocollo IP si intende la versione IPv4, salvo un'esplicita dichiarazione. *Il formato del datagram* IPv4 è mostrato nella Figura 4.4. L'intestazione del datagram è formata da un insieme di campi a lunghezza fissa, ad eccezione del campo IP Option, che può essere portato fino ad una lunghezza massima di 32 bit. Il significato dei diversi campi sarà definito in dettaglio in un prossimo paragrafo.

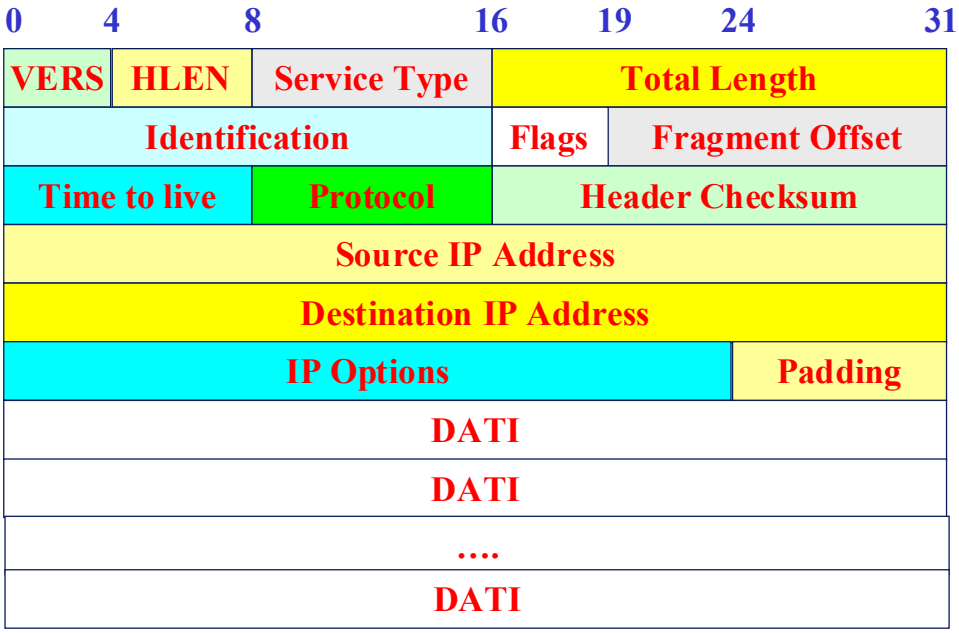

Figura 4.4 – Formato del datagram IPv4.

Incapsulamento e frammentazione del datagram

I datagram IP hanno una lunghezza massima uguale a 65.535 byte. Molte reti, che possono essere utilizzate a livello 1 e 2 per la trasmissione del datagram IP, accettano unità dati (frame) più piccole. Per questo motivo possono verificarsi varie situazioni (Figura 4.5):

- *Il datagram ha una lunghezza inferiore a quella del frame*. In questo caso il datagram è inserito direttamente nel frame (**incapsulamento del datagram**), come mostrato nella Figura 4.5.a.
- *Il datagram ha una lunghezza maggiore del frame*. Il datagram viene diviso in varie parti (**frammentazione del datagram**) e viene inserito in frame diversi, come mostrato nella Figura 4.5.b.

Ogni tipologia di rete ha una propria lunghezza massima di dati che il frame può trasportare; essa è spesso indicata come massima unità di trasferimento (**MTU = Maximun Tranfer Unit**)

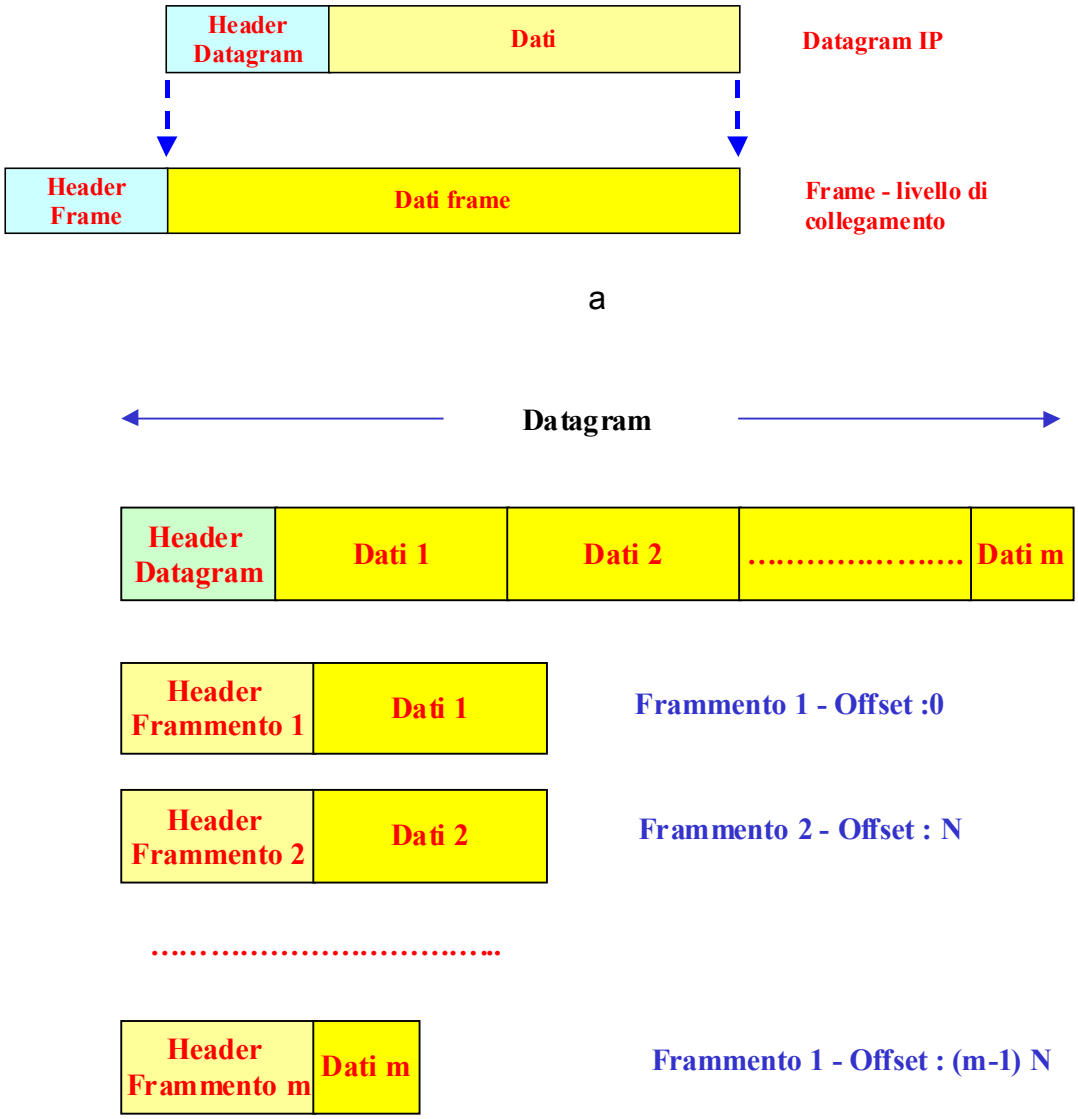

Figura 4.5 – Incapsulamento e frammentazione di un datagram IP : a) incapsulamento; b) frammentazione.

Poiché Internet è composta da molte reti con caratteristiche diverse, non è possibile prevedere a priori la lunghezza massima accettabile del frame. La scelta di utilizzare MTU piccole non è ottimale, perché rende inefficiente le tratte in cui la rete può supportare frame di dimensioni maggiori. Non essendo possibile conoscere a priori nemmeno quali cammini saranno utilizzati per la comunicazione, la soluzione attualmente adottata da TCP/IP è quella di far fissare alla sorgente una lunghezza iniziale del datagram e prevedere che questo venga frammentato in pezzi più piccoli, quando deve attraversare una rete con un MTU più basso. Ogni frammento ha lo stesso formato del datagram originario. I frammenti possiedono tutti la stessa lunghezza, sono sempre multipli di un byte, ma l'ultimo sarà di lunghezza minore od uguale agli altri. La massima efficienza viene ottenuta quando il datagram è trasportato da un unico frame.

Il protocollo IP implementa dunque un meccanismo di frammentazione e ricomposizione dei frammenti in datagram. Al protocollo IP deve essere specificato l'MTU minimo, che viene utilizzato come limite superiore alla dimensione dei datagram generati. La frammentazione può avvenire in un punto qualsiasi del cammino di un datagram verso la stazione di destinazione.

Esempio

Consideriamo le tre reti mostrate nella Figura 4.6. La seconda rete ha un MTU uguale a 128 byte, per cui i datagram provenienti dalla rete 1 o 3 devono essere suddivisi in unità con lunghezza massima di 128 byte.

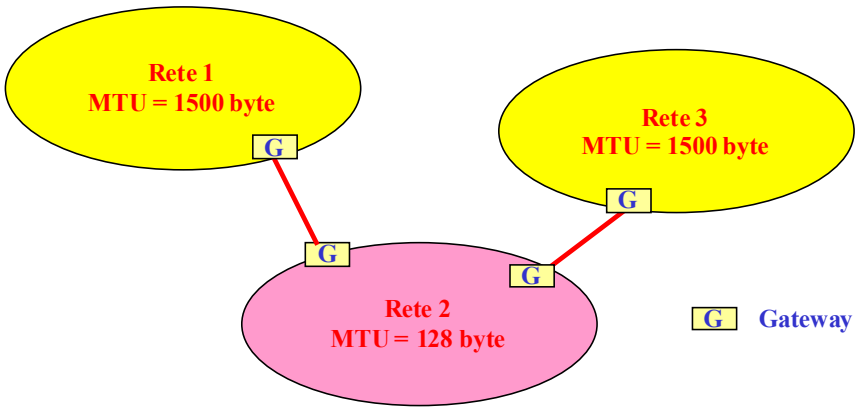

Figura 4.6 – Frammentazione di datagram in rete con diversi MTU.

Una volta che un datagram è stato frammentato, i frammenti sono trasmessi in modo del tutto indipendente attraverso la rete e vengono riassemblati solamente quando giungono a destinazione. Questa tecnica presenta i seguenti problemi:

1. I frammenti generati da una rete con un piccolo MTU potrebbero attraversare reti con grande MTU, causando quindi una perdita di efficienza;
2. Se un frammento viene perso, è necessario ritrasmettere tutto il datagram. Quando il computer in ricezione riceve un frammento, viene fatto partire un timer di riassemblamento. Se il timer si esaurisce prima che siano giunti tutti i frammenti, il computer ricevente scarta tutti i frammenti giunti fino a quel momento; per questo motivo all'aumentare del numero dei frammenti aumenta la probabilità di perdita di un datagram.

Nonostante questi svantaggi, la tecnica datagram presenta una notevole semplicità, per cui è stata adottata per la rete Internet.

Formato del datagram IP

Descriviamo brevemente il significato dei diversi campi presenti nel formato del datagram IP mostrato nella Figura 4.4:

- Vers: composto da 4 bit, contiene la versione del protocollo IP usato per la creazione del datagram. Serve a verificare che il trasmettitore e il ricevitore utilizzino lo stesso formato. Attualmente è utilizzata la versione 4.
- Hlen: composto da 4 bit, fornisce la lunghezza dell'header del datagram IP (che può risultare variabile a causa del campo option), espressa come numero di parole da 32 bit. L'header più comune, che non presenta il campo option né il campo padding, misura 20 byte. In questo caso HLEN = 5.
- Service type: composto da 8 bit, specifica come un protocollo di livello superiore deve trattare il pacchetto. Questo campo è suddiviso in 4 sotto campi, come mostrato nella Figura 7 con il seguente significato:
- I bit di PRECEDENCE indicano il grado di importanza del datagram, con valori che vanno da 0 (priorità normale) a 7 (controllo della rete). Anche se spesso questo campo è ignorato dal software IP, tale valore fornisce un meccanismo per dare priorità ai pacchetti di controllo rispetto a quelli dati.
- I bit D, T e R specificano il tipo di trasporto richiesto per il datagram ed indicano:
  - campo D: Ritardo accettabile (1 = basso ritardo, 0 = valore non critico)
  - campo T: Throughput accettabile (1 = alto throughput, 0 = valore non critico)
  - campo R: Affidabilità (1 = alta affidabilità, 0 = valore non critico)

  Questi campi servono per il routing

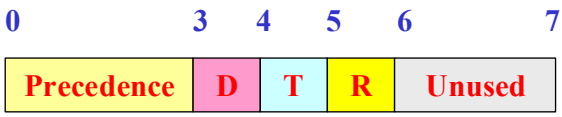

Figura 4.7 – Suddivisione del campo *Service type* del datagram IP

- Total length: composto da 16 bit, contiene la lunghezza del datagram o del frammento misurata in byte, inclusa l'intestazione. La massima lunghezza del datagram è 65.535 byte.
- Identification: composto di 16 bit; contiene un numero intero che identifica il datagram e quindi permette di individuare il datagram a cui appartiene un "frammento" che arriva a destinazione.
- Flags: composto da 3 bit. Il primo bit indica se il frammento è o no l'ultimo di un datagram. Il secondo bit indica se è possibile frammentare o no un datagram, mentre l'ultimo bit indica se il datagram è o no un frammento.
- Offset: composto da 13 bit; indica l'offset del frammento in multipli di 8 byte. Il primo frammento ha un offset uguale a 0. L'ultimo frammento ha un offset multiplo dei precedenti, ma con un total length minore od uguale ai precedenti.
- Time to live: composto da 8 bit; esprime il tempo in secondi che il datagram può rimanere nella rete Internet prima di essere scartato. Quando un datagram viene generato, questo campo viene posto uguale a 255 (massimo valore). Il valore del contatore viene diminuito con il passaggio del tempo; quando il contatore diviene uguale a 0, il datagram viene scartato. Questo contatore evita che un datagram permanga troppo a lungo nella rete a causa di malfunzionamenti.

- Protocol: composto da 8 bit; specifica il protocollo di livello superiore utilizzato per creare il pacchetto contenuto nel campo dati del datagram.
- Header checksum: composto da 16 bit; controlla se l'header del datagram contiene o no errori. Se l'header contiene errori, il datagram viene scartato. Il protocollo TCP dovrà provvedere al recupero del datagram.
- Source IP address: composto da 32 bit; contiene l'indirizzo IP della stazione che ha generato il datagram.
- Destination IP address: composto da 32 bit; contiene l'indirizzo IP della stazione a cui è diretto il datagram.
- Option: ha una lunghezza variabile; consente di fornire una serie di opzioni, quali la sicurezza e il source routing. Il source routing impone a un datagram un determinato percorso e può essere utile per analizzare il comportamento di una rete.
- Padding: questo campo ha una lunghezza variabile; serve a rendere l'header un multiplo di 32 bit.
- Dati: questo campo ha una lunghezza variabile; contiene i dati informativi.

Indirizzi IP

Ogni computer connesso alla rete Internet ha un **indirizzo IP** che lo individua in modo univoco. L'indirizzo IP è costituito da un numero binario di **32 bit**, cioè da 4 byte. Per semplicità gli indirizzi IP sono generalmente espressi mediante i 4 numeri decimali corrispondenti ai 4 byte separati dal carattere punto.

Esempio
L'indirizzo IP in binario 10000010000011100000001000011110 diventa in rappresentazione decimale 130.14.2.30

Per facilitare l'uso degli indirizzi, si ricorre spesso ad **indirizzi simbolici**.

Esempio
L'indirizzo numerico del server informativo dell'Università di Siena è 193.205.4.2, mentre quello simbolico è unisi.it

I 32 bit che formano un indirizzo IP possono essere divisi in due campi (Figura 4.8):

- **Indirizzo della rete** (**netid**), che identifica la rete su cui si trova il computer;
- **Indirizzo del computer** (**hostid**), che identifica il computer all'interno della rete.

Figura 4.8 – Formato di un indirizzo IP.

Gli indirizzi IP possono essere divisi in 5 classi (Figura 4.9), denominate **classe A, B, C, D e E**. L'identificazione della classe avviene osservando i primi bit. Il formato degli indirizzi Internet per le diverse classi è mostrato nella Figura 4.9.

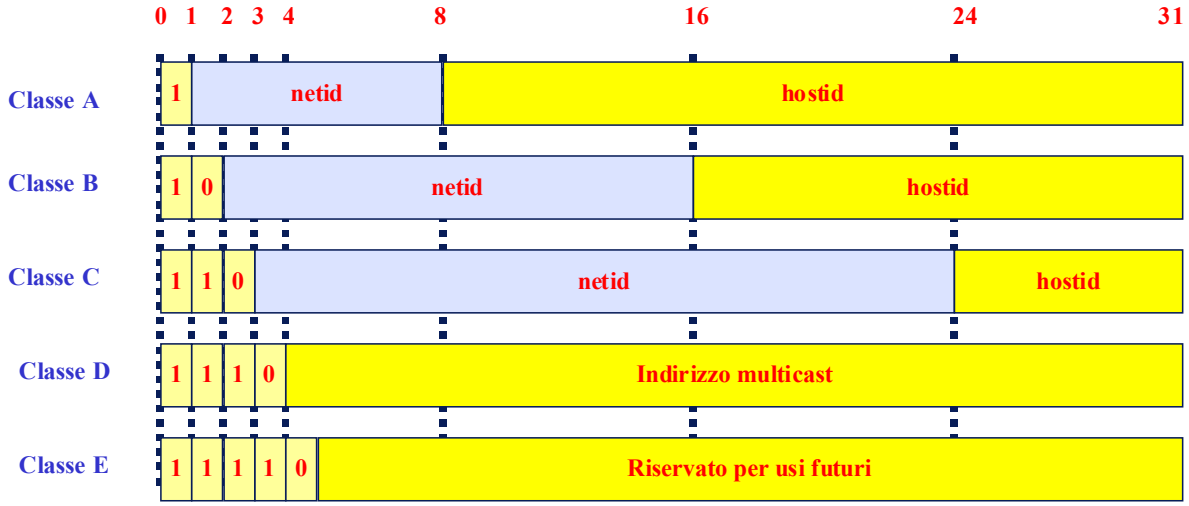

Figura 4.9 - Formato degli indirizzi Internet per le varie classi.

Caratteristiche delle classi degli indirizzi IP

Classe A: utilizza 7 bit per netid e 24 per hostid. Possono perciò esistere un numero massimo di 128 reti di classe A, ciascuna delle quali può contenere al massimo 2^{24} = 15.777.216 computer.
Classe B: utilizza 14 per bit per netid e 16 per hostid. La classe B è adatta a reti che hanno un numero di computer compreso tra 256 e 2^{16} = 65536.
Classe C: utilizza 21 bit per netid e 8 bit per hostid. La classe C viene utilizzata per reti che hanno un numero di computer inferiore a 256.
Classe D: è riservata ad applicazioni di multicast.
Classe E: è stata definita per usi futuri. Gli indirizzi di questa classe sono facilmente riconoscibili in quanto il primo campo dell'indirizzo è compreso tra 240 e 255.

Nella tabella 4.1 sono riassunte le dimensioni delle varie classi IP.

	N. MAX RETI	N. MAX COMPUTER PER RETE
CLASSE A	128	16777216
CLASSE B	16384	65536
CLASSE C	2097152	256

Tabella 4.1 - Dimensioni delle reti IP nelle varie classi.

Gli indirizzi IP sono stati realizzati in modo che sia possibile estrarre facilmente il **netid,** ovvero l'identificatore alla rete a cui appartiene un dato computer. Oltre che per le interfacce, gli indirizzi IP vengono usati anche per identificare una **rete**. Per convenzione, l'indirizzo di una rete è quello che ha l'**hostid con tutti 0**.

Esempio
L'indirizzo 10000001000001110000000000000000 corrisponde alla rete di classe B con il seguente indirizzo 130.14.0.0

Indirizzi IP speciali

Un'altra convenzione utilizzata è quella che riguarda il **broadcast** su una rete (***directed broadcast***). Il broadcast su una rete si ottiene ponendo **tutti i bit dell'hostid a 1**.

Sottoreti IP

Nello schema di indirizzamento IP originale (divisione dell'indirizzo IP in netid e hostid) si prevede che ad ogni rete fisica venga assegnato un unico indirizzo di rete. Questo aspetto e l'enorme crescita di Internet ha comportato i seguenti problemi:

- La più piccola rete prevista è quella di classe C (256 indirizzi): se una classe C viene assegnata ad una rete con pochi computer, la maggior parte degli indirizzi rimane inutilizzata.
- L'enorme numero di reti collegate ad Internet, molte delle quali di piccole dimensioni e ciascuna con un proprio indirizzo di rete, comporta l'aumento delle dimensioni delle tabelle di routing e dei compiti di gestione degli indirizzi.

Per ridurre il numero degli indirizzi di rete, è stato necessario trovare un meccanismo per far condividere lo stesso indirizzo di rete IP a più reti fisiche. Tale tecnica prende il nome di **subnet addressing** o **subnetting** ed è obbligatoria in ogni implementazione del protocollo IP.

Esempio

Nella Figura 4.10 è mostrato un esempio di subnetting. Il router R utilizza lo stesso indirizzo in classe B (128.10.0.0) per le due reti fisiche ad esso collegate.

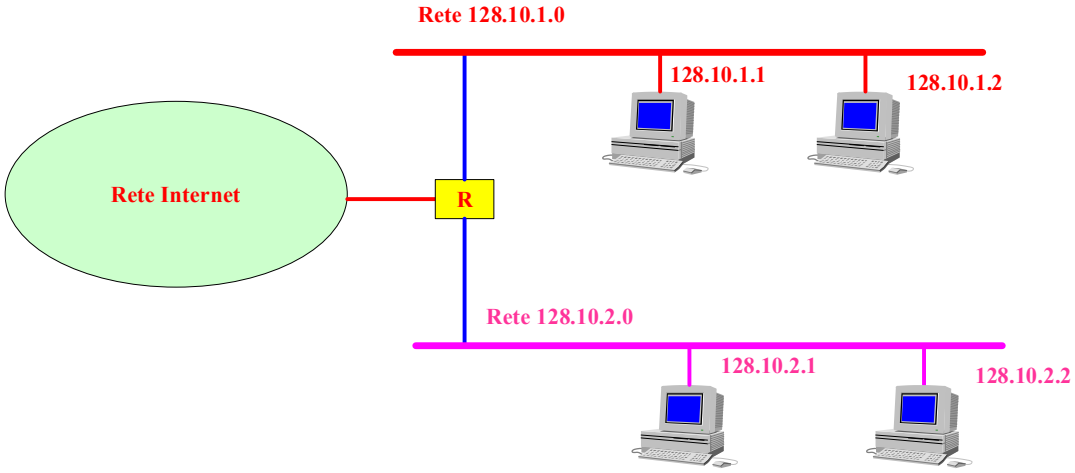

Figura 4.10 - Un sito con due reti fisiche che usa la subnettazione dell'indirizzo.

Tutti i router di Internet, tranne R, vedono le due reti come un'unica rete fisica di indirizzo 128.10.0.0. Quando un datagram arriva al router R, questo provvede ad inviarlo sulla rete fisica di destinazione. Per rendere più efficiente questa fase, l'amministratore di rete ha utilizzato il terzo byte dell'indirizzo per distinguere le due reti.

Le macchine sulla prima rete hanno un indirizzo che inizia con 128.10.1, mentre le macchine sulla seconda rete hanno l'indirizzo che inizia con 128.10.2. Per instradare il datagram il router R esamina i primi tre byte dell'indirizzo IP di destinazione.

Suddividere una rete in sottoreti significa interpretare l'indirizzo IP in modo leggermente diverso rispetto a quanto esposto fino ad ora (come mostrato nella Figura 4.11). Nella Figura 4.11.a è mostrato lo schema classico di un indirizzo IP, mentre nella Figura 4.11.b è mostrato uno schema di una possibile subnettizzazione. In esso il campo netid rimane inalterato, mentre il campo hostid viene suddiviso in due parti, cioè:

- subnet
- host

a

b

Figura 4.11 – Caratteristica dell'indirizzo IP nel caso di subnettizzazione : a) indirizzo IP standard; b) indirizzo IP nel caso di subnettizzazione.

L'ampiezza dei campi subnet e host può essere definita in modo molto flessibile tramite un parametro detto **netmask**. La netmask è composta da bit uguali ad 1 in corrispondenza dei campi di netid e subnet e uguale a 0 in corrispondenza del campo host. *Anche le netmask sono spesso rappresentate da una notazione dotted decimale*.

Esempio
La netmask 11111111 11111111 11111111 00000000 corrisponde all'indirizzo IP 255.255.255.0 ed indica che il campo host coincide con l'ultimo byte.

Dato un indirizzo IP, *per estrarre l'indirizzo della rete e del subnet si effettua un'operazione AND bit a bit tra l'indirizzo IP e la netmask.*

Esempio
Sia dato l'indirizzo IP 128.10.2.2; la sua rappresentazione binaria risulta:
100000000 00001010 00000010 00000010
Utilizzando la netmask 255.255.250.0 la cui rappresentazione binaria è:
11111111 11111111 11111010 00000000
ed effettuando l'operazione AND tra le due sequenze si ottiene la sequenza:
10000000 00001010 00000010 00000000
che corrisponde all'indirizzo:
128.10.2.0
e quindi l'indirizzo IP 128.10.2.2 appartiene alla rete 128.10.2.0. La rete 128.10.2.0 è una sottorete della rete in classe B 128.10.0.0. Utilizzando la netmask con l'indirizzo IP 128.10.3.4 si ottiene ancora 128.10.2.0 e quindi anche questo indirizzo IP appartiene alla stessa rete di 128.10.2.2

Con riferimento agli indirizzi di rete IP in classe C si possono definire le subnet mask contigue mostrate nella tabella 4.2.

Subnet mask (dotted decimal notation)	Subnet mask (ultimo byte)	N. reti/classe C	N. indirizzi IP/sottorete	N. host/sottorete
255.255.255.0	00000000	1	256	254
255.255.255.12	10000000	2	128	126
255.255.255.19	11000000	4	64	62
255.255.255.22	11100000	8	32	30
255.255.255.24	11110000	16	16	14
255.255.255.24	11111000	32	8	6
255.255.255.25	11111100	64	4	2

Tabella 4.2 – Subnet mask contigue.

Protocolli di instradamento IP

Le tecniche di instradamento dei datagram possono essere divise in **routing diretto** e **routing indiretto**. Per **routing diretto** si intende la trasmissione di un datagram da una macchina direttamente ad un'altra. Il routing diretto si può avere solo se entrambe le macchine sono connesse alla stessa rete fisica. Viene utilizzato il **routing indiretto** quando le macchine comunicanti non si trovano sulla stessa rete fisica, costringendo il mittente ad inviare il datagram ad un router per la consegna.

Routing diretto

La trasmissione di un datagram IP tra due macchine della stessa rete fisica non coinvolge i router. Il mittente incapsula il datagram in un frame fisico, mappa l'indirizzo IP con l'indirizzo fisico e invia il frame direttamente alla macchina destinazione. Poiché gli indirizzi IP di tutte le macchine che appartengono alla stessa rete fisica hanno lo stesso netid e poiché il netid può essere estratto dall'indirizzo IP con poche istruzioni, risulta semplice scoprire se una macchina può essere raggiunta direttamente. Il routing diretto può essere visto come consegna del datagram e rappresenta anche l'ultimo passo nella trasmissione del datagram. Infatti l'ultimo router del percorso è connesso alla stessa rete fisica della

macchina destinazione, dunque userà il routing diretto per consegnare il datagram alla macchina destinazione.

Il routing diretto può essere visto come un caso particolare del routing generico, in cui il datagram non transita in nessun router intermedio.

Routing indiretto

Il routing indiretto presenta una complessità superiore al routing diretto, perché il mittente deve individuare il router a cui inviare il datagram; il router deve poi inoltrare il datagram verso la destinazione finale. Il percorso dalla sorgente alla destinazione è realizzato mediante la trasmissione del datagram da un router all'altro, fino a quando il datagram non arriva direttamente al router a cui la stazione di destinazione è collegata. Sulla rete Internet i router costituiscono una struttura cooperativa e interconnessa.

Tabelle di routing

L'instradamento del traffico IP si basa sulle **tabelle di routing,** tabelle in cui vengono memorizzate le informazioni su come raggiungere una qualsiasi destinazione. Poiché sia gli host che i router instradano datagram, entrambi hanno le tabelle di routing; ogni volta che il protocollo IP deve inviare un datagram consulta la tabella di routing per decidere dove inviarlo. Se ogni tabella di routing contenesse le informazioni su ogni singolo indirizzo IP presente su Internet, avrebbe dimensioni enormi. Il concetto che si cerca di applicare è quello di avere un **routing con la minima informazione**. Per esempio, sarebbe preferibile confinare l'informazione sulle singole macchine nell'ambito locale in cui si trovano e consentire la loro raggiungibilità alle macchine remote senza che queste ne conoscano i dettagli. Per questo gli indirizzi IP sono assegnati in modo tale che tutte le macchine sulla stessa rete fisica abbiano lo stesso prefisso (il netid). Ciò consente di riportare nelle tabelle di routing solamente gli indirizzi identificativi delle reti e non gli indirizzi IP completi, riducendo quindi la dimensione delle tabelle, migliorando l'efficienza dell'informazione e confinando i dettagli sui singoli host nell'ambito locale in cui essi operano.

Tipicamente una routing table è costituita da una sequenza di coppie (**N**,**G**) dove:

- **N** indica l'indirizzo IP della rete di destinazione;
- **G** indica il **next hop**, ovvero l'indirizzo IP del prossimo router lungo il percorso verso la rete N.

Questo indirizzo deve riferirsi sempre ad una macchina direttamente raggiungibile dalla macchina a cui appartiene la tabella di routing.

Le tabelle di routing per gli host devono contenere la minima informazione possibile: l'idea è quella di costringere gli host ad affidarsi ai router per la maggior parte delle procedure di instradamento.

La scelta di effettuare un routing basato sul netid comporta tutta una serie di conseguenze, quali:

- Nella maggior parte dei casi tutto il traffico diretto ad una data destinazione prende la stessa strada, per cui, anche se esistono più percorsi per quella destinazione, non possono essere usati in modo concorrente.
- Poiché l'host di destinazione è connesso ad un router, solamente quest'ultimo può determinare se l'host esiste o è attivo. Quindi è necessario che il router implementi un meccanismo che segnali la situazione di errore alla sorgente.
- Poiché ogni router instrada il traffico in modo indipendente, i datagram che viaggiano da un host A ad un host B possono seguire un percorso completamente diverso di quelli che viaggiano da B a A. I router devono dunque collaborare per fare in modo che la comunicazione nei due sensi sia sempre possibile.

Autonomous system

Le reti e i router collegati alla rete Internet sono divise in gruppi; ogni gruppo è controllato da un'unica autorità amministrativa, chiamata **Autonomous system (AS)**. Ogni AS è identificato da un numero univoco a livello internazionale e rilasciato dall'autorità che assegna gli indirizzi Internet. I router possono essere classificati in:

- **Router esterni**: se appartengono a diversi AS (Figura 4.12), in questo caso utilizzano per scambiare informazioni di instradamento un protocollo **EGP (Exterior Gateway Protocol)**.
- **Router interni**: se appartengono allo stesso AS; in questo caso utilizzano per scambiare informazioni di instradamento un protocollo **IGP (Interior Gateway Protocol)**. All'interno di un AS tutti i router utilizzano generalmente lo stesso IGP.

4.4 Protocolli EGP

La rete Internet può utilizzare diversi tipi di protocolli EGP. In questo paragrafo descriviamo i principali tipi della classe di protocolli EGP; tra cui uno dei più noti è indicato con la sigla EGP.

Protocollo EGP

Rappresenta il primo protocollo utilizzato in modo esteso all'interno della rete Internet. Questo protocollo, definito nel 1984, è simile ad un protocollo distance vector, ma non utilizza nessuna metrica per caratterizzare la raggiungibilità di una rete; per questo motivo esso è un algoritmo che opera in modo soddisfacente su reti ad albero, ma non su reti a maglia.

La costruzione e l'aggiornamento delle tabelle di routing viene effettuata dal protocollo EGP mediante la generazione di pacchetti di routing che contengono informazioni sulla raggiungibilità di una rete.

Il protocollo EGP presenta numerosi svantaggi e in particolare non ammette la presenza di una topologia a maglia, per cui tutti gli AS devono essere collegati in modo stellare ad un core system.

Protocollo BGP

Border Gateway Protocol (BGP) è un protocollo che dovrebbe sostituire EGP, eliminandone i principali inconvenienti. BGP utilizza un algoritmo di routing del tipo distance vector, in cui al posto del fattore di costo per raggiungere una destinazione viene inviata la sequenza di AS da attraversare per raggiungere tale destinazione. Ogni router definisce una "via" preferita per una fissata destinazione e la comunica ai router BGP adiacenti tramite un distance vector.

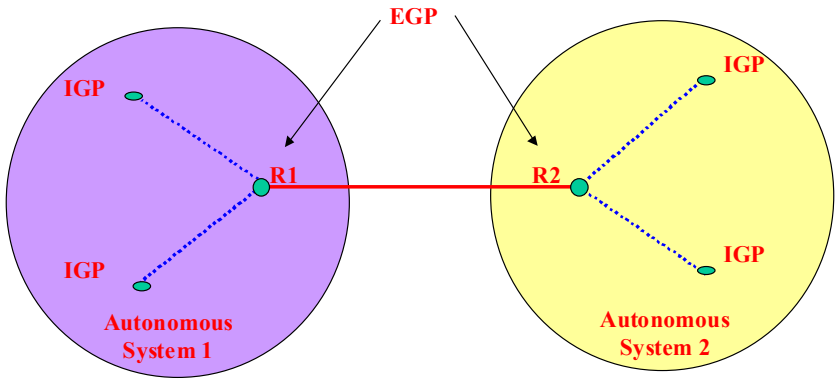

Figura 4.12 – Protocolli di routing.

4.5 Protocolli IGP

I protocolli IGP servono a facilitare lo scambio di informazioni tra router all'interno di un AS. I router all'interno di un AS si scambiano informazioni sulle proprie reti; quando la situazione dell'AS è stata completamente definita, un router (corrisponde all'exsterior router di quel dominio) provvede a informare gli altri AS mediante un protocollo EGP. Come mostrato nella Figura 4.12, un router (ad esempio il router R1 o R2) può utilizzare contemporaneamente due diversi protocolli di routing: un protocollo IGP per comunicare all'interno del proprio AS e un protocollo EGP per comunicare con altri AS.

Descriviamo adesso le caratteristiche dei principali algoritmi IGP utilizzati sulla rete Internet.

Protocollo RIP

RIP (Routing Information Protocol) è il più noto tra i protocolli IGP ed è conosciuto anche con il nome del programma routed. Il successo di questo protocollo è dovuto al fatto che il programma routed veniva incluso nella versione 4BSD di Unix, prodotto dall'università di Berkeley in California, per cui esso è diventato uno standard prima di apparire in un RFC.

Il RIP è un routing protocol di tipo distance vector. Suddivide le macchine in attive e passive. Le macchine attive informano le altre sul routing, quelle passive aggiornano le tabelle di routine, ma non diffondono informazioni sul routing. Tipicamente i **routers** sono **macchine attive** mentre gli **host** sono **macchine passive**.

Protocollo OSPF

Per superare i problemi posti dai protocolli IGP IETF ha costituito nel 1988 un gruppo di studio per la realizzazione di un IGP basato sull'algoritmo SPF. In questo modo è stato definito il protocollo **Open Shortest Path Find (OSPF)**.

Le principali caratteristiche dell'OSPF sono:

- Le **specifiche** dell'OSPF sono **pubbliche**, per cui esso è un protocollo utilizzabile da chiunque in modo aperto senza costi.
- OSPF prevede un routing variabile in base al tipo di servizio; in questo modo possono essere previsti cammini multipli, ciascuno per un tipo di servizio (p.es basso ritardo o alto throughput).

4.6 I protocolli ARP e RARP

L'indirizzo IP assegnato ad un host (indirizzo di livello 3) non corrisponde all'indirizzo fisico (indirizzo di livello 2) della rete a cui l'host è realmente collegato. I protocolli **Address Resulotion Protocol (ARP)** e **Reverse Address Resolution Protocol (RARP)** consentono di rilevare in modo automatico la corrispondenza tra gli indirizzi IP e gli indirizzi fisici.

Il protocollo ARP viene utilizzato tutte le volte che una stazione collegata ad una LAN deve inviare un messaggio ad un nodo sulla stessa LAN di cui conosce solo l'indirizzo IP.

Il protocollo RARP viene utilizzato dalle stazioni senza memoria di massa per scoprire il loro indirizzo IP nella fase iniziale di avviamento. Il computer invia in broadcast una richiesta contenente il proprio indirizzo fisico; il server RARP risponde inviando al computer il suo indirizzo IP. I protocolli ARP e RARP si appoggiano direttamente sulle reti e non su IP.

Il metodo più utilizzato per effettuare la mappatura tra indirizzo IP e fisico consiste nell'inviare un pacchetto speciale all'utente di cui si conosce solo l'indirizzo IP chiedendo di comunicare il proprio indirizzo fisico. Soltanto l'utente in possesso di quell'indirizzo IP può rispondere, comunicando il proprio indirizzo fisico alla stazione che ne aveva fatto richiesta. E' chiaro che questo protocollo di comunicazione in broadcast deve essere ripetuto il meno possibile, e quindi, una volta ottenuta la risoluzione dell'indirizzo, questo è posto in un’apposita memoria di cache, in modo da venire utilizzata successivamente senza dover ripetere tutta la procedura. Per evitare che l'utente a cui è stata indirizzata la richiesta di risolvere il proprio indirizzo debba ripetere tutta questa fase per comunicare con il suo interlocutore, si fa in modo che nel datagram ARP sia contenuta anche l'informazione relativa all'indirizzo fisico di chi sta trasmettendo. Per ottenere ulteriori miglioramenti, quando è in corso la comunicazione broadcast tutte le macchine presenti in quel momento sulla rete ricevono la richiesta di risoluzione e anche se non rispondono possono però prendere nota della risoluzione dell'indirizzo IP in indirizzo fisico del richiedente e memorizzarla in modo da realizzare un aggiornamento su tutte le macchine.

Per lo stesso motivo, si può prevedere che ogni macchina che fa un boot sulla rete invii un messaggio broadcast ARP per presentarsi con il proprio indirizzo fisico a tutte le altre

macchine presenti. A questo punto, se una macchina ha nella propria cache l'indirizzo IP del nuovo arrivato, controlla che i due indirizzi fisici coincidano, ed in caso contrario aggiorna il contenuto della cache.

Prendiamo come riferimento le reti **Ethernet**. Nelle reti Ethernet, l'indirizzo fisico è associato all'**interfaccia hardware** (muovendo la scheda su un altro computer, l'indirizzo contenuto nella scheda viene associato al nuovo computer). Gli indirizzi Ethernet sono fissati a livello internazionale: ogni costruttore di schede ha a disposizione un proprio set di indirizzi. L'ARP nasconde la rete sottostante ai protocolli superiori, consentendo a questi di usare gli indirizzi IP.

Formato del pacchetto ARP

Nella Figura 4.13 è mostrato il formato del pacchetto ARP. Vediamo in dettaglio il significato dei principali campi del pacchetto.

- Hardware type : indica il tipo di interfaccia hardware per cui il trasmettitore aspetta una risposta;
- Protocol type: specifica il tipo di protocollo utilizzato dal trasmettitore;
- HLEN: specifica la lunghezza dell'indirizzo hardware in modo da poter utilizzare diversi tipi di rete;
- PLEN: specifica la lunghezza dell'indirizzo IP utilizzato;
- Operation: specifica il tipo di operazione effettuata, cioè se è una richiesta ARP, una risposta RARP, una richiesta RARP oppure una risposta RARP;
- HA1: indica l'indirizzo hardware dell'apparecchiatura che ha inviato la richiesta;
- IP1: indica l'indirizzo IP dell'apparecchiatura che ha inviato la richiesta;
- HA2: indica l'indirizzo hardware dell'apparecchiatura a cui è inviata la richiesta;
- IP2: indica l'indirizzo IP dell'apparecchiatura a cui è inviata la richiesta.

Protocollo ICMP

Il protocollo **ICMP (Internet Control Message Protocol)** permette ad un router di inviare messaggi di errore o di controllo ad altri router o host. ICMP fornisce un metodo per la comunicazione tra il protocollo IP su una macchina e il protocollo IP su un'altra macchina ed è considerato parte essenziale del protocollo IP, per cui deve essere presente in ogni sua implementazione. I messaggi ICMP viaggiano attraverso la rete nella parte dati del datagram.

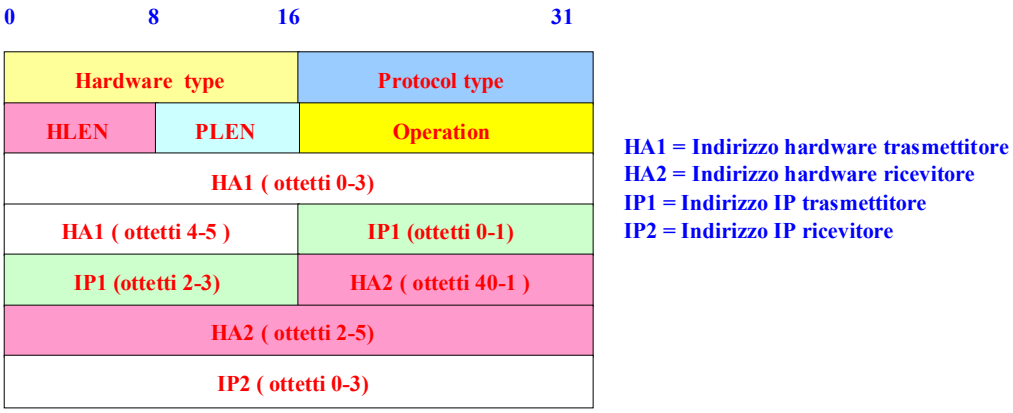

Figura 4.13 – Formato del pacchetto ARP.

Nonostante le linee guida raccomandino l'uso di alcuni messaggi ai soli router, in generale anche gli host possono scambiare messaggi ICMP; in questo modo è possibile utilizzare un solo meccanismo per tutti i messaggi di controllo.

ICMP serve solo a segnalare al mittente di un datagram la presenza di un errore, ma non fornisce nessuna informazione sulla loro natura, per cui è compito del mittente individuare le cause di errore e intraprendere le azioni per correggerle.

Le cause di errore possono nascere in vari punti del sistema di trasmissione, ma ICMP può segnalare la condizione di errore soltanto al mittente, poiché il datagram ricevuto contiene l'indirizzo del mittente e non l'indicazione del percorso effettuato.

Ogni messaggio ICMP è inserito nella parte dati del datagram IP, che viene a sua volta trasportato da un frame, come mostrato nella Figura 4.14. I datagram che trasportano messaggi ICMP sono instradati come gli altri datagram nella rete.

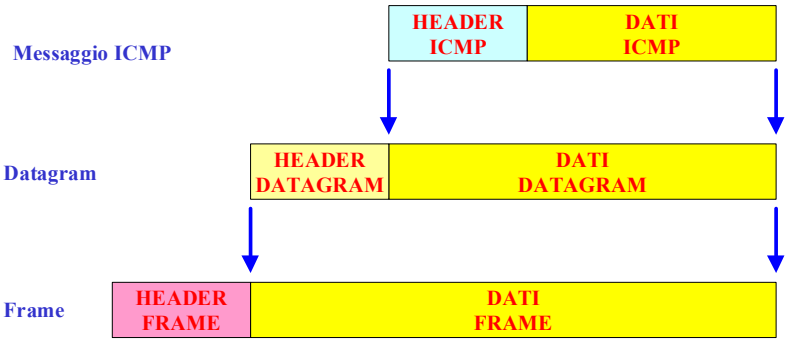

Figura 4.14 – Formato del datagram ICMP.

Messaggi ICMP

Il protocollo ICMP può inviare una serie di messaggi diversi, come mostrato nella tabella 4.3. Ogni messaggio ha un formato proprio. Descriviamo brevemente alcuni messaggi ICMP:

- Echo request e echo reply
  Attraverso questo messaggio una macchina A invia un echo request ad un'altra macchina B; quest'ultima risponde con un messaggio echo reply diretto alla macchina A. L'utilizzo più diffuso di questa procedura è legato all'applicazione **ping**, che serve a verificare se una stazione con cui ci si vuole collegare è raggiungibile ed attiva.
- Unreachable destination
  Questo messaggio viene inviato al mittente nel caso in cui un datagram non possa essere consegnato.
- Source quench
  Questo messaggio viene inviato per risolvere i problemi di congestione che si stanno verificando nella rete (ad esempio in un router). La condizione di congestione è quella in cui i datagram arrivano in un nodo ad un ritmo più elevato di quanto il nodo stesso riesca a trasmetterli. Un router invia un messaggio di source quench per ogni datagram scartato. Alcuni router utilizzano tecniche più sofisticate, monitorando i datagram ed inviando i source quench alle sorgenti con traffico più elevato. Una sorgente che riceve un messaggio di source quench deve ridurre la velocità di trasmissione fino a quando non cessa di ricevere tali messaggi.
- Redirect
  Questo messaggio può essere utilizzato tra router e host che si trovino sulla stessa rete e serve ad ottimizzare l'instradamento dei messaggi.
- Time exceed
  Quando un router invia un datagram, un contatore detto "time-to-live" viene messo ad un valore prefissato. Il contatore viene decrementato con il passare del tempo e quando raggiunge il valore 0 il datagram è scartato e un messaggio di time exceed è inviato al mittente del datagram. Questo meccanismo serve ad evitare ritardi troppo elevati nella consegna di un datagram e il fenomeno chiamato "routing cycle", in cui due o più ruoter si inviano un datagram in circolo.
- Parameter problem
  Questo messaggio viene inviato quando il datagram ricevuto contiene gravi errori; in particolare, quando l'header del datagram non è corretto o non sono corretti i parametri relativi ad un'opzione contenuta in qualche campo dell'header.
- Timestamp
  Serve per sincronizzare gli orologi di più macchine.
- Address mask request e reply
  Servono per ottenere una subnet mask.

TIPO DI CAMPO	TIPO DI MESSAGGIO ICMP
0	Echo Replay
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
11	Time Exceeded for a Datagram
12	Parameter on a Datagram
13	Timestamp Request
14	Timestamp Reply
15	Information Request (obsolete)
16	Information Reply (obsolete)
17	Address Mask Request
18	Address Mask Reply

Tabella 4.3 - Tipi di messaggio ICMP.

4.7 Protocollo IPv6

La crescita enorme subita da Internet avvenuta a partire dai primi anni '90 ha posto in evidenza le limitazioni del sistema IPv4 attualmente utilizzato. Le previsioni indicano che lo spazio di indirizzamento disponibile con IPv4, in cui gli indirizzi IP sono composti da 32bit, sarà esaurito nel 2010. Per questo motivo *IETF* (*Intenet Engineering Task Force*) ha elaborato nel 1994 un nuovo protocollo IP, noto con il nome di **IPv6**, descritto in **RCF 1752** "Raccomandazioni per il protocollo IP di prossima generazione". IPv6 introduce numerose modifiche rispetto a IPv4 e in particolare presenta indirizzi IP composti da 128 bit. Il numero di indirizzi in IPv6 è molto alto e seguendo la creazione di una gerarchia di indirizzi IPv6 può offrire 1500 indirizzi per ogni metro quadrato del pianeta. Questo numero così alto di indirizzi presenta numerosi vantaggi, quali:

- semplifica le procedure di instradamento;
- aumenta l'efficacia delle tabelle di instradamento;
- permette l'attuazione corretta delle procedure di configurazione automatica;
- consente di risolvere il problema degli indirizzi Internet per molti anni.

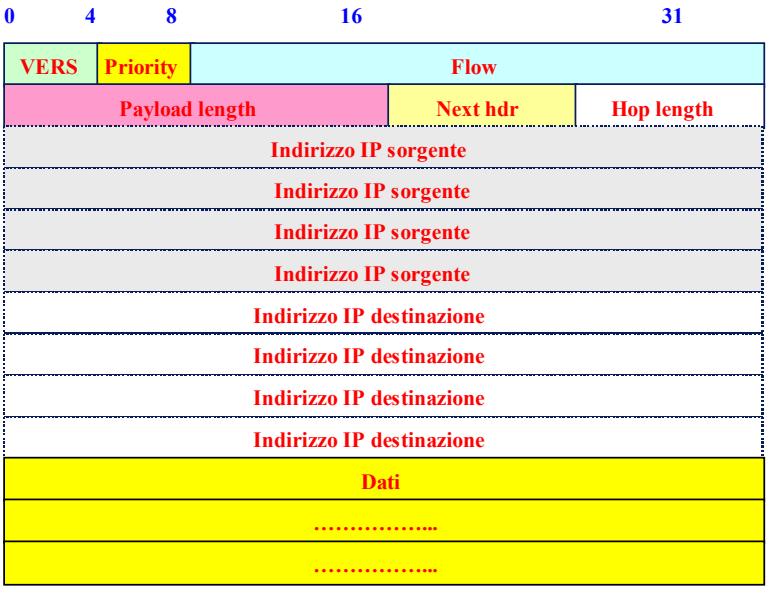

Figura 4.15 – Formato del datagram IPv6.

L'intestazione del datagram nel protocollo IPv6 è diversa rispetto a IPv4 e presenta le seguenti caratteristiche:

- il numero dei campi dell'header è ridotto da 10 a 6;
- la lunghezza dell'header in IPv6 è fissa; in questo modo l'elaborazione del pacchetto è più semplice e veloce;
- una parte dello spazio previsto per l'indirizzo in IPv6 è riservata dagli indirizzi IPv4. Inoltre è possibile scrivere gli indirizzi IPv4 come indirizzi compatibili IPv6. In questo modo il passaggio da IPv4 a IPv6 è notevolmente semplificato.

Il formato del pacchetto Ipv6 è mostrato nella Figura 4.15; vediamo in dettaglio la funzionalità dei diversi campi del pacchetto IPv6.

Vers
Il campo formato da 4 bit, indica il numero della versione del protocollo Internet. Nel caso di IPv6 questo campo assume il valore 6.

Priorità
Il campo è formato da 4 bit ed indica la priorità dei diversi pacchetti.

Flow label
Questo campo, formato da 28 bit, può essere utilizzato da un host per caratterizzare la qualità di servizio per ciascun pacchetto (ad esempio pacchetti da trasmettere in tempo reale).

Payload length
Questo campo, formato da 16 bit, indica la lunghezza in byte dei dati (payload), che seguono l'intestazione.

Next header
Questo campo, formato da 8 bit, identifica il tipo di intestazione che segue quella dell'IPv6 ed usa gli stessi valori del campo analogo di IPv4.

Hop limit
Il campo, formato da 8 bit, serve a limitare il numero di nodi attraversati per evitare problemi di "loop" o ritardi troppo elevati. Il valore del campo è inizializzato a 255 quando il pacchetto viene trasmesso e il suo valore è diminuito di 1 ogni volta che il pacchetto attraversa un nodo. Quando il campo assume il valore 0, il pacchetto viene scartato.

Indirizzo di sorgente
Il campo, formato da 128 bit, rappresenta l'indirizzo IP della sorgente che ha generato il pacchetto.

Indirizzo di destinazione
Il campo, formato da 128 bit, rappresenta l'indirizzo IP della stazione di destinazione.

Configurazione automatica di un indirizzo IP

Una caratteristica molto interessante di IPv6 è rappresentata dalla sua capacità di configurare in modo automatico un host con un indirizzo IP: in altre parole un host può acquisire automaticamente l'informazione necessaria per la definizione del suo indirizzo IP.

Questa operazione, che consente di ridurre notevolmente la complessità ed il costo di gestione delle grandi reti e di configurare in modo semplice le piccole reti, può essere fatta con due modalità diverse:

- se l'host è collegato ad un router o a un server di indirizzi locali (ad esempio, in una rete locale), esso rivela l'indirizzo della rete su cui si trova, aggiunge una parte relativa al proprio indirizzo e invia un messaggio per verificare se esiste un altro indirizzo uguale. Se non esistono indirizzi uguali, il router o il server degli indirizzi viene aggiornato con questo nuovo indirizzo.
- se l'host è collegato ad una rete di grandi dimensioni, l'host invierà una richiesta ad un server di indirizzi, che provvederà a fornire l'indirizzo.

Coesistenza e transizione da IPv4 a IPv6

Con l'introduzione di IPv6 sarà necessario garantire la convivenza tra IPv4 e IPv6 per un lungo periodo di tempo. La prima soluzione è quella di realizzare in un nodo ambedue gli algoritmi IPv4 e IPv6; questa soluzione, indicata con il nome **dual stack**, è mostrata nella Figura 4.16. I nodi di questo tipo sono indicati come IPv6 / IPv4.

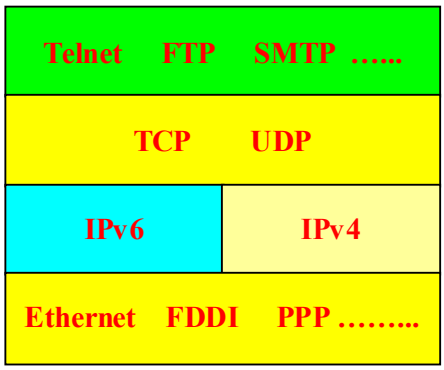

Figura 4.16 – Nodo dual stack IPV6/IPv4.

Un altro problema è rappresentato dal trasporto di un datagram IPv6 attraverso una rete IPv4; come mostrato nella Figura 4.17 in cui due nodi IPv6/IPv4 sono collegati attraverso una rete IPv4. Per risolvere questo problema si può utilizzare una tecnica di **tunneling**, in cui il datagram IPv6 viene incapsulato in un datagram IPv4 mediante l'inserimento di un header IPv4 (Figura 4.18).

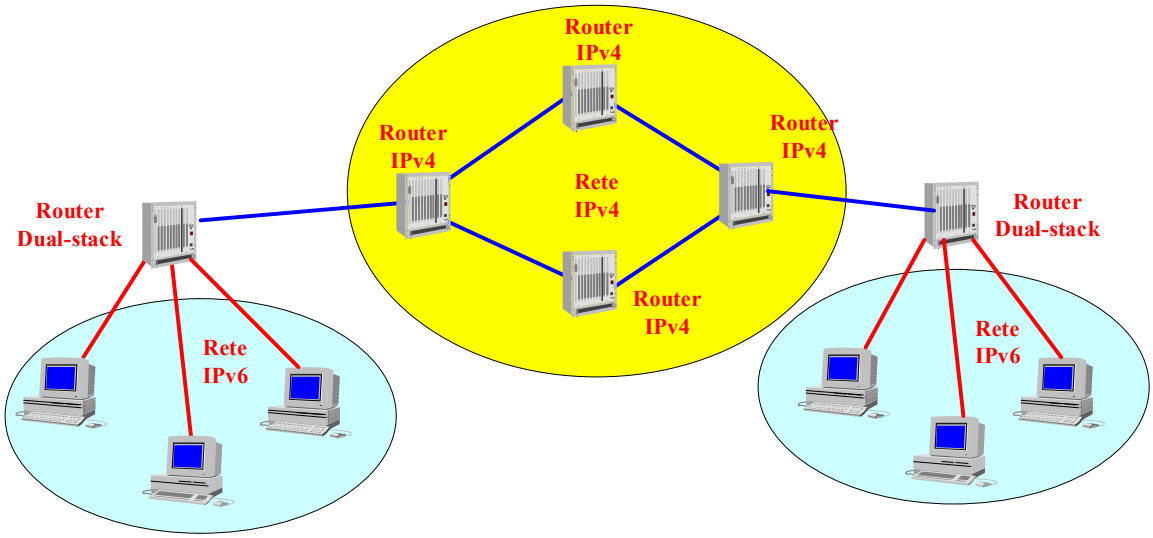

Figura 4.17. Collegamento tra due reti IPv6 mediante una rete IPv4.

Gestione della multimedialità e di applicazioni in tempo reale

Il protocollo IPv6 consente una gestione migliore delle applicazioni multimediali e di quelle in tempo reale, quali videoconferenza o audio conferenza. Il campo priority, formato da 4 bit, classifica i pacchetti secondo la loro importanza per assicurare che, in caso di deterioramento della qualità della rete, i dati ad alta priorità arrivino a destinazione. Ad esempio, nel caso di una videoconferenza, l'audio è considerato più importante del video ed il video a bassa definizione avrà una priorità maggiore del video ad alta definizione.

Sicurezza

Il protocollo IPv6 consente di avere un maggiore livello di sicurezza rispetto alle versioni precedenti; in particolare IPv6 considera la sicurezza della comunicazione, cioè:

- la convalida, che consente al destinatario di essere certo dell'origine di un messaggio;
- la crittografia, che rende il messaggio illeggibile da chiunque non sia il destinatario.

L'intestazione di convalida protegge le reti contro i rischi di instradamento di sorgente e di attacchi contro l'host, mentre la crittografia garantisce la riservatezza delle informazioni.

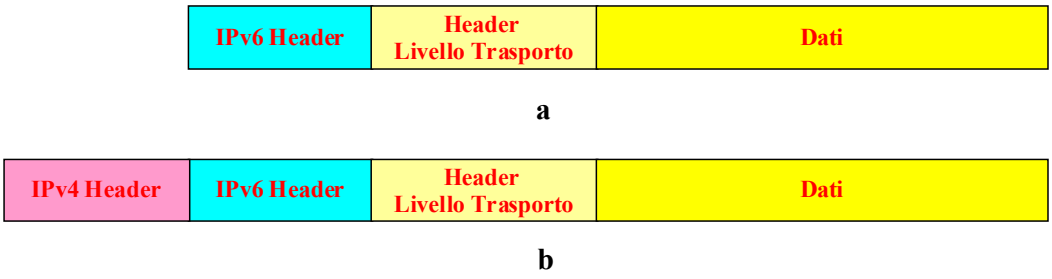

Figura 4.18 – Tecnica di tuneling: a) datagram IPv6; b) incapsulamento di un datagram IPv6 in un datagram IPv4.

4.8 Protocolli TCP e UDP

TCP (Transmission Control Protocol) e **UDP (User Datagram Protocol)** sono protocolli utilizzati nel livello di trasporto ed operano perciò a livello superiore rispetto a IP. Essi hanno caratteristiche diverse: il protocollo TCP garantisce un trasferimento affidabile dell'informazione, mentre UDP non garantisce l'affidabilità dei dati inviati ai livelli superiori.

I due protocolli sono in grado di trasferire i datagram IP tra computer utilizzando l'indirizzo IP. Tuttavia, un indirizzo IP identifica un computer e non è possibile effettuare distinzioni tra datagram diretti ad applicazioni sullo stesso computer. Il protocollo UDP aggiunge un meccanismo che consente di separare i datagram ricevuti da un computer tra le diverse applicazioni. In questo modo il protocollo di trasporto aggiunge alle funzioni dell' IP un meccanismo che permette alle singole applicazioni che sono eseguite su un computer di inviare e ricevere i datagram in modo indipendente l'una dall'altra.

Infatti, i sistemi operativi moderni supportano la multiprogrammazione, cioè permettono a più applicazioni di essere eseguite contemporaneamente. Ciascuna applicazione prende il nome di **processo** o di **task**. Per inviare un datagram ricevuto al processo a cui è destinato, UDP introduce un'astrazione che prende il nome di **porta**. Ogni porta è individuata da un numero intero positivo. Il sistema operativo locale fornisce un meccanismo di interfaccia che i processi utilizzano per specificare una porta o per accedervi. Generalmente il sistema operativo fornisce un accesso **sincrono** alle porte, nel senso che avvengono delle interruzioni di calcolo durante un'operazione di accesso ad una porta. Il protocollo UDP garantisce inoltre un processo di bufferizzazione dei pacchetti

per ciascuna porta. In questo modo i pacchetti in arrivo sono memorizzati in una coda di dimensioni finite, fino quando il processo non li estrae.

Per poter comunicare con una determinata porta, il mittente deve conoscere sia l'indirizzo IP della macchina destinazione che il numero della porta nella macchina destinazione; inoltre ciascun pacchetto contiene sia il numero della porta mittente che quella della porta destinazione. In questo modo è possibile per qualsiasi processo che riceve un messaggio di replicare al mittente.

Il protocollo TCP implementa il meccanismo delle porte, ma in modo diverso rispetto all'UDP. Infatti, nell'UDP una porta è considerata come un singolo oggetto, a cui può essere associata una coda. Nel TCP una porta non è un singolo oggetto e la connessione diventa l'astrazione principale. Una connessione è formata da una coppia di **endpoints**. Un endpoint è una coppia di interi (host, port) dove host è l'indirizzo IP di una macchina e port è la porta TCP su quella macchina. Dunque il TCP identifica una connessione con una coppia di endpoints. La conseguenza di ciò è che un dato numero di porta TCP su una macchina può essere condiviso da più connessioni.

Esempio
L'endpoint (150.217.11.23,25) rappresenta l'indirizzo IP 150.217.11.23 e la porta 25. Per rappresentare l'endpoint si può usare anche la notazione (150.217.11.23: 25).

Ad esempio un programma di gestione della posta elettronica su un server ha bisogno di una sola porta TCP, anche se consente l'accesso concorrente ai suoi servizi da parte di più utenti connessi in remoto.

Il protocollo TCP

Il TCP è un protocollo di livello trasporto. Anche se è stato introdotto come parte del TCP/IP, il TCP è di uso generale; infatti è stato preso come base per il protocollo TP-4 proposto dall'ISO nell'ambito del modello OSI.

A livello di collegamento il protocollo IP fornisce un servizio inaffidabile di distribuzione dei pacchetti. Infatti i pacchetti possono andare perduti od essere distrutti per una serie di cause. Le applicazioni, che si trovano ai livelli superiori, richiedono che il trasferimento dei dati sia affidabile. Nel caso della rete Internet il protocollo TCP assicura questo risultato.

Le principali caratteristiche del protocollo TCP sono:

- è orientato alla connessione, con conferma e controllo del flusso;
- fornisce un servizio full-duplex;
- utilizza una commutazione a circuiti virtuali.

Il protocollo TCP fornisce i seguenti servizi:

- definisce il formato dei dati e degli acknowledgement che due computer si devono scambiare per avere una trasmissione affidabile;
- definisce le procedure per assicurare la correttezza dei dati;
- implementa un meccanismo per distinguere le destinazioni multiple all'interno di una stessa macchina;

- implementa meccanismi per gestire la perdita e la duplicazione di pacchetti;
- specifica il modo in cui due computer iniziano la trasmissione della sequenza di bit e come questi si accordano sulla fine della trasmissione.

Il TCP non impone particolari limitazioni alla rete sottostante, per cui funziona su molte reti, quali collegamenti telefonici commutati, reti locali, reti ad alta velocità in fibra ottica, collegamenti punto-punto a bassa velocità, ecc.

Formato del segmento TCP

L'unità informativa del TCP prende il nome di segmento. I segmenti vengono scambiati tra due computer per:

- stabilire la connessione;
- trasferire i dati;
- inviare gli ACK;
- inviare i window advertisement;
- chiudere la connessione.

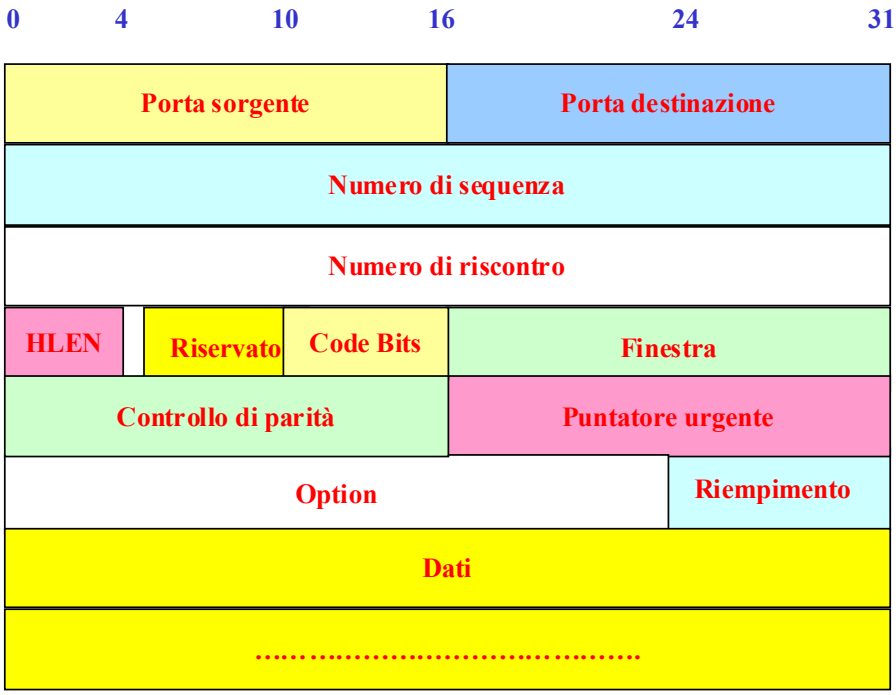

Figura 4.19 - Formato del segmento TCP.

Il formato di un segmento TCP è diviso in due campi, l'header e il campo dati ed è mostrato nella Figura 4.19. Descriviamo in dettaglio i diversi campi:

- porta sorgente: numero della porta TCP a cui sono associati gli applicativi;
- porta destinazione: numero della porta TCP di destinazione a cui sono associati gli applicativi;

- numero sequenza: numero di sequenza (posizione) del primo byte del campo dati del messaggio. Viene utilizzato anche come identificatore della "*sliding window*".
- numero di riscontro: contiene il numero di sequenza del primo byte che la stazione sorgente si aspetta di vedere confermato. Rappresenta perciò il campo di acknowledgment con tecnica di piggybacking della trasmissione nella direzione apposta.
- HLEN: indica la lunghezza dell'header TCP, misurata in multipli di 32 bit; questo valore è necessario perché il campo option ha una lunghezza variabile.
- Riservato: indica un campo riservato ad applicazioni future.
- Code bit: contiene informazioni sulle caratteristiche e finalità del segmento.
- Finestra : contiene la dimensione della finestra di ricezione TCP in trasmissione e quindi le dimensioni del buffer per il traffico in ingresso.
- Controllo di parità: viene utilizzato per rivelare errori sull'header.
- Puntatore urgente: indica il primo byte urgente del pacchetto.
- Option: serve per negoziare con il TCP dell'altra macchina vari parametri, quali la massima lunghezza di un segmento.
- Riempimento: è un campo fittizio la cui lunghezza dipende da quella del campo opzioni. La somma delle lunghezze dei due campi deve essere uguale a 32 bit.

Meccanismo per il controllo degli errori

Il protocollo TCP garantisce l'affidabilità dell'informazione ricevuta mediante l'utilizzo di una appropriata tecnica ARQ. Esso utilizza una tecnica **sliding window** (finestra a scorrimento) per regolare le trasmissioni e le ritrasmissioni. La ricezione dei dati viene confermata dalla stazione ricevente; tale conferma può essere inviata in modo separato oppure essere inserita in un datagram in transito in direzione opposta con una tecnica di piggybacking. Il meccanismo a finestra del TCP risolve il problema di una trasmissione efficiente e del controllo di flusso. Il TCP vede la sequenza di bit da trasmettere come una sequenza di byte che viene trasmessa in gruppi detti **segmenti**. In genere ciascun segmento viaggia in Internet in un singolo datagram IP.

Come tutti i protocolli a finestra, TCP ha un numero massimo di dati in attesa di conferma; nel TCP tale dimensione massima è specificata come numero di byte e non come segmenti. I byte di una sequenza sono numerati e la stazione trasmittente utilizza tre puntatori per ciascuna connessione che svolgono le seguenti funzioni:

- il primo puntatore C1 separa, sulla sua destra, i byte che sono stati trasmessi e che hanno ricevuto un ACK;
- il secondo C2 individua l'ultimo byte che può essere trasmesso prima che venga ricevuto un ACK;
- il terzo C3 separa, all'interno della finestra, i byte che sono stati trasmessi da quelli da trasmettere.

Nella Figura 4.20 è mostrato un esempio di applicazione della finestra scorrevole del TCP. Nell'esempio i primi due byte sono stati trasmessi ed hanno ricevuto un riscontro positivo. I byte tra C1 e C2 (cioè i byte 3, 4 5,6) sono stati spediti e sono in attesa di riscontro mentre i byte tra C2 e C3 (cioè i byte 7, 8, e 9) devono essere ancora trasmessi, ma possono essere trasmessi anche senza ricevere un ACK. I bit successivi al contatore C3 (nell'esempio i byte dal 10 in poi) non possono essere trasmessi fino a quando non sarà ricevuto un ACK, che faccia scorrere la finestra verso sinistra.

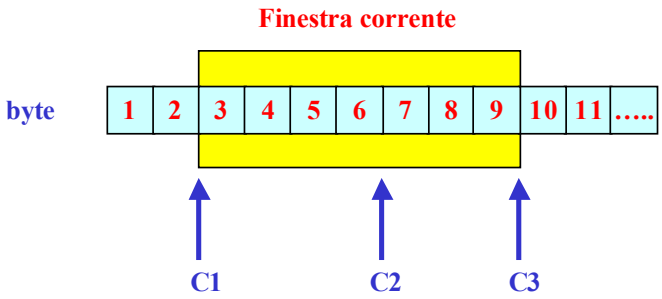

Figura 4.20 – Meccanismo a finestra utilizzato nel TCP per il controllo degli errori.

Il TCP risolve il problema del controllo di flusso rendendo variabili le dimensioni della finestra. Ciascun ACK specifica anche il **window advertisement**, ovvero quanti byte addizionali saranno accettati dal destinatario. Se questo numero aumenta, il mittente aumenta la dimensione della finestra; se diminuisce, il mittente diminuisce la dimensione della finestra. In casi estremi, il destinatario può inviare un window advertisement di zero per bloccare completamente la trasmissione.

Il protocollo UDP

Il protocollo UDP è un protocollo alternativo al TCP. UDP risulta molto più semplice del TCP, ma, contrariamente al TCP, non garantisce l'affidabilità dei dati ricevuti. UDP utilizza IP per la consegna dei pacchetti e, rispetto al livello IP, fornisce soltanto la possibilità di distinguere, attraverso il meccanismo delle porte, tra destinazioni multiple all'interno della stessa macchina. In particolare:

- non usa tecniche ARQ
- non ordina i pacchetti giunti a destinazione
- non fornisce feedback per controllare la velocità con cui scorre l'informazione tra le macchine

Per questo motivo i pacchetti UDP possono essere persi o duplicati e possono arrivare fuori ordine. Inoltre i pacchetti UDP possono arrivare più velocemente di quanto il destinatario riesca a processarli. Tutti questi aspetti devono dunque essere gestiti dai processi a livello superiore.

Spesso le applicazioni che si poggiano su un UDP funzionano bene in ambito locale, su reti affidabili e ad alta velocità, mentre non funzionano correttamente sulle reti geografiche.

Formato del pacchetto UDP

Il pacchetto UDP è chiamato **user datagram (datagram di utente)** ed è mostrato nella Figura 21. L'header del pacchetto UDP è costituito dai seguenti **4 campi**, ciascuno lungo 16 bit:

- Porta sorgente UDP: questo campo è opzionale. Quando è presente indica la porta a cui devono essere inviate eventuali repliche. Se non è utilizzati viene messo uguale a 0.
- Porta di destinazione UDP: indica il numero della porta di destinazione.
- Lunghezza messaggio UDP: indica la lunghezza in byte del datagram UDP, incluso l'header e i dati. Il valore minimo è 8, corrispondente alla lunghezza della sola intestazione.
- Somma di parità UDP: questo campo è opzionale. Se non è usato, deve essere 0. Considerando che il protocollo IP non calcola il checksum sui dati (ma solo sull'header), questo è l'unico elemento che indichi se i dati siano giunti a destinazione in modo corretto. Il cheksum viene calcolato dividendo i dati a gruppi di 16 bit e prendendo il complemento a 1 della somma dei complementi a 1.

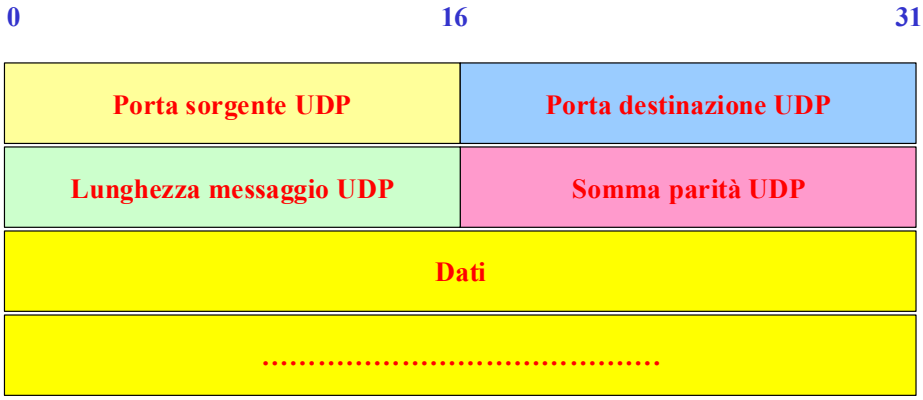

Figura 4.21 - Formato del datagram di utente nel protocollo UDP.

La somma di parità UDP consente anche di verificare che il datagram UDP ha raggiunto la destinazione corretta e viene effettuata utilizzando il datagram ricevuto e un pseudo-header, che non viene trasmesso, ma che il ricevitore è in grado di ricostruire dalle informazioni contenute nel datagram. L'UDP antepone una pseudo-intestazione al datagram UDP, aggiunge 8 bit uguali a zero per riempire il datagram in modo che il suo contenuto sia multiplo intero di 16, ed esegue la somma di verifica sull'intero oggetto (Figura 4.22). Il byte utilizzato per il riempimento e la pseudo-intestazione non sono trasmessi con il datagram UDP, e nemmeno vengono presi in considerazione per il calcolo della lunghezza nell'header UDP.

Per calcolare la somma di verifica, il software mette tutto a zero il campo della somma di verifica, e dopo aggiunge una somma in complemento ad uno dell'intero oggetto, compresi la pseudo-intestazione, l'intestazione dell'UDP e dei dati di utente. La chiave di interpretazione della pseudo-intestazione risiede nella comprensione che la destinazione è costituita da una macchina specifica e da una specifica porta di protocollo, all'interno della stessa macchina. L'intestazione dell'UDP specifica di per sé stessa soltanto il numero della porta di protocollo. In questo modo, per verificare la destinazione, l'UDP presente nella macchina mittente calcola una somma di verifica che comprende l'indirizzo IP di destinazione insieme al datagram UDP. All'arrivo alla destinazione finale, il software di UDP controlla la somma di verifica che comprende l'indirizzo IP di destinazione ricavato dall'intestazione del datagram IP che viene portato all'interno del messaggio UDP. Se la somma di verifica coincide, si può asserire che il datagram ha raggiunto l'host di destinazione voluto ma anche la corretta porta di protocollo all'interno dello stesso host.

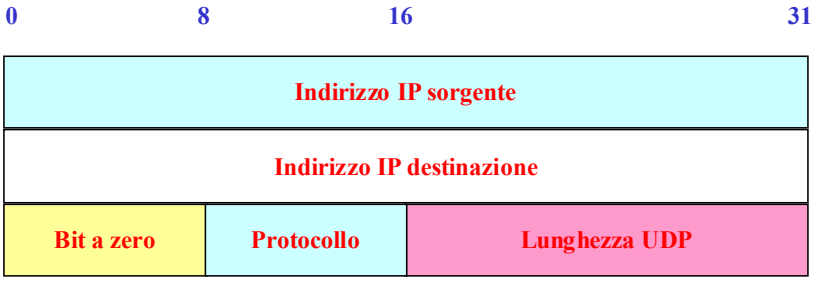

Figura 4.22. Formato del pseudo-header UDP.

Lo scopo dello pseudo-header è quello di verificare che l'UDP datagram abbia raggiunto la corretta destinazione, ovvero una specifica macchina e una specifica porta all'interno di quella macchina. Alla destinazione, il software deve ricostruire lo pseudo-header estraendo l'indirizzo IP di destinazione e deve ricalcolare il checksum. Lo pseudo-header consiste in **12 byte** in cui sono presenti:

- Indirizzo IP della sorgente (32 bit);
- Indirizzo IP di destinazione (32 bit);
- Bit uguali a zer (8 bit);
- Tipo di protocollo (8 bit);
- Lunghezza UDP (16 bit).

Il calcolo del checksum nel protocollo UDP rappresenta una **violazione** del modello ISO/OSI, che si basa sulla separazione delle funzioni di ciascun strato. Infatti, per il calcolo del checksum, il protocollo UDP deve interagire con il livello IP sottostante, per farsi fornire l'indirizzo IP destinazione, ma soprattutto l'indirizzo IP sorgente, che non può essere noto a priori dai livelli superiori all'IP, in quanto dipende dalla scelta di routing.

Il protocollo UDP, attraverso il meccanismo delle porte, implementa la funzione multiplexing e demultiplexing, tipica dei protocollo del livello di trasporto (Figura 4.23).

Figura 4.23 – Formato del datagram utilizzato per il calcolo del checksum nel protocollo UDP.

Nella pratica, ciascuna applicazione deve negoziare con il S.O. l'utilizzo di una porta prima di poter inviare UDP datagrams. Spesso in questa fase di negoziazione, il S.O. crea la coda associata a quella porta; l'applicazione può specificare o modificare la dimensione della coda.