PER ALTRI APPUNTI CONSULTARE IL SITO:

https://luigi-v.github.io/Appunti_Universita/

Per altri appunti consultare il sito: *https://luigi-v.github.io/Appunti_Universita/*

By *Vollono Luigi*

CRC (controllo ciclico di ridondanza)

la codifica polinomiale, nota anche come CRC (controllo ciclico di ridondanza). Le codifiche polinomiali sono basate sul fatto di trattare le sequenze di bit come dei polinomi a coefficienti che possono assumere solo i valori 0 oppure 1. Un frame di k bit è visto come una lista di coefficienti per un polinomio con k termini che variano da x^k-1 a x^0. Tale polinomio è detto di grado k-1. Il coefficiente di termine più alto (quello più a sinistra) è il coefficiente per x^k-1; il successivo è per x^k-2 e così via. Per esempio 110001 ha 6 bit e quindi rappresenta un polinomio di 5° grado con coefficienti 1, 1, 0, 0, 0 e 1 : x^5 + x^4 + x^0.

L'aritmetica dei polinomi si gestisce in modulo 2 secondo le regole della teoria dei campi algebrici. Non ci sono riporti per le addizioni o prestiti per le sottrazioni. Sia l'addizione sia la sottrazione sono identici all'OR esclusivo. Per esempio:

10011011	00110011	11110000	01010101
+ 11001010	+ 11001101	− 10100110	− 10101111
01010001	11111110	01010110	11111010

Le divisioni lunghe vengono eseguite come in binario, salvo che le sottrazioni sono in modulo 2 come spiegato sopra. Si dice che un divisore "sta" nel dividendo se il dividendo ha tanti bit quanti il divisore.

Quando si utilizza una codifica polinomiale, la sorgente e la destinazione devono mettersi d'accordo in anticipo su un polinomio generatore, G(x). Il generatore deve avere i bit di ordine più alto e più basso uguali a 1. Per poter calcolare il checksum di un frame di m bit che corrisponde al polinomio M(x), il frame deve essere più lungo del polinomio generatore. L'idea è quella di aggiungere un checksum alla fine del frame in modo che il polinomio rappresentato dal frame con checksum sia divisibile per G(x). Quando la destinazione riceve il frame con checksum prova a dividerlo per G(x). Se c'è un resto vuol dire che c'è stato un errore di trasmissione. L'algoritmo per calcolare il checksum è il seguente:

1. posto r il grado di G(x), aggiungere r bit con valore zero dopo la parte di ordine più basso del frame, così che adesso contenga m + r bit e corrisponda al polinomio x^r M(x)

2. dividere la sequenza di bit corrispondente a G(x) per la sequenza corrispondente a x^r M(x) usando la divisione modulo 2. [NdR - il dividendo è x^r M(x) mentre il divisore è G(x)]

3. sottrarre il resto (che contiene sempre al massimo r bit) dalla sequenza corrispondente a x^r M(x) usando la sottrazione in modulo 2. Il risultato è il frame con checksum pronto per la trasmissione. Chiamiamolo polinomio T(x).

La Figura 3.8 illustra il calcolo per il frame 1101011011 usando il generatore G(x) = x^4 + x+ 1.

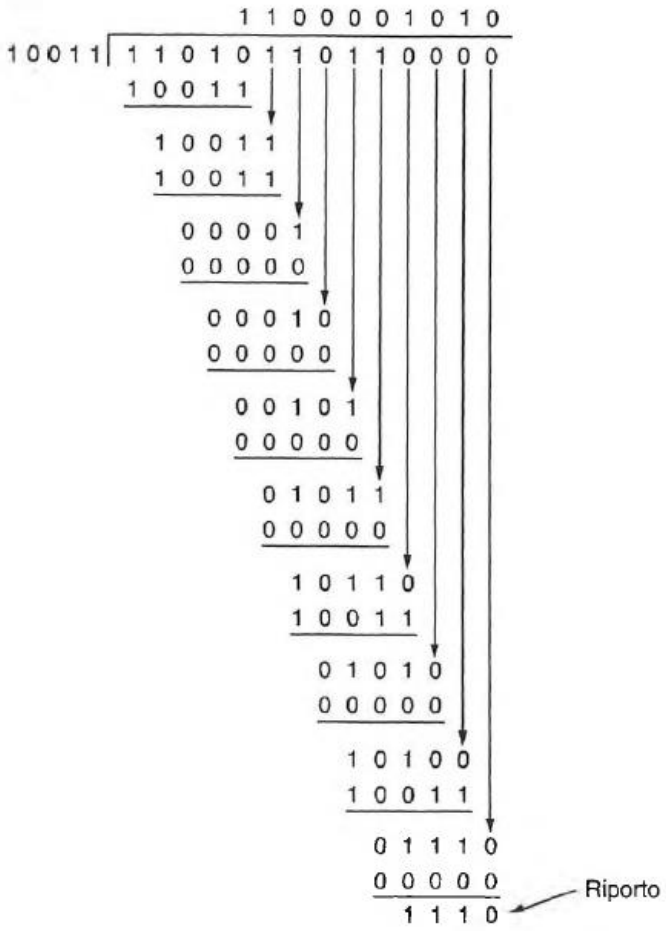
Dovrebbe essere chiaro che T(x) è divisibile (modulo 2) per G(x). In ogni divisione, se si sottrae il resto dal dividendo, quello che resta è divisibile per il divisore. Vediamo un esempio: in base 10, se dividiamo 210.278 per 10.941 il resto è 2.399. Sottraendo 2.399 da 210.278, quello che rimane (207.879) è divisibile per 10.941.

Analizziamo la potenza di questo metodo. Quali tipologie di errori potrà rilevare? Immaginiamo che accada un errore in modo che al posto della sequenza di bit T(x) arrivi una sequenza T(x) + E(x). Ogni bit a 1 in E(x) corrisponde a un bit che è stato invertito. Se ci sono k bit a 1 in E(x), vuol dire che ci sono stati k errori su singoli bit. Un singolo burst di errori è caratterizzato da un 1 iniziale, una miscela di 0 e 1 e un 1 finale, mentre tutti gli altri bit sono a 0.

Frame : 1 1 0 1 0 1 1 0 1 1

Generatore: 1 0 0 1 1

Messaggio dopo l'aggiunta di quattro bit a zero: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Frame trasmesso: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Figura 3.8. Calcolo del checksum a codifica polinomiale.

DATA LINK (Collegamento dati)

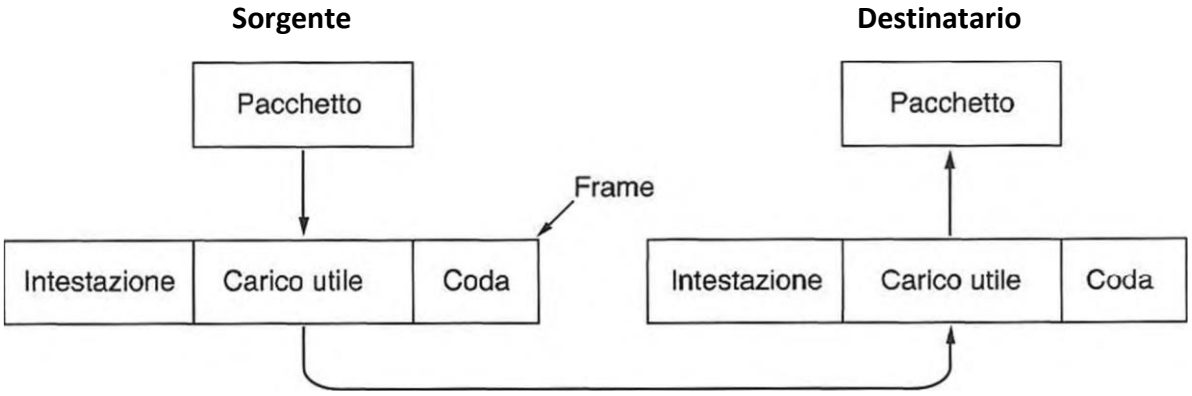
Il suo compito è quello di ***organizzare*** il trasferimento dei dati tra ***due apparati adiacenti, logicamente connessi*** da un ***canale***, e di fornire una interfaccia definita per consentire allo strato di rete di ***accedere*** ai ***servizi*** offerti. Si prende carico di diverse funzioni fra cui:

1. fornire un ben definito servizio d'interfaccia per lo strato network

2. gestire gli errori di trasmissione

3. regolare il flusso dati in modo che i dispositivi riceventi lenti non vengano sopraffatti dai trasmettitori veloci.

Per raggiungere questi obiettivi, lo strato data link prende i pacchetti provenienti dallo strato di rete e li incapsula in ***frame*** prima di trasmetterli. Ogni ***frame*** contiene un'intestazione (***header***), una sezione per contenere il pacchetto (***payload field***) e una sequenza di chiusura (***frame trailer***).



La funzione del livello data link consiste nel fornire servizi allo strato di rete. Quello principale è quello di trasferire dati dallo strato network della macchina sorgente allo strato di rete della macchina destinazione.

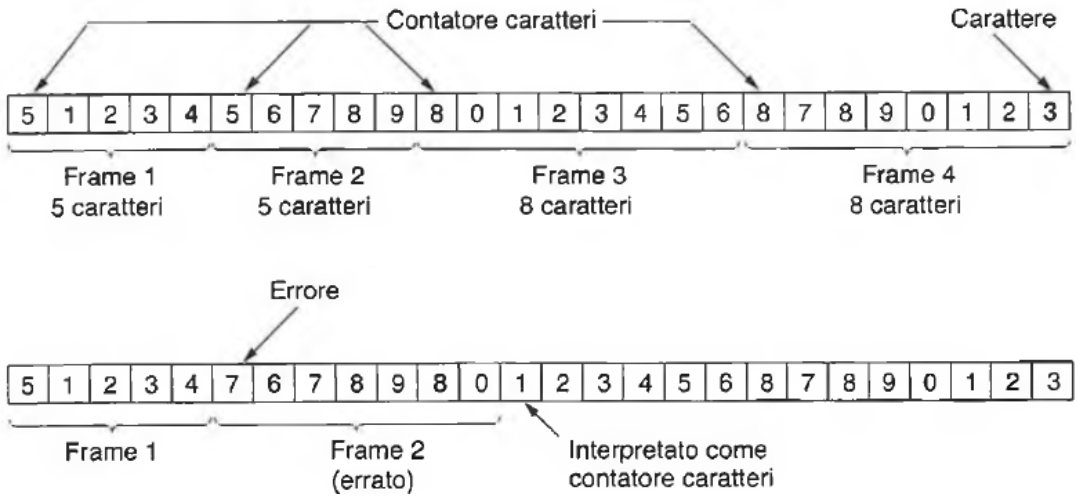
Tre servizi che vengono comunemente forniti a questo livello sono:

1. **servizio senza conferma *(unacknowledged)* senza connessione**, consiste nell'avere una macchina sorgente che invia dei frame indipendenti alla macchina destinazione, senza che quest'ultima debba dare conferma dell'avvenuta ricezione (es. Ethernet). L'uso di questa classe di servizio è legittimo quando la frequenza degli errori di trasmissione è molto bassa, così che la correzione può essere fatta dagli strati superiori.
2. **servizio con conferma** (***acknowledged***) **senza connessione,** questo genere di servizio continua a non usare nessun tipo di connessione logica, però ciascun frame è inviato individualmente e ne viene fatto *l'acknowledge* (*conferma della ricezione*). In questo modo il mittente riesce a sapere se un frame è arrivato a destinazione in modo corretto oppure no, con la possibilità di essere rispedito (es. reti wireless).
3. **servizio con conferma orientato alla connessione,** con questo tipo di servizio le macchine sorgente e destinazione stabiliscono una connessione prima d'iniziare a trasferire i dati. Ogni frame trasferito attraverso la connessione è numerato, e lo strato data link garantisce che venga effettivamente ricevuto. Risulta anche garantita la ricezione dei frame con l'ordine corretto. Quando si usa un servizio con connessione, il trasferimento dei dati avviene in tre fasi distinte. Nella prima fase viene stabilita la connessione; il mittente e il ricevente inizializzano variabili e contatori necessari per tenere traccia di quali frame sono stati ricevuti e quali no. Nella seconda fase uno o più frame vengono trasmessi. Nella terza fase la connessione viene rilasciata, liberando le variabili, i buffer e le altre risorse usate per mantenere la connessione (es. subnet di una WAN).

Framing

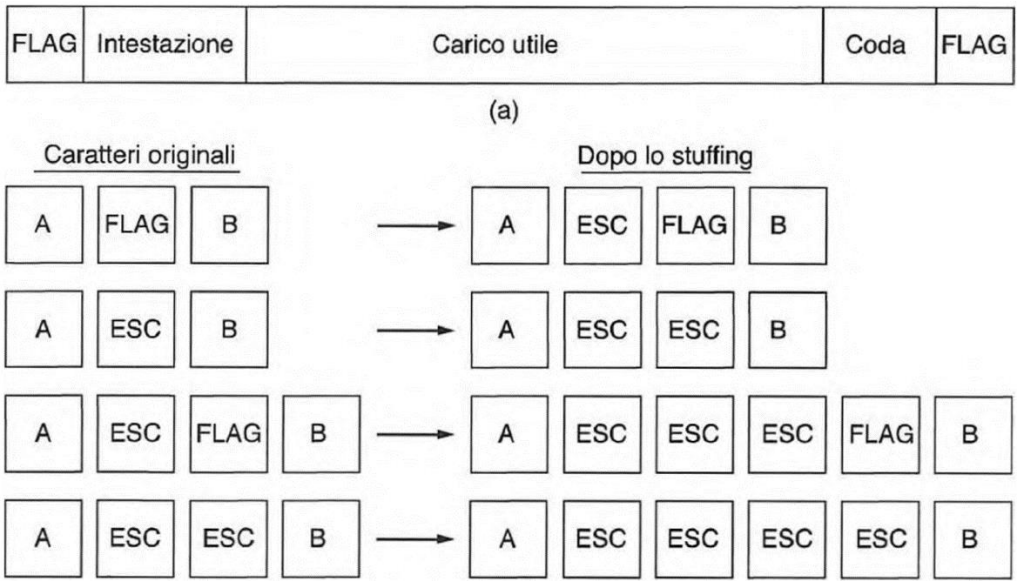
Il tipico approccio per lo strato data link è quello di suddividere il flusso di bit in una serie discreta di frame e calcolare il ***checksum*** per ogni frame. Esistono diverse tecniche:

1. ***conteggio dei caratteri***, usa un campo nell'intestazione per specificare il numero di caratteri nel frame. Quando lo strato data link della destinazione legge tale numero, sa quanti caratteri seguiranno e quindi sa dove si trova la fine del frame.



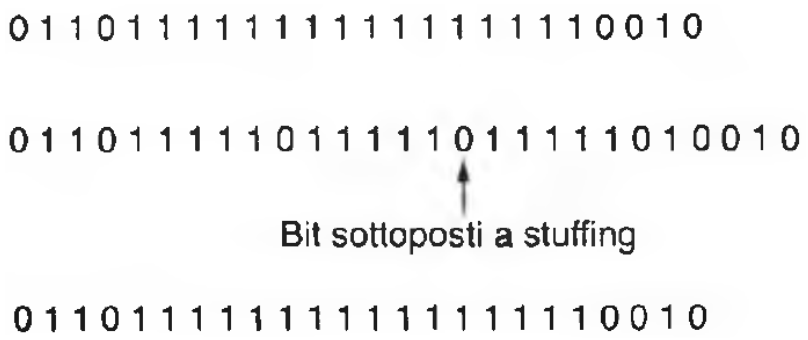
   Il problema di questo algoritmo è che il conteggio può essere alterato da un errore di trasmissione.

2. ***flag byte con byte stuffing,*** introduce un byte speciale all'inizio e al termine di ogni frame chiamato ***flag byte***, per delimitare sia l'inizio sia la fine dei frame



   Può accadere che il valore corrispondente al flag byte compaia dentro ai dati, interferendo così con le operazioni di framing. La risoluzione consiste nel far sì che la sorgente inserisca un byte di ***escape*** (***ESC***) subito prima di ogni occorrenza del flag byte nei dati. Lo strato data link della destinazione provvederà a rimuovere i byte di escape prima di passare i dati allo strato network. Questa tecnica è chiamata ***byte stuffing*** o anche ***character stuffing***.

3. ***flag di inizio e fine con bit stuffing,*** ogni frame comincia e finisce con un gruppo speciale di bit, 01111110, che in sostanza è un flag byte. Ogni volta che lo strato data link della sorgente incontra cinque 1 consecutivi nei dati inserisce automaticamente un bit con valore 0 nel flusso in uscita.

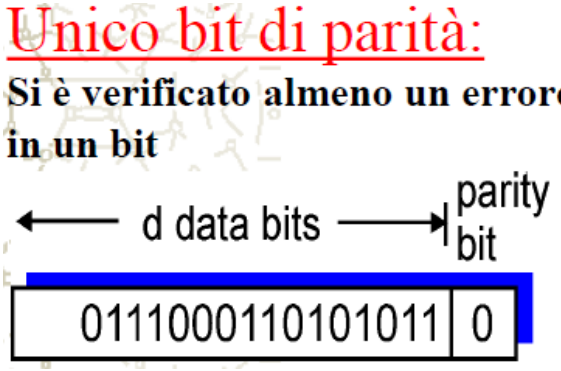


Questa operazione è chiamata ***bit stuffing.*** Quando la destinazione riceve cinque bit consecutivi con valore 1 seguiti da uno 0, automaticamente elimina lo 0.

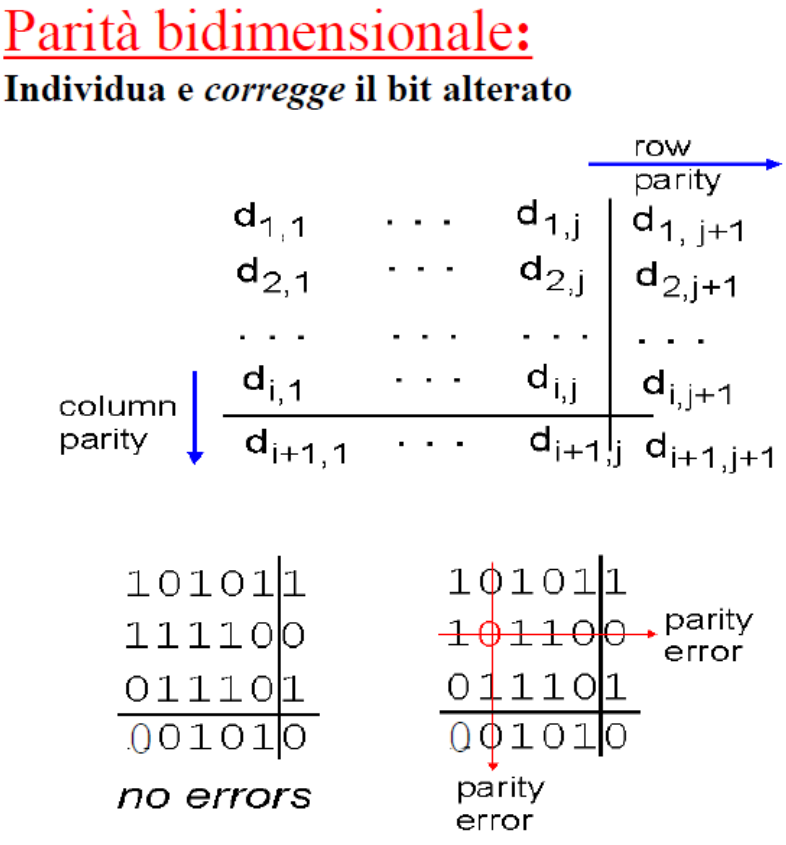
Controllo degli errori

Il controllo dell'errore si basa su codici di ridondanza, che aggiungono bit alla parola dati per verificarne la correttezza. Tali codici si suddividono in:

- ***codici rilevatori***: in grado unicamente di rilevare la presenza o meno di errori nel frame, ma non la loro posizione, in questo caso il ricevente può chiedere la ritrasmissione del messaggio.



- ***codici correttori***: in grado di rilevare una o più posizioni errate nel frame e quindi di correggerle per semplice inversione del bit.



Vi sono 3 principali codici a rilevazione d'errore, diversi tra loro:

1. ***Bit di parità***, sono in grado di rilevare la presenza di un numero dispari di errori. Il bit di parità aggiunto assume il valore 0 o 1 per rendere pari il numero di "1" nella sequenza da trasmettere (parità pari) o per renderlo dispari (parità dispari). Per calcolare il bit di parità è sufficiente effettuare l'XOR dei bit di dato nella parola nel caso di parità pari e negare tale risultato nel caso di parità dispari. La parola trasmessa sarà formata dalla parola originale e dal bit di parità. Il ricevitore provvederà a ricalcolare la parità sulla configurazione ricevuta (escludendo il bit di parità aggiunto): confrontando il nuovo bit di parità con quello ricevuto è possibile stabilire se la trasmissione è avvenuta correttamente o no.
2. ***Check-sum***, si basa sull'aggiunta di simboli calcolati non sui singoli codici delle parole che costituiscono la comunicazione, ma valutati su un blocco di parole. L'obiettivo è quello di ottenere la più alta possibilità di rilevare errori con la minor ridondanza introdotta. La Check-sum è una delle tecniche di rilevazione errori maggiormente utilizzata per trasmissione a breve distanza. Dato un blocco di informazioni da codificare, la parola di controllo si calcola effettuando la somma in algebra modulo 2 (XOR) di tutti i codici delle singole parole. Il risultato è un byte che viene anch'esso trasmesso. Il ricevitore si preoccuperà di ripetere l'operazione sul pacchetto ricevuto confrontando il suo risultato con quello inviatogli dal trasmettitore: se le due somme coincidono significa che la trasmissione è avvenuta senza interferenze.
3. ***Codici di ridondanza ciclica (CRC) o codifica polinomiale***, È un altro metodo per la rilevazione degli errori è quello dei codici ciclici. Gli n bit del blocco da trasmettere vengono considerati come coefficienti di un polinomio di grado n-1 nella variabile x. Tale polinomio, che chiameremo M(x), viene poi diviso per un altro polinomio fissato dalle convenzioni internazionali, chiamato polinomio generatore e indicato con G(x). Per rilevare la presenza di un errore il ricevitore divide il messaggio ricevuto per G(x) e verifica che il resto sia nullo. Se non lo è il ricevitore deve chiedere la ripetizione del messaggio.

Controllo di flusso 3.1.4

IPv6 - https://www.memexcomputer.it/indirizzo-ipv6/

La Figura 5.68 mostra l'intestazione IPv6. Il campo ***version*** contiene sempre il valore 6 associato a IPv6 (4 è il valore associato a IPv4). Durante il periodo di transizione da IPv4, che probabilmente si protrarrà per una decina di anni, i router saranno in grado di esaminare questo campo e identificare il tipo di pacchetto in transito. E interessante notare che questo test fa sprecare alcune istruzioni nel percorso critico, perciò molte implementazioni probabilmente tenteranno di non eseguirlo, usando un campo nell'intestazione data link che distingue i pacchetti IPv4 dai pacchetti IPv6; in tal modo, i pacchetti possono essere passati direttamente al gestore di strato network corretto.

Rendere lo strato data link conscio del tipo di pacchetto di rete viola però completamente la regola essenziale che afferma che ogni strato non dovrebbe essere a conoscenza del significato dei bit ricevuti dallo strato sovrastante. I dibattiti tra i sostenitori del "Fallo bene" e del "Fallo velocemente" saranno senza dubbio lunghi e vigorosi. Il campo ***traffic class*** (classe del traffico) è utilizzato per distinguere i pacchetti con diversi requisiti di distribuzione in tempo reale. Un campo progettato per questo scopo esiste in IP fin dall'inizio, ma è stato implementato solo sporadicamente dai router. Sono in corso esperimenti per determinare il miglior utilizzo per la distribuzione dei dati multimediali.

Anche il campo ***flow label*** (etichetta di flusso) è ancora sperimentale, ma sarà utilizzato per consentire a una sorgente e a una destinazione di impostare una pseudoconnessione con particolari proprietà e requisiti. Per esempio, un flusso di pacchetti generato da un processo su un particolare host sorgente e diretto a un particolare processo su un particolare host di destinazione potrebbe avere rigorosi requisiti di ritardo e quindi aver bisogno di banda riservata. Il flusso può essere impostato in anticipo e gli si può assegnare un identificatore. Quando appare un pacchetto con flow label diversa da zero, tutti i router cercano l'etichetta nelle loro tabelle interne per scoprire il tipo di trattamento speciale richiesto dal pacchetto. In realtà, i flussi sono un tentativo di combinare la flessibilità della rete a datagrammi e le garanzie di una sottorete a circuito virtuale.

Ogni flusso è rappresentato dall'indirizzo sorgente, dall'indirizzo di destinazione e dal numero di flusso, perciò tra una data coppia di indirizzi IP possono essere attivi molti flussi. Inoltre, in questo modo, anche se due flussi provenienti da host diversi ma con la stessa etichetta di flusso passano attraverso lo stesso router, il router sarà in grado di distinguerli usando gli indirizzi di origine e di destinazione. Poiché ci si aspetta che le etichette di flusso siano scelte a caso e non siano assegnate in modo sequenziale partendo da 1, i router devono eseguire su di esse un'operazione di hash.

Il campo payload lenght (lunghezza del carico utile) indica il numero di byte che seguono l'intestazione di 40 byte mostrata nella Figura 5.68. È stato abbandonato il nome del campo total lenght adottato da IPv4 perché il significato è leggermente cambiato: i 40 byte di intestazione non sono più conteggiati nel calcolo della lunghezza.

Il campo ***next header*** (intestazione successiva) fa trapelare un segreto. Il motivo per cui è stato possibile semplificare l'intestazione è che ci possono essere altre intestazioni estese

(opzionali). Questo campo indica quale delle sei (per il momento) intestazioni estese, se presente, segue l'intestazione corrente. Se questa intestazione è l'ultima intestazione IP, il campo next header indica il gestore del protocollo di trasporto (TCP o UDP) al quale va passato il pacchetto.

Il campo ***hop limit*** è utilizzato per impedire ai pacchetti di vivere per sempre. In pratica, funziona come il campo lime to live di IPv4, vale a dire che è decrementato a ogni salto del pacchetto. In teoria, in IPv4 il valore rappresentava una durata espressa in secondi, ma nessun router utilizzava l'etichetta in quel modo, perciò il nome è stato modificato per riflettere il modo in cui è realmente utilizzato.

Gli ultimi campi rappresentano l'indirizzo sorgente (***source address***) e l'indirizzo di destinazione (***destination address***). La proposta originale di Deering (SIP) utilizzava indirizzi a 8 byte, ma durante il processo di revisione in molti pensavano che IPv6 avrebbe esaurito gli indirizzi entro pochi decenni usando 8 byte, mentre con indirizzi a 16 byte non si sarebbero mai esauriti. Altri sostenevano che 16 byte erano troppi, mentre altri ancora avrebbero preferito utilizzare indirizzi a 20 byte compatibili con il protocollo di data- gramma OSI. Un'altra fazione proponeva indirizzi di lunghezza variabile. Dopo un lungo dibattito vennero scelti gli indirizzi a 16 byte di lunghezza fissa, il miglior compromesso tra tutte le possibilità.

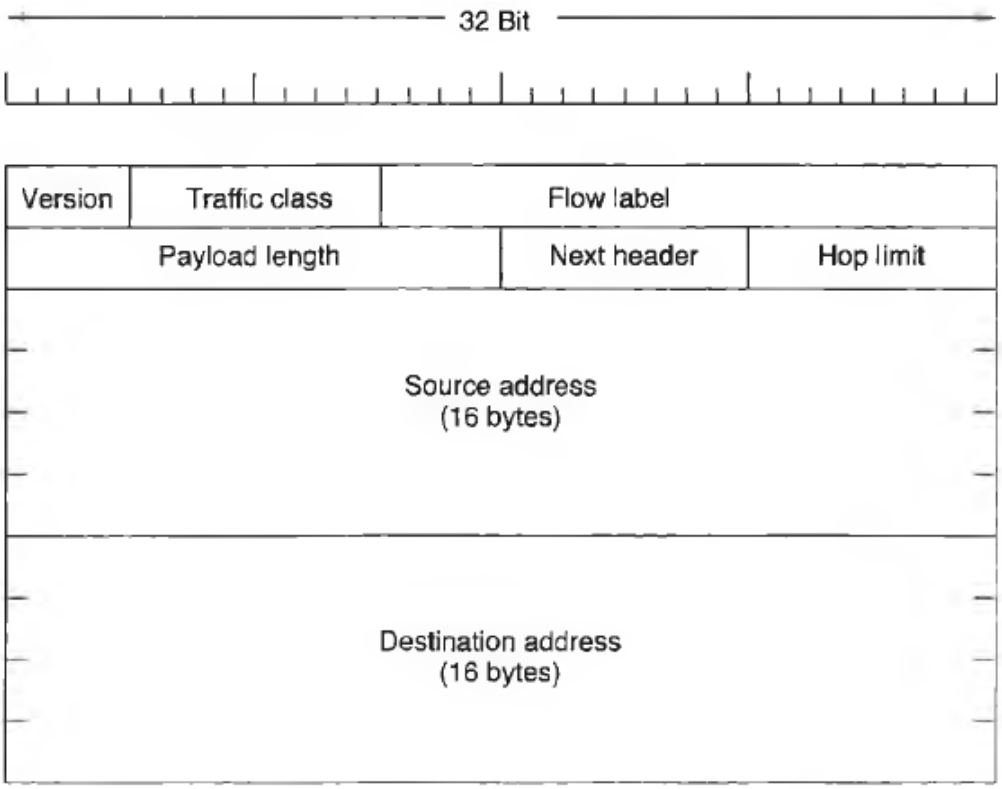


Figura 5.68. L'intestazione fissa IPv6 (obbligatoria).

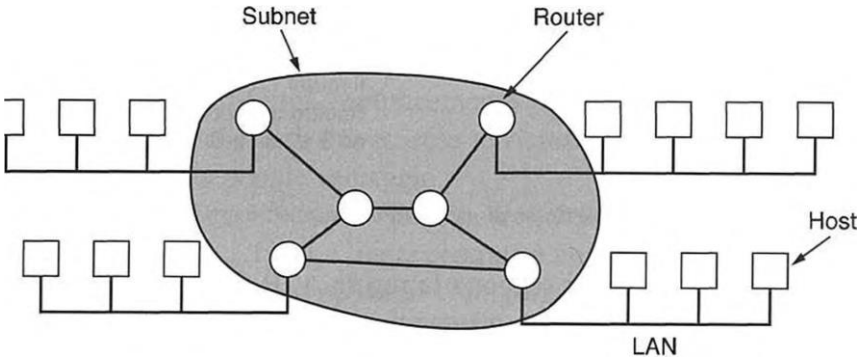
Subnet - https://it.wikipedia.org/wiki/Sottorete

per brevità chiamata semplicemente subnet (sottorete).

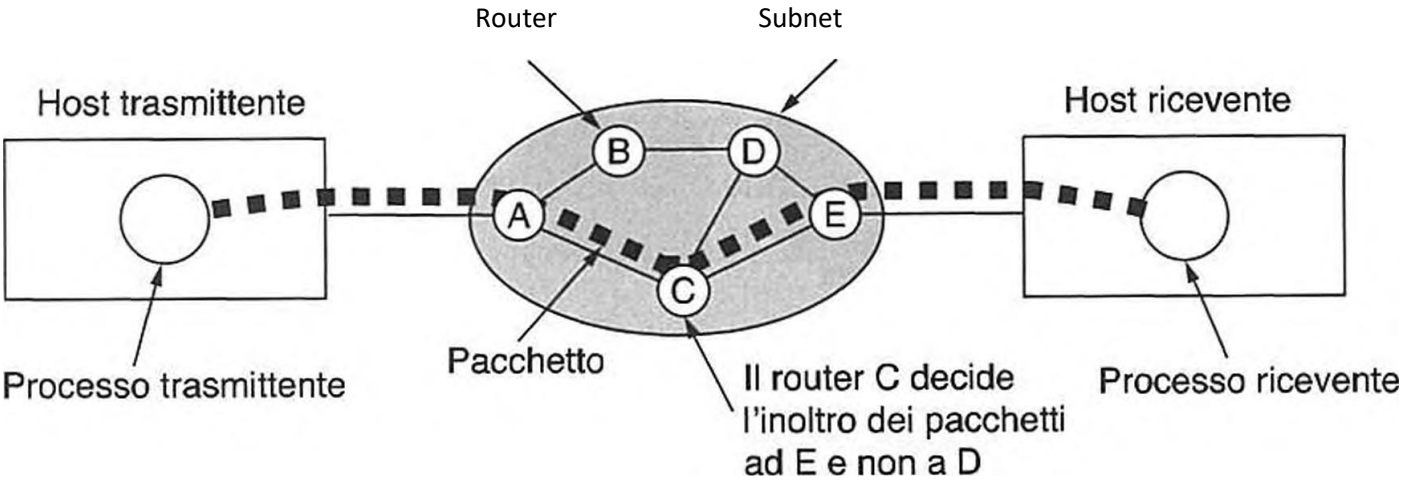
Il compito della subnet è quello di trasportare i messaggi da un host all'altro, come la rete telefonica trasporta le parole da chi parla a chi ascolta. La separazione dell'aspetto di pura comunicazione della rete (la subnet) dagli aspetti applicativi (gli host) semplifica notevolmente il progetto dell'intera rete.

Nella maggior parte delle WAN, la subnet è formata da due componenti: linee di trasmissione ed elementi di commutazione. Le linee di trasmissione spostano i bit tra le macchine. Possono essere realizzate con cavo in rame, fibra ottica, o anche collegamenti radio. Gli elementi di commutazione sono computer specializzati che collegano tre o più linee di trasmissione. Quando i dati arrivano da una linea ricevente, l'elemento di commutazione deve scegliere una linea di uscita su cui inoltrarlo.

L'insieme di linee di comunicazione e router (ma non gli host) forma la subnet.



Quando un pacchetto viene inviato da un router verso un altro, attraverso router intermedi, il pacchetto viene ricevuto integralmente da ciascun router intermedio, memorizzato finché non si libera la linea di uscita necessaria, e poi inoltrato. Una subnet organizzata secondo questo principio si chiama store-and-forward o packet-switched (a commutazione di pacchetto).



Spesso si fa confusione tra subnet, reti e internetwork. Subnet è un termine particolarmente appropriato al contesto delle Wide Area Network, dove si riferisce all'insieme di router e linee di comunicazione possedute dall'operatore di rete.

Lo strato network controlla il funzionamento della subnet. Un problema chiave riguarda la modalità con cui i pacchetti sono inoltrati dalla sorgente alla destinazione. L'inoltro si può basare su tabelle statiche che sono "cablate" dentro la rete e raramente modificate. Si può anche stabilire all'inizio di ogni conversazione che può essere per esempio una sessione terminale (login su un computer remoto). Infine, potrebbe essere altamente dinamico, determinato per ogni pacchetto in modo da riflettere lo stato corrente del carico della rete.

Quando nella subnet sono presenti contemporaneamente troppi pacchetti, vanno a interferire l'un l'altro formando colli di bottiglia. Anche il controllo di queste congestioni spetta allo strato network.

Per implementare le sottoreti, il router principale ha bisogno di una maschera di sottorete (subnet mask) che indichi il punto di demarcazione tra numero di rete (inclusivo di quello della sottorete) e quello di host,

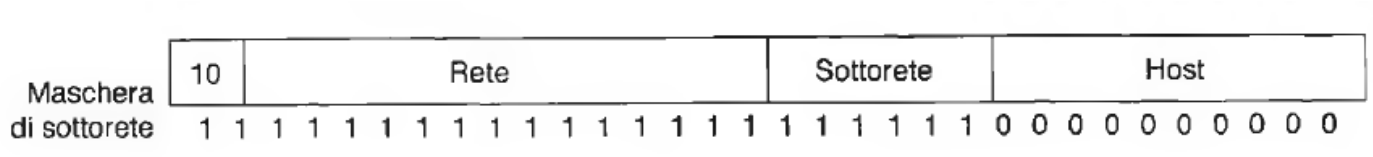


Figura 5.58. Una rete di classe B divisa in 64 sottoreti.

Anche le maschere di sottorete sono scritte in notazione decimale a punti, oppure da una barra seguita dal numero di bit della parte che rappresenta la rete più la sottorete. Nel caso della Figura 5.58 la maschera di sottorete può essere scritta come 255.255.252.0, oppure dalla notazione alternativa /22 che indica che la maschera di sottorete è lunga 22 bit.