Appunti di Reti

Sommario

Questo documento contiene appunti di Reti di Calcolatori, estrapolati dall'omonimo libro di Andrew A.Tanenbaum.

Indice

1 Strato Fisico

1.1 Mezzi di Trasmissione

1.1.1 Basi teoriche della comunicazione dati

Le informazioni posso essere trasmesse via cavo variando alcue proprietà fisiche (tensione/corrente). Rappresentando la tensione/corrente in una funzione f(t) è possibile analizzare il segnale. Fourier dimostrò che un segnale di questo tipo, periodico e abbastanza regolare può essere descritto da una ideale somma infinita di seni e coseni (**Serie di Fourier**). Una funzione può essere ricostruita a partire dalla sua serie di Fourier. Analizziamo un segnale di trasmissione analogica e proviamo a ricostruirlo con Fourier. Dobbiamo tener conto che nessun canale trasmissivo è perfetto, per cui c'è sicuramente attenuazione. L'intervallo di frequenze trasmesse senza forte attenuazione è detto **banda passante**. Anche in un canale perfetto ci si accorge comunque che un segnale digitale non può essere trasmesso a velocità troppo elevate, esistono però alcuni schemi di codifica per aumentare la velocità di trasmissione.
Nyquist/Shannon dimostrò che la velocità massima di trasmissione è:

$$V_{max} = 2H \log_2 Vbit/sec$$

Mentre il livello di rumore si misura facendo il **rapporto segnale-rumore**. Solitamente viene indicata tale misura antecedendo $10log_{10}$ e misurando in dB.

$$Segnale/Rumore = 10log_{10}S/NdB$$

Un risultato notevole ottenuto da Shannon fu:

$$MAX_{bit/s} = Hlog_2(1 + S/N)$$

con H pari all'ampiezza di banda in Hz.

1.1.2 Mezzi Magnetici

Sistema molto semplice, utilizzato da sempre e basato su un funzionamento banale: si salvano i dati su nastri magnetici (dischi rimovibili) e si trasportano fisicamente a destinazione dove verranno letti. Se si pensa a un tir che trasporta un centinaio di HD da 1TB che percorre qualche Km per consegnare questi dischi si può intuire che la larghezza di banda è elevatissima e con un costo irrisorio. La cosa banalmente poco buona è l'enorme ritardo nella trasmissione dati.

1.1.3 Doppino

Il doppino è composto da 2 conduttori di rame isolati attorcigliati tra loro a forma elicoidale (stile DNA), questo per evitare interferenze tra di loro (risulterebbero un ottima antenna). Viene largamente utilizzato nel sistema telefonico, questo perché il doppino può attraversare diversi Km senza bisogno di amplificare il segnale così dall'abitazione si può agevolmente arrivare alla centrale. Si possono usare per trasmettere dati Analogici o anche Digitali e la larghezza di banda dipende dal diametro del cavo e dalla distanza percorsa. Esistono più categorie di questi cavi che differiscono sostanzialmente per il numero di spire per cm per ridurre le interferenze. Il doppino cat3 (usato sino al 1988) sono composte da 2 cavi isolati cavi attorcigliati. Il doppino cat5 sono come i cat3 ma utilizzano più spire per cm questo li rendono più adatti a trasmissione ad alta velocità. Il doppino può arrivare a una banda di 250-600MHz. Questi cavi sono detti anche UTP (*Unshielded Twisted Pair*).

1.1.4 Cavo coassiale

Essendo più schermato del precedente il cavo coassiale può estendersi per distanze maggiori. Esistono 2 tipi di cavi coassiali: da 50 Ohm per le trasmissioni digitali e da 75 Ohm per le analogiche. Composto da un nucleo di rame, rivestito da materiale isolante a sua volta rivestito da una calza conduttrice il tutto ricoperto da una guaina protettiva, il cavo coassiale è caratterizzato da un eccellente immunità al rumore. L'ampiezza di banda di questi cavi arriva attorno a 1GHz e dipende dalla lunghezza, dalla qualità e dal rapporto segnale-rumore del segnale.

1.1.5 Fibra Ottica

Un sistema di trasmissione ottico è formato principalmente di 3 parti: sorgente luminosa, mezzo di trasmissione e rilevatore di luce. La sorgente di luce è rappresentata o da LED o semiconduttori laser. Il mezzo trasmissivo ovviamente è la fibra composta da un nucleo (core) di vetro di pochi micron avvolto in una guaina di vetro (cladding) con indice di rifrazione più basso e infine la solita rivestitura con guaina in plastica. La fibra si basa su un principio molto semplice, ovvero che la luce che la attraversa viene riflessa al suo interno fino ad arrivare all'altra estremità del cavo, questo avviene perchè la luce immessa nel core incontra il cladding con indice di rifrazione minore e il raggio luminoso viene così riflesso (se possiede un inclinazione corretta). La velocità di tale raggio è circa quella della luce infatti il limite di banda della fibra non è dovuto alla velocità di trasmissione ma di decodifica del

segnale luminoso in impulso elettrico. Una fibra può contenere più raggi che si riflettono in essa l'importante è che il loro angolo di riflessione sia diverso. Questo tipo di fibre è detto **multimodale**. Le fibre che invece permettono la trasmissione di luce in linea retta sono le **monomodali** che non sono altro che guide d'onda ma possono raggiungere i 50Gbps per 100Km senza attenuazione.
Un problema delle fibre è il collegamento tra 2 i esse che può avvenire in 3 modi:

1. Le fibre vengono inserite in apposite prese grazie a dei connettori con perdita di circa 10-20% della luce

2. Le fibre vengono attaccate meccanicamente, messe di fronte una all'altra e poi viene avvolto in una macchina particolare per poi essere pinzate, con perdita comunque di circa il 10% della luce

3. Le fibre vengono fuse tra loro. Una soluzione quasi ottimale, anche se difficile, genera una piccola attenuazione di segnale.

Le LAN basate sulle fibre ottiche solitamente sono ad anello con congiunzione a T per ogni pc o a stella passiva. La configurazione ad anello ha il difetto che se una congiunzione si guasta salta tutta la rete.

Tabella riassuntiva

	Mezzi Magnetici	Doppino	Coassiale	Fibra ottica
Anno	1950	1980	1980	1980
Tipo dati	-	A/D	A/D	A/D
Banda(Hz)	Elevata	10M-600M	< 1G	1G - 10G (100G)
Ritardo	Elevato	Accettabile	Basso	Quasi nullo
Immunita Rumore	-	Eccellente	Eccellente	Eccellente
Costo	Basso	Basso	Intermedio	Elevato
Estensione	-	Diversi Km	Decine Km	100Km
Interferenze	-			Immune
Intrusione	-	Facile	Facile	Difficile

Figura 1: Riassunto caratteristiche dei mezzi trasmissivi.

1.2 Trasmissioni Wireless

1.2.1 Lo spettro elettromagnetico

Lo spostamento di elettroni crea campi magnetici. Questa osservazione fu fatta per la prima volta da Hertz. Il numero di oscillazioni al secondo di un'onda è chiamato **frequenza** e si misura in Hz. La distanza massima tra 2 picchi (o minimi) è chiamata **lunghezza d'onda**. Tutte le trasmissioni wireless si basano sul principio che un antenna collegata a un circuito elettrico invia onde elettromagnetiche che possono essere captate da un ricevitore posto a una distanza appropriata. Le onde viaggiano nell'etere alla velocità della luce. Qui di seguito è visualizzato lo spettro elettromagnetico e la sua suddivisione.

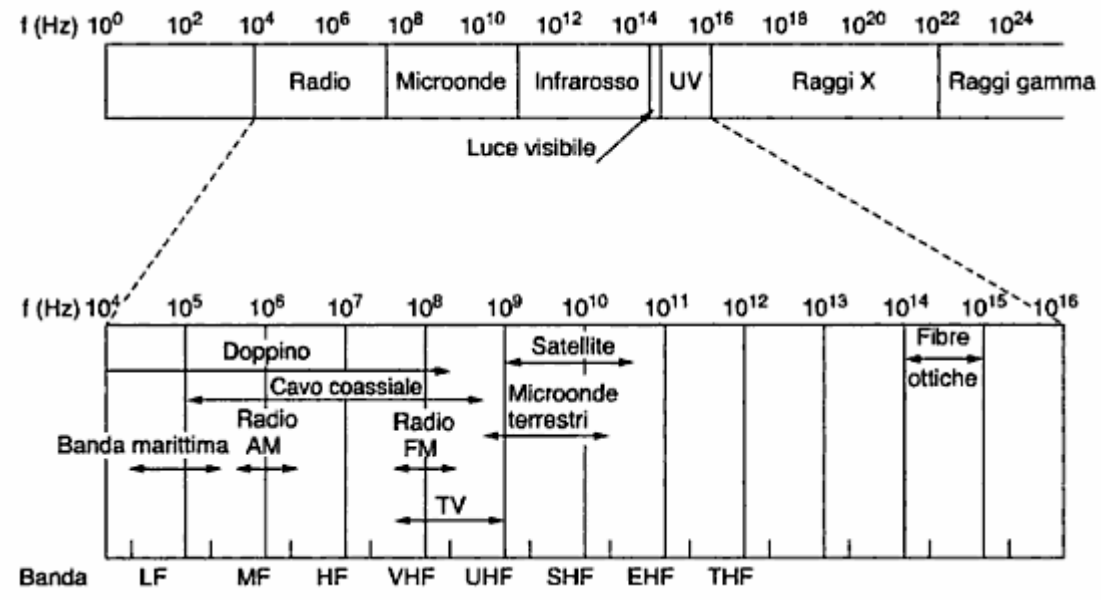


Figura 2: Spettro elettromagnetico.

1.2.2 Trasmissioni radio

Le onde radio sono onde omnidirezionali semplici da riprodurre e viaggiano per lunghe distanze attraversando gli edifici. Non necessita di alcun allineamento trasmettitore-ricevente. Nell'aria l'attenuazione delle onde con frequenze più basse è di circa $1/r^2$. Nelle bande VLF, LF, MF le onde seguono la forma del terreno e possono viaggiare per circa 1000 Km. Nelle alte frequenze invece le onde che riescono ad entrare nella ionosfera vengono riflesse e ritornano sulla terra permettendo così di percorrere distanze notevoli.

1.2.3 Trasmissione a microonde

Sopra i 100 MHz le onde viaggiano quasi in linea retta per cui è necessario un allineamento trasmettitore-ricevente. Concentrando l'onda in un piccolo raggio si ottiene un ottimo rapporto segnale/rumore. Il problema di queste trasmissioni sta nelle lunghe distanze e dalla curvatura della terra, la quale porta alla necessità di ripetitori. L'utilizzo di antenne molto alte riduce l'effetto della curvatura. Alcune onde posso rinfrangersi negli strati più bassi dell'atmosfera e arrivare in ritardo rispetto a quelle dirette e addirittura fuori fase causandone l'annullamento (**multipath fading**). Le microonde non attraversano gli edifici molto bene, e sono soggette alle condizioni climatiche. Comunemente si usano bande sopra i 10 GHZ ma sopra i 40 GHz la pioggia comincia ad assorbire le onde. Acquisto e installazione di apparecchi per questo tipo di trasmissione è molto basso.

Divisione dello spettro elettromagnetico

Tutti vogliono un pezzo di spettro per aumentare la velocità di trasmissione e quindi bisogna regolarne tale divisione, se ne occupa l'ITU-T. In passato per compiere tale divisione sono stati utilizzati 3 modi:

1. Concorso di bellezza: ognuno di coloro che voleva spettro doveva dare un motivo per il quale doveva averlo proprio lui. Problemi: corruzioni e scelte arbitrarie senza senso.

2. Lotteria: veniva fatta una vera e propria lotteria per assegnare lo spettro. Problemi: partecipavano anche i non interessanti solo per guadagnare soldi dalla rivendita dello spettro.

3. Asta: vendita all'asta dello spettro. Problema: banca rotta delle aziende.

4. Libertà: approcio che non prevede assegnamento di spettro, lasciando trasmissione libera ma regolata. Ovvero la potenza doveva essere utilizzata in modo da limitare la portata per evitare interferenze.

1.2.4 Infrarossi

Sistema economico e facile da costruire con il difetto di non riuscire ad attraversare gli ostacoli. Queste onde, infatti, si avvicinano alle onde di tipo luminose. Sono più sicuri delle onde radio per la difficoltà di intercettazione, ma risentono molto degli ostacoli. Questo a volte può rappresentare anche

un vantaggio (es. telecomandi tv). Per questi motivi sono usati per distanze brevi (collegamento pc-stampanti, telecomandi ecc..), e hanno un ruolo secondario nelle telecomunicazioni. Il sistema infrarosso non richiede alcuna licenza governativa.

1.2.5 Trasmissioni a onde luminose (LASER)

Sistema poco costoso che offre una banda molto elevata, facile da installare e non richiede licenze. La sua debolezza sta nel raggio molto sottile (e unidirezioneale) e quindi difficile da indirizzare verso il bersaglio esatto. Per ovviare al problema a volte vengono inserite lenti per rendere il raggio meno focalizzato. Il raggio laser per di più non attraversa la pioggia e la nebbia ed è soggetto a fenomeni di convezione (turbolenza provocata da fonti di calore).

1.3 Satelliti

Un satellite può essere immaginato come un grande ripetitore di microonde collocato nel cielo, che contiene molti **trasponder** (ricetrasmettitori satellitari). I raggi verso la terra possono essere più o meno grandi, questa modalità è detta **bent pipe**. I satelliti hanno un periodo orbitale che dipende dalla loro altezza rispetto alla terra. Un problema è la presenza delle **fasce di Val Allen**, strati di particelle molto cariche, che distruggerebbero un satellite in poco tempo.

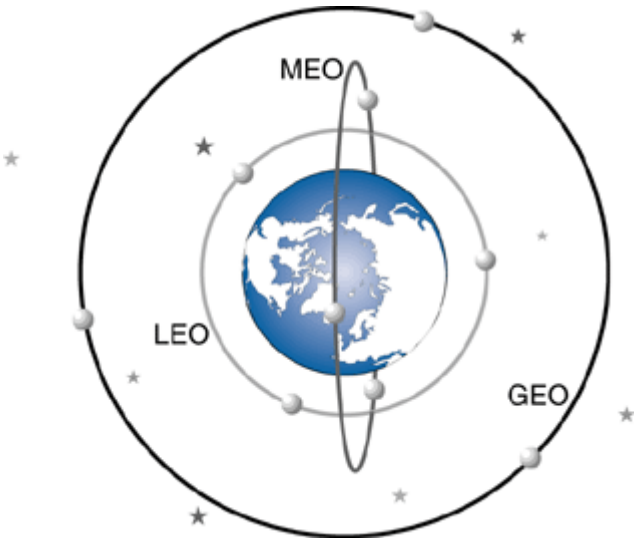


Figura 3: Posizione dei salettiti GEO, MEO e LEO

1.3.1 Satelliti Geostazionari (GEO)

Questi satelliti sono posti in orbite molto alte e con le tecnologie odierne non si possono collocare 2 satelliti GEO a meno di 2 gradi nel piano equatoriale, quindi con 180 satelliti si copre tutto. L'allocazione degli slot spaziali è gestito dall'ITU. L'ITU inoltre ha assegnato alcune bande di frequenza alle applicazioni satellitari in modo da non interferire con i sistemi a microonde preesistenti. I segnali inviata da questi satelliti viaggia alla velocità della luce, ma essendo molto lontani dalla terra hanno comunque un ritardo di circa 300 ms. I primi satelliti GEO con una singola emissione coprivano circa 1/3 della terra, chiamata **impronta**. Poi con lo sviluppo delle tecnologie si è cominciato a concentrare i raggi trasmissivi (spot) in aree geografiche più piccole (centinaia di Km). Un nuovo passo avanti nel settore delle comunicazioni satellitari si ebbe con le stazioni VSAT, piccole stazioni con una'antenna da

circa 1m che comunicano con i satelliti GEO e per le loro piccole dimensioni e potenza non essendo in grado di comunicare tra loro si è dovuto ideare alcune stazioni particolari più potenti per fare da ponte. Sono ovviamente mezzi di trasmissione broadcast e per quanto riguarda la sicurezza sono un disastro. Il costo della trasmissione satellitare non dipende dalla distanza, ma è costante. Hanno una reattività quasi istantanea e un ottimo tasso di errore.

1.3.2 Satelliti su orbite medie (MEO)

Questi satelliti si muovono sopra di noi a una velocità relativamente bassa percorrendo il giro del pianeta in circa 6 ore. Coprono un area più piccola dei GEO e si possono raggiungere con mezzi meno potenti. I 24 satelliti GPS che orbitano a 18000 Km sono di tipo MEO.

1.3.3 Satelliti su orbite basse (LEO)

Si spostano molto velocemente e per realizzare un sistema completo sono necessari molti satelliti di questo tipo. Essendo vicini alla crosta terrestre le stazioni non hanno bisogno di molta energia per comunicare con poco ritardo.

Iridium

Lanciati nel 1997 i 66 satelliti LEO del progetto Iridium (Motorola) vennero acquistati da un investitore riprendendo il servizio nel marzo 2001, che era stato fermato nel 1999. Il progetto fornisce un servizio di telecomunicazione a livello mondiale basato su 'cellulari particolari'. I satelliti Iridium sono collocati a 750 Km di altezza, con un satellite ogni 32 gradi. Le trasmissioni avvengono nello spazio: ogni satellite comunica con altri satelliti fino a destinazione.

GlobalStar

Basato su 48 satelliti LEO utilizzando uno schema diverso dal precedente. Il satellite che riceve la chiamata trasmette a una centrale terrestre che comunica con altre fino al satellite posto sulla cella del destinatario. La complessità resta quindi terrestre facilitandone la gestione.

Teledesic

Progetto mirato per gli utenti internet, con l'idea di offrire 100 Mbps in trasmissione e 720 in ricezione. Il sistema è composto da 32 satelliti LEO

con un impronta più grande. Basato sulla commutazione di pacchetto, con ogni satellite in grado di instradare ogni singolo pacchetto.

1.3.4 Satelliti o fibra?

E' ovvio che la comunicazione con le fibre sia molto veloce ma ci sono molti settori in cui i satelliti non possono essere rimpiazzati dalla fibra, come ad esempio la comunicazione mobile. Le comunicazioni broadcast per eccellenza sono satellitari. Un altro settore è la comunicazione in luoghi ostili, in cui le infrastrutture terrestri scarseggiano e la posa di cavi ottici non sarebbe il massimo. Inoltre anche le zone in cui i costi di posa sono elevati il satellite può essere un ottima alternativa. In sostanza per la comunicazione terrestre è sicuramente migliore la fibra ma il satellite rimarrà indispensabili per altri settori.

1.4 Sistema Telefonico

1.4.1 Struttura della rete

Inizialmente il mercato dei telefoni prevedeva la vendita di 2 apparecchi e spettava all'utente tirare il cavo tra i 2 telefoni. Questo intuitivamente creò una struttura di rete troppo confusa. Bell notato questo particolare aprì il primo ufficio di commutazione nel 1878. Le chiamate quindi dovevano passare per la centrale nella quale un addetto si occupava di collegare con un cavo il chiamante al chiamato. La rete però, sebbene meno complessa, rimase ancora troppo confusa poichè non era pensabile collegare ogni ufficio di commutazione a una centrale. Vengono così ideati i livelli delle centrali di commutazione. Inizialmente con 2 livelli fino ad arrivare a 5. In generale una comunicazione avviene a più livelli:

1. La richiesta di chiamata arriva alla centrale locale del chiamante alla quale è collegato direttamente con 2 cavi di rame (doppino di categoria 3). Se il chiamato appartiene alla stessa centrale locale avviene il collegamento tra le parti.

2. La centrale locale è collegata a una centrale interurbana (con cavi in fibra/microonde/coassiale),e come per quella locale se la centrale locale del chiamato è collegata alla stessa centrale interurbana allora le parti si collegano.

3. Le centrali interurbane sono connesse a centrali intermedie e la trasmissione avviene analogamente alle precedenti.

Le trasmissioni sono preferibili in digitale per la non necessità di accuratezza, per il basso costo e per la semplicità di gestione. Si possono quindi individuare 3 componenti fondamentali del sistema telefonico:

1. Collegamenti locali: rappresentano il collo di bottiglia del sistema
2. Linee: collegamenti in fibra tra le centrali
3. Centrali di commutazione: che spostano le chiamate tra le linee

1.4.2 Modem, ADSL, Wireless

Il modem ha la funzione di convertire i dati dalla forma digitale del pc alla forma analogica necessaria per inviare i dati attraverso un collegamento locale. Nella centrale poi i dati vengono ritrasformati in digitale e poi ritrasmessi in linee a lunga distanza. Dall'altro capo poi ci sarà il modem per la conversione inversa alla precedente. I problemi principali delle linee di trasmissioni sono 3:

1. Attenuazione: rappresenta la perdita di energia (in dB/Km) dalla propagazione del segnale. Tale perdita dipende dalla frequenza.
2. Distorsione: rappresenta la differenza di velocità tra le varie componenti del segnale.
3. Rumore: rappresenta energia indesiderata all'interno del segnale originale, causate da sorgenti di trasmissione esterne.

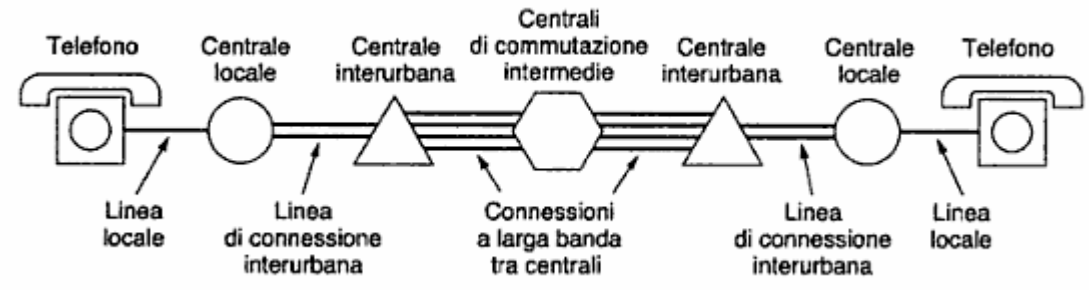


Figura 4: Percorso tipico di una chiamata a media distanza

Modem

Per riuscire a inviare dati in forma digitale è necessario un ampio spettro di frequenza questo rende adatta la trasmissione in banda base (DC) solo a basse velocità e distanze brevi. Il problema è aggirato utilizzando una trasmissione (AC) aggiungendo un segnale portante tra i 1000 e i 2000 Hz. Nella modulazione di ampiezza (ASK) sono utilizzate 2 diverse ampiezze 0 e 1, nella modulazione di frequenza (FSK) si utilizzano 2 o più toni. In quella di fase invece l'onda portante è spostata di 0 o 180 gradi a intervalli regolari.

Un apparecchio che utilizza uno di questi metodi per 'tradurre' un flusso

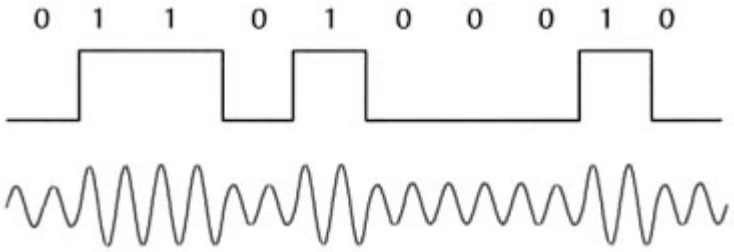


Figura 5: Modulazione d'ampiezza

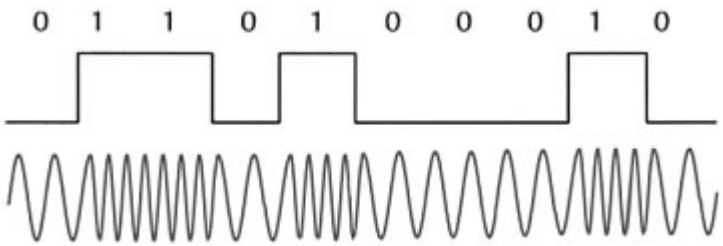


Figura 6: Modulazione di frequenza

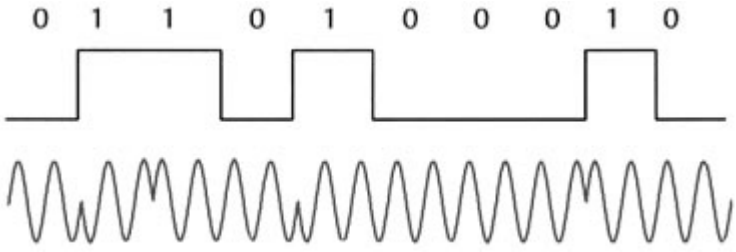


Figura 7: Modulazione di fase

di bit in segnale analogico è detto modem. La maggior parte dei modem

campiona 2400 volte al secondo. Il numero di campionamenti al secondo si misura in **baud**. Durante ogni baud è trasmesso un simbolo. Concetti da ricordare:

- Banda passante: intervallo di frequenza passante nel mezzo con un attenuazione minima(Hz)
- Baud rate: numero di campioni per secondo (=frequenza simboli)
- Modulazione: determina il numero di bit per simbolo
- Frequenza di bit: quantità di (simboli/sec)*(bit/simbolo)

La trasmissione digitale è adatta in banda base (DC) a basse velocità. Per aggirare i problemi di tale banda si usa la trasmissione AC introducendo un segnale costante detto **portante d'onda sinusoidale**. La sua ampiezza, frequenza o fase posso essere modulate per inviare informazioni. Nella **modulazioni di ampiezza** vengono usate 2 ampiezze diverse per rappresentare 1 e 0 mentre in quella di **frequenza** (FSK) si utilizzano più toni. Infine nella **modulazione di fase** più semplice l'onda viene spostata di 0 o 180 gradi (schemi migliori utilizzano spostamenti più piccoli). I modem odierni utilizzano modulazioni ibride per avere un maggior baud rate. Un esempio è la QPSK (*Quadrate Phase Shift Keying*). Questa tecnica di modulazione prevede l'utilizzo di più fasi e più ampiezze, e in base al numero di combinazioni nascono i nomi QAM-16 (4 bit per simbolo utilizzando 4 fasi e 4 ampiezze, 9600 bps), QAM-64 (*Quadrature Amplitude Modulation*)e così via. Uno schema con costellazione molto fitta è soggetto a errori per questo

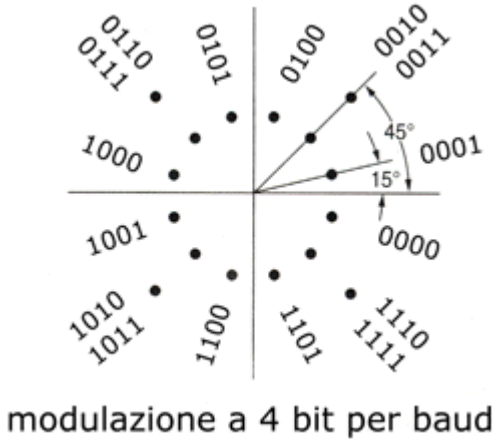


Figura 8: Diagramma costellazione QAM

i modem che li adottano utilizzano meccanismi di correzione degli errori, per esempio con un bit extra di parità. Gli standard utilizzati dai modem più conosciuti sono:

- V.32: trasmette 4 bit più 1 di parità a 2400 baud (9600 bps)
- V.32 bis: trasmette 6 bit più 1 di parità a 2400 baud (14400 bps)
- V.34: utilizza 12 bit per simbolo a 2400 baud (28800 bps)
- V.34 bis: utilizza 14 bit per simbolo a 2400 baud (33600 bps)

Una connessione che permette ai dati di viaggiare in entrambe i sensi è detta **full duplex**, mentre se lo permette ma solo uno alla volta è detta **half duplex** se invece è permesso un solo senso è **simplex**. In base quanto detto la velocità massima dei modem è 34 Kbps questo non è vero perché l'ampiezza del canale telefonico è 4 MHz quindi per Nyquist il numero di campioni massimo è 8000, per gli 8 bit per campione usato negli U.S. si hanno 64 Kbps! In realtà non è così perché 1 bit serve per il controllo per cui si riduce il tutto a 56 Kbps che è lo standrard **V.90**.

Linee DSL

Servizi con banda maggiore a quella appena descritta sono detti a **banda larga**. Per ottenere questo aumento di banda viene usato un artificio che si basa sostanzialmente sulla rimozione del filtro che limitava la capacità del collegamento locale a 3100 Hz. Questi servizi, detti xDSL, sarebbero dovuti funzionare sui doppini già installati nelle abitazioni, senza creare problemi ai telefoni, con costi limitati e non legati al tempo di utilizzo e ovviamente dovevano velocizzare di molto quei 56 Kbps.

- Proposta AT&T: divisione dello spettro delle reti locali in POTS, upstream e downstream.
- DMT (*Discrete MultiTono*): lo spettro è diviso in 256 canali indipendenti. Il primo canale utilizzato per POTS, i 5 successivi non vengono utilizzati per limitare le interferenze e tutti gli altri per i dati, e chi fornisce il servizio decide come dividerli tra up e down (solitamente 10%-90%).

Quest'ultima idea fa nascere l'ADSL che può arrivare a circa 8 Mbps in ricezione e 1 Mbps in trasmissione. In realtà questi valori sarebbero maggiori ma il rapporto segnale/rumore non li permettono. Per usufruire dell'ADSL è necessario installare un **NID** e uno **splitter** spesso all'interno di uno stesso pezzo, questo per filtrare le bande e un modem. Se quest'ultimo non è interno al pc allora si deve collegare a esso con ethernet, USB o rete wireless.

1.4.3 Multiplexing

Per limitare i costi le aziende hanno ideato modi per convogliare più conversazioni nello stesso mezzo fisico, appunto il **Multiplexing**. Esistono sostanzialmente 2 categorie di quest'ultimi:

- FDM: Frequency Division Multiplexing
- TDM: Time Division Multiplexing

In FDM lo spettro fi frequenze è diviso in bande e ogni utente ne possiede una. In TDM gli utenti si danno il cambio, tipo round-robin.

Multiplexing a divisione di frequenza

La banda è limitata dai filtri a 3.1 KHz, e nell'unione in multiplexing viene allocato uno spazio un po superiore, 4 KHz per avere un po di tolleranza. Poi ogni canale voce viene aumentato di una frequenza diversa e unito agli altri senza sovrapposizione. Nonostante questi accorgimenti 2 canali adiacenti avranno u po di sovrapposizione che potrebbe tramutarsi in un po di rumore nei 2 canali. Uno standard comune prevede 12 canali voce uniti in multiplexing nella banda tra 60-108 KHz. Questa unità è chiamata **gruppo** che possono essere uniti in multiplexing in un **supergruppo** e a loro volta ancora in un **mastergroup**.

Multiplexing a divisione di lunghezza d'onda

Utilizzato per i canali in fibra, il **WDM** (*Wavelenght Division Multiplexing*) si fonda sul principio della combinazione e divisione di lunghezze d'onda. Più fibre vengono combinate convogliando ogni segnale in un unico canale nella cui estremità c'è uno splitter utile a ripristinare i segnali delle fibre di partenza. La differenza sostanziale rispetto all'FDM è il sistema ottico completamente passivo. Un sistema con molti canali e lunghezze d'onda ravvicinate è definito **DWDM** (Dense WDM).

Multiplexing a divisione di tempo

Gestita completamente da dispositivi elettronici digitali, per cui è necessaria una conversione da parte della centrale prima di trasmettere il segnale sulla linea di uscita. La centrale locale digitalizza il segnale analogico producendo numero a 8 bit (grazie al **codec**, *coder-decoder*), elaborano 8000 campioni al secondo. Questa tecnica è chiamata PCM e costituisce il cuore del sistema

telefonico odierno. Nel mondo esistono molti schemi PCM diversi incompatibili tra loro. In Giappone e in Nord America si utilizza la portante T1 in altre zone del mondo quella E1. Una tecnica chiamata **differential pulse code modulation** al posto di inviare l'ampiezza digitalizzata invia la differenza rispetto alla precedente così da ridurre il numero di bit (da 7 a 5) utili supponendo che sia poco probabile il salto di ± 16. Una variante di questa tecnica detta **modulazione delta** si basa su un principio simile: ogni valore campionato differisce dal precedente di ± 1 sotto le condizioni che può essere trasmesso un singolo bit che dice se il nuovo campione è maggiore o minore del precedente. Questa tecnica che ipotizza una bassa variazione di segnale può avere problemi con bruschi cambiamenti di livello. Esistono altre tecniche dette **codifiche per ipotesi** che utilizzando pochi valori precedenti prevedono il successivo.

1.4.4 Commutazione

Commutazione di circuito

Quando viene avviata una telefonata l'apparecchio di commutazione del sistema telefonico prova a creare un percorso fisico tra il chiamante e il chiamato.

Commutazione di messaggio

Questa tecnica non prevede un collegamento fisico a priori ma si basa su un idea diversa, ovvero un passo alla volta. Il messaggio viene inviata alla prima centrale di commutazione, la quale dopo averlo esaminato per vedere gli eventuali errori, lo ritrasmette alla successiva fino ad arrivare al destinatario. Questa tecnica è chiamta di **store and forward**.

Commutazione di pacchetto

Questa tecnica è molto diversa dalle precedenti e si basa sull'idea di dividere i dati in pacchetti limitati i quali partono e possono arrivare anche in ordine sparso sarà compito del destinatario riordinarli. Non c'è bisogno di alcun collegamento predefinito e ogni pacchetto può percorrere strade diverse. E' più resistente agli errori della commutazioni di circuito , poiché si possono aggirare commutatori bloccati passando per un altro percorso. Inoltre la commutazione di pacchetto non riserva alcuna ampiezza di banda per cui in linea generale è più efficiente. L'addebito dipende sia dal tempo che dalla distanza.

Caratteristica	Com. Circuito	Com. Pacchetto
Instaurazione chiamata	Richiesta	Non richiesta
Percorso fisico dedicato	Si	No
Ogni pacchetto segue la stessa strada	Si	No
I pacchetti arrivano in ordine	Si	No
Il guasto dello switch è fatale	Si	No
Banda disponibile	Fissa	Dinamica
Instante di congestione	Avvio connessione	A ogni pacchetto
Banda sprecata	Si	No
Store and Forward	No	Si
Trasparenza	Si	No
Tariffa	A minuto	A pacchetto

Figura 9: Confronto tra commutazioni

1.5 Sistema telefonico mobile

Esistono 3 generazioni di telefoni cellulari:

1. Voce analogica
2. Voce digitale
3. Voce e dati digitali

1.5.1 Cellulari di I generazione

Il primo esempio di 'cellulare' lo si ha nel 1946 quando venne creato il sistema premi e parla, come ad esempio quella dei CB. Negli anni sessanta scompare il tasto per parlare grazie all' IMTS (*Improved Mobile Telephone System*) che utilizzava un trasmettitore ad alta potenza posto in una collina, il quale utilizzava 2 frequenze, una per la ricezione e una per trasmettere. IMTS utilizzava solo 23 canali distribuiti tra 150 e 450 MHz. Il numero limitato di canali faceva si che alcuni utenti dovevano aspettare molto prima di aver segnale libero.

Sistema telefonico mobile avanzato

Cambiò tutto grazie a AMPS (*Advanced MPS*). Ogni area geografica era divisa in **celle**, in AMPS grandi 10-20 Km. Ogni cella utilizzava un insieme di frequenze diversa da quelle vicine. L'utilizzo di celle piccole richiede meno potenza. L'idea principale sta proprio qui, in celle piccole e grande riutilizzo delle frequenze. Nelle aree in cui il numero di utenti è elevato e il sistema

tende a sovraccaricarsi ,le celle vengono a loro volta divise in **microcelle** così da aumentare il riuso delle frequenze. Tanto più piccole sono le celle tanto meno potenti devono essere i dispositivi. Da notare il fatto che una frequenza utilizzata da una cella non è più usata nell'area cuscinetto attorno ad essa (area di circa 2 celle). Al centro di ogni cella si trova una stazione la quale è collegata a un dispositivo chiamato MTSO (*Mobile Telephone Switching Office*) o MSC(Mobile Switching Center). In sistemi più grandi sono necessari più MTSO che quindi vengono divisi in livelli. Ogni MTSO colloquia con gli altri. Un telefonino in ogni istante è logicamente posizionato in una certa cella e ogni qualvolta il segnale in tale cella si affievolisce, la stazione base colloquia con le adiacenti per delegare la gestione dell'apparecchio alla cella col segnale più forte. Questo processo è chiamato **handoff** e richiede 30 msec. Esistono 2 tipi di handoff:

- soft handoff: l'acquisizione della nuova stazione avviene prima di interrompere il segnale precedente.
- hard handoff: la vecchia stazione rilascia il telefono prima che la nuova lo acquisisca. La chiamata viene bruscamente interrota.

Gestione della chiamata

Ogni telefono AMPS ha un numero seriale di 32 bit e un numero di telefono 10 cifre. Ogni volta che viene acceso, il telefono esplora i vari canali e trova il segnale più potente. Il telefono quindi trasmette in broadcast il proprio seriale e il numero di telefono con un codice di correzione degli errori. La stazione base aggiorna l' MTSO e ogni 15 minuti circa aggiorna la posizione corrente. Per chiamare il telefono acceso invia il numero del chiamato e i propri dati attraverso il canale di accesso e quando riceve la richiesta la stazione base informa l'MTSO. Se il chiamante appartiene a quell'MTSO cerca un canale libero per la chiamata, e trasmette il numero del canale al telefono. Il processo di ricezione è diverso: ogni telefono è in ascolto nel canale di trasferimento e quanto l'MTSO riceve il pacchetto che richiede il destinatario lo passa alla stazione base la quale chiede conferma al telefono. In caso affermativo la stazione invia il numero del canale con la chiamata e inizia la conversazione.

1.5.2 Cellulari di II generazione

Nel mondo sono sostanzialmente utilizzati 4 sistemi: D-AMPS, GSM, CDMA e PDC utilizzato solo in Giappone e molto simile al D-AMPS.

D-AMPS (*Digital* AMPS)

Il D-AMPS è totalmente digitale. Progettato per coesistere con AMPS utilizza gli stessi canali a 30 KHz con le stesse frequenze. Si è resa disponibile una nuova banda di frequenza 1850-1910 MHz per sostenere l'aumento del carico. Alcuni cellulari erano in gradodi utilizzare entrambe le bande disponibili. Su un telefono D-AMPS il segnale voce preso dal microfono viene digitalizzato e compresso dal **vocoder**, questa compressione permette la condivisione di una coppia di frequenze (upstream/downstream) fino a 3 utenti con multiplexing a divisione di tempo. Ogni coppia di frequenza supporta 25 frame/sec di 40 msec. Ogni frame è diviso in 6 slot temporali. Gruppi di 16 frame costituiscono un superframe, con alcune informazioni di controllo. Concettualmente funziona come AMPS: viene acceso il telefono, viene contattata la stazione e poi rimane in ascolto. Nei tempi in cui il cellulare non riceve ne trasmette viene testata la qualità della linea. Questa tecnica è chiamata **MAHO**(*Mobile Assisted HandOff*)

Comunicazioni GSM *(Global System for Mobile communications)*

Molto simile ad D-AMPS però ha i canali più ampi così possono supportare ben 8 utenti in una coppia di frequenze. Un sistema GSM ha 124 coppie di canali simplex ampi 200 KHz e supporta 8 connessioni grazie a multiplexing a divisione di tempo. A ogni stazione attiva è assegnato uno slot temporale su una coppia di frequenze. Quindi teoricamente supporta 992 canali molti di questi però utilizzati come canali di controllo. Trasmissione e ricezione non avvengono nello stesso intervallo di tempo perché il sistema non è in grado di gestirlo. Questo protocollo ha introdotto le schede SIM che contengono al loro interno IMSI (identifica la SIM) e la chiave di crittografi (Ki). L'identificazione avviene così:

1. Il cellulare manda Ki e IMSI in broadcast
2. L'operatore lo riceve e manda un numero casuale
3. Il cell lo rimanda firmato con Ki
4. L'operatore controlla

Il **canale di controllo broadcast** è un flusso continuo di dati trasmessi dalla stazione base che annuncia identità e stato del canale. Il **canale di controllo dedicato** è utilizzato per aggiornare la posizione, registrare il terminale nella rete e configurare la chiamata. Infine c'è un **canale di controllo comune** che è diviso in 3 sottocanali logici. Il primo è il **canale di paging** utilizzato

dalla stazione per annunciare le chiamate in arrivo. Poi c'è il **canale ad accesso casuale** e permette agli utenti d richiedere uno slot sul canale di controllo dedicato. Infine c'è il **canale di assegnazione dell'accesso** che assegna il lo slot del canale di controllo dedicato per ripetere le richieste efettuate dal secondo canale.

CDMA *(Code Division Multiple Access)*

Miglior sistema rispetto a quelli presentati e base per la III generazione a volte chiamato **cdmaOne**. CDMA permette la trasmissione per tutto il tempo attraverso l'intero spettro. Queste trasmissioni multiple simultanee vengono separate tramite tecnica di codifica. L'idea sta nel fatto che i segnali si sommino linearmente. Per cui tutti comunicano ma ogni coppia lo fa in 'lingua diversa'. Per risalire a ciò che viene detto basta togliere il rumore aggiunto dalle conversazioni di altri. Per riuscire a filtrare tale segnale rumoroso vengono utilizzate le matrice di Hadamard. Tecnicamente il CDMA funziona così:
ogni tempo di bit è diviso in m intervalli chiamati **chip** (generalmente 64-128 chip per bit) e ad ogni stazione viene assegnata una **sequenza di chip** univoca. Per trasmettere un 1 la stazione deve semplicemente inviare tale sequenza se invece invia uno zero deve farne il complemento. Non sono ammessi altri schemi. Per aumentare la quantità di informazione inviabile basta passare da b bit/sec a mb chip/sec aumentando l'ampiezza di banda di un fattore m. CDMA è una forma di comunicazione a spettro distribuito. Ognuna di queste sequenze di chip sono mutualmente ortogonali, ovvero ogni prodotto interno normalizzato di qualunque coppia di sequenze è uguale a 0 (ottenute con i **codici Walsh**).

$$S * T = \frac{1}{m} \sum_{i=1}^{m} S_i T_i = 0$$

Da cui si deduce che S*S = 1 e S*$\overline{S}$ = -1. Ora, ogni stazione invia queste sequenze come bit le quali si 'mischiano' con le altre, tecnicamente si sommano con gli altri segnali di altre stazioni. Una volta che il segnale arriva alla stazione di destinazione per sapere il bit inviato dalla sorgente basterà moltiplicare il segnale per la sequenza di chip della sorgente e si otterrà il bit inviato. Matematicamente se si deve capire il messaggio C allora sarebbe **S = A + $\overline{B}$ + C**:

$$S * C = (A + \bar{B} + C) * C = A * C + \bar{B} * C + C * C = 0 + 0 + 1 = 1$$

Per la proprietà di ortogonalità tutti i prodotti si sono annullati a parte quello interessato. Questo sistema è in genere utilizzato per reti wireless.

1.5.3 Cellulari di III generazione

La prima proposta di cellulari di III generazione fu fatta dall'ITU con l'intenzione di lanciarli nell'anno 2000 con ampiezza di banda di 2 MHz e 2Mbps per tutti. Un sogno irrealizzabile che ha visto il tutto slittare qualche anno più avanti e con alcune specifiche smussate come i 2Mbps per chi stava fermo e circa 400 per gli utenti che camminavano e 144 per quelli che si spostano a più alte velocità. I servizi che avrebe dovuto fornire IMT-2000 (così si chiamva la proposta) erano:

- trasmissione voce ad alta qualità;
- trasmissione messaggi;
- applicazioni multimediali;
- accesso internet.

Per riuscire ad utilizzare questi servizi in tutto il mondo si era pensati di creare un unica tecnologia per rendere tutto più semplice. Furono fatte diverse proposte.

W-CDMA (*Wideband CDMA*)

Questo protocollo fu proposto da Ericsson. In Europa battezzato col nome **UMTS** *(Universal Mobile Telecommunications System)*. Si basa sui fondamenti del CDMA utlizzando però una banda larga a 5 MHz ed è stato progettato per interagire con il sistema GSM anche se non compatibile. Data rate 384 Kbps.

CDMA2000

Proposto da Qualcomm, simile al precedente con la differenza di non interagire con GSM. Altre differenze col precedente sono il tempo di frame, di spettro e una diversa tecnica di sincronizzazione. Data rate 144 Kbps.

EDGE *(Enhanced Data for GMS Evolution)*

Uguale al GSM con un numero maggiore di bit per baud che comportano però più errori per baud. Sistema pensato durante il passaggio da II a III generazione, definito infatti 2.5G.

GPRS *General Packet Radio Service*

Altro schema per 2.5G. Una rete di pacchetti costruita sopra ad D-AMPS e GSM. GPRS permette di inviare e ricevere pacchetti IP in una cella basata su sistema vocale. Quando attivo alcuni slot temporali vengono dedicati al traffico dei pacchetti. Questi slot sono divisi in canali logici. Ogni canale è utilizzato per scaricare i pacchetti nei quali c'è indicato il destinatario. Per inviare un pacchetto la stazione mobile richiede uno o più slot e effettua la richiesta alla base. La base poi invia il pacchetto via internet tramite rete via cavo.

1.5.4 Oltre al 3G

HSDPA *(High Speed Downlink Packet Access)*

HSUPA *(High Speed Uplink Packet Access)*

HSOPA *(High Speed OFDM Packet Access)*

2 Lo strato Data Link

2.1 Progetto dello strato

Funzioni principali:

1. Definire interfaccia per lo strato network
2. Gestione errori
3. Regolare il flusso

Per fare questo i pacchetti che arrivano dallo strato network vengono incapsulati in frame, con un header, un corpo e una coda.

2.1.1 Servizi

La funzione di questo strato è fornire servizi allo strato network. Il servizio principale è quello di consegnare i dati pervenuti dallo strato network allo stesso del destinatario. Tre servizi vengono comunemente forniti:

1. Servizio unacknoledged senza connessione: una macchina invia dei frame e non è necessaria una connessione dedicata e nemmeno una risposta e/o conferma del destinatario. Se il pacchetto viene perso non viene fatto nulla. Utile per canali con bassa percentuale di errori o per trasmissioni voce in cui il ritardo è peggiore di un errore.
2. Servizio acknoledged senza connessione: anche qui non è necessaria una connessione, ma ogni frame è inviato singolarmente e si attende una conferma, entro un limite di tempo massimo, del destinatario. Se il limite è superato si reinvia il frame. Servizio utile per reti poco affidabili come le reti wireless.
3. Servizio acknoledged orientato alla connessione: c'è bisogno di connessione tra le parti, poi si inizia a inviare. Ogni frame viene numerato. Tramite ACK viene garantita la ricezione e grazie alla numerazione si cerca di rilevarli nell'ordine esatto. Alla fine la comunicazione viene chiusa rilasciato i relativi buffer, variabili e risorse.

2.1.2 Frame

L'approccio di questo strato già menzionato è la suddivisione del flusso di bit in frame e calcolarne il cheksum il quale viene ricalcolato dal destinatario che controlla la corrispondenza, in caso contrario si è verificato un errore e

vengono presi i provvedimenti necessari. Un problema non banale è capire la suddivisione in frame, ovvero dove finisce o inizia un frame. Esistono 4 modi principali per farlo:

1. Conteggio dei caratteri
2. Flag byte con byte stuffing
3. Flag di inizio e fine con bit stuffing

Il **primo metodo** banalmente inserisce nella testa un campo con il numero di caratteri di lunghezza del frame. Il problema principale è l'errore nel conteggio di caratteri durante la trasmissione che sfaserebbe tutto. Richiedere la ritrasmissione non è una soluzione perché il destinatario non capisce quanti caratteri deve saltare. Il **secondo metodo** aggira il problema inserendo un byte prima e dopo ogni frame, chiamato **flag byte**. Quindi in caso di perdita di sincronizzazione basterà cercare questo byte speciale. Un possibile problema è che all'interno dei dati ci sia un flag byte. Per risolverlo basta che la sorgente in tali casi inserisca un **byte di escape** (*ESC*) subito prima di ogni occorrenza e la destinazione provvederà a toglierli (**destuffing**). Questa tecnica è chiamata **byte stuffing** (o **character stuffing**). Paradossalmente un ESC deve essere preceduto a sua volta da un ESC! Il problema di questo metodo è il legame con la decodifica dei caratteri con 8 bit che non è sempre vera. Il **terzo metodo** applica una tecnica simile alla precedente: ogni frame comincia e finisce con un gruppo speciale di bit, 01111110, in sostanza un flag byte. Ogni volta che lo strato Data link della sorgente incontra cinque 1 inserisce uno 0 subito dopo. Questa tecnica è detta **bit stuffing**. Il destinatario ovviamente incontrando cinque 1 e uno 0 non deve far altro che eliminare lo 0 per ottenere il messaggio originale.

2.1.3 Controllo degli errori

Per riuscire a rilevare banalmente errori nella trasmissione dei dati principalmente le cose da fare sono:
ogni frame deve avere un numero di sequenza in modo che la destinazione accetta solo i frame che non ha già ricevuto, la sorgente ha un timer utile per quando attende che l'ACK ritorni dal destinatario in modo da non fare attesa infinita. Se non riceve ACK in un tempo limite stabilito dal protocollo il messaggio viene reinviato.

2.1.4 Controllo del flusso

Esistono principalmente 2 tecniche per la gestione del flusso (evitando così sovraccaricamenti del desinatario):

1. Tramite Feedback: la destinazione manda alla sorgente informazioni per darle il permesso di inviare altri dati o comunque per informarla dello stato in cui si trova.

2. Tramite limitazione di velocità: il protocollo contiene al suo interno un meccanismo che limita la velocità della sorgente senza utilizzo di feedback.

2.2 Rilevazione e correzione degli errori

2.2.1 Codici per la correzione degli errori

Esistono principalmente 2 tipi di decodifica: a semplice rilevazione di errori e a correzione di errore. Entrambe consistono nell'aggiunta di informazioni ridondanti per rilevare e nel secondo caso correggere gli errori. Le 2 tecniche vanno applicate ovviamente in ambiti differenti, la prima più indicata per trasmissioni veloci tipo con fibra, mentre la seconda per trasmissioni più rumorose come quelle wireless.
Come avviene l'identificazione di un errore: in primo luogo indichiamo con n la lunghezza di una **codeword** formata da m bit per il frame e r per il controllo (n = m+r). Confrontando ora 2 codeword di lunghezza uguale facendo semplicemente lo XOR troviamo tutti 0 se le parole coincidono oppure alcuni 1. La quantità di 1 presenti corrisponde alla **distanza di Hamming**. Una distanza di Hamming d rappresenta il numero di errori per convertire una sequenza nell'altra. Essendo che non tutte le possibili 2^m codeword sono accettabili, un modo per intuire la codeword esatta è enumerare tutte le possibili codeword e scegliere quelle con distanza di Hamming minima. In base a alla distanza d+1 della codifica si riescono a trovare d errori. Mentre per correggerne d ci vorrebbe una codifica con distanza 2d + 1. Un esempio lo è il bit di parità.

Codifica di Hamming

Vengono inseriti nella sequenza di bit da inviare bit di parità posti nelle posizioni che sono una potenza di 2. Questi bit di parità sono legati ad alcuni bit di dati con la seguente regola: il bit di dati k-esimo è controllato da quei bit la cui somma forma k. Per esempio k = 11 = 1 + 2 + 8, allora il primo, secondo e ottavo bit sono quelli di parità per k. Quindi se troviamo errori di inversione nei bit 1,2 e 8 allora l'11 è invertito. Questa codifica corregge solo errori singoli. Esiste un trucco nell'invio di dati per correggere gli errori improvvisi, detti **burst**, che è quello di inviare le varie codeword sotto forma matriciale inviando i dati colonna per colonna in modo che gli

errori burst siano solamente singoli bit per ogni riga e si sa che tramite hamming si possono correggere.

2.2.2 Codifiche a rilevazione d'errore

CRC *(Cyclic Redundance Check)*

L'idea di base è quella di trattare le sequenze di bit come coefficiente di polinomi. Si applica per cui l'aritmetica dei polinomi in modulo 2. Quindi addizioni e sottrazioni sono fatte con lo XOR. Un divisore sta in un dividendo se ha gli stessi bit. Quando si utilizza CRC la sorgente e la destinazione devono accordarsi su un **polinomio generatore** *G(x)* che deve avere come primo e ultimo bit 1. Il frame da controllare, che corrisponde al polinomio *M(x)*, deve essere di ordine maggiorne di G(x). L'idea è quella di aggiungere un checksum alla fine del frame in modo che il polinomio rappresentato dal frame col checksum sia divisibile per G(x). Per cui la destinazione riceve il polinomio e prova a dividerlo per G(x). Se c'è resto allora c'è errore.
Il calcolo del checksum avviene così:

1. Posto r il grado di G(x), aggiungere r zeri alla fine del frame, così da avere m+r bit e corrisponde al polinomio $x^r M(x)$
2. Dividere la sequenza corrispondente a $x^r M(x)$ per quella di G(x) usando la divisione modulo 2.
3. Sottrarre il resto, al massimo r bit, dalla sequenza corrispondente $x^r M(x)$, sottrazione in modulo 2. Il risultato sarà *T(x)*

2.3 Protocolli Data Link elementari

2.3.1 Protocollo stop-and-wait

Protocollo molto semplice per il controllo del flusso e si può utilizzare in canali simplex o half duplex. Quando il mittente invia un blocco aspetta che il ricevente invii una conferma (ACK). Lo svantaggio è l'attesa della risposta, ma in compenso non c'è bisogno di regolare la velocità. Gli errori possibili ora sono 2:

1. Sul frame: ovvero non arriva mai a destinazione e quindi il mittente aspetta all'infinito. C'è bisogno di un timeout come detto precedentemente.

2. Sul ACK: la conferma non arriva al mittente, il quale al termine del time out reinvia il pacchetto. Al destinatario arriva 2 volte ma grazie al numero di pacchetto (vedi su) il pacchetto non viene preso in considerazione.

2.4 Protocolli sliding window

Un protocollo per il controllo di flusso sicuramente più efficiente di quello appena presentato. Un dettaglio sicuramente a suo favore è l'utilizzo della tecnica di **piggybacking**, che consiste nello sfruttare un messaggio del destinatario al mittente come 'passaggio' per il messaggio di conferma ACK, in modo da non perdere tempo e sfruttare maggiormente il canale di comunicazione. Il campo *ack* è posto nella testa del frame. Sorge il problema di quando fare piggybacking, un attesa troppo lunga può rendere vano il tutto perché il mittente fa un reinvio del frame. Quindi se il pacchetto arriva velocemente viene fatto piggybacking dell'ACK altrimenti si invia l'ACK separatamente. L'essenza del protocollo è che ogni partecipante alla comunicazione deve tener sotto controllo 2 finestre: quella dei frame in entrata e quella dei frame in uscita. Ogni frame in uscita contiene un numero di sequenza e il destinatario deve tener traccia di questi per la ricezione mentre il mittente per l'invio.

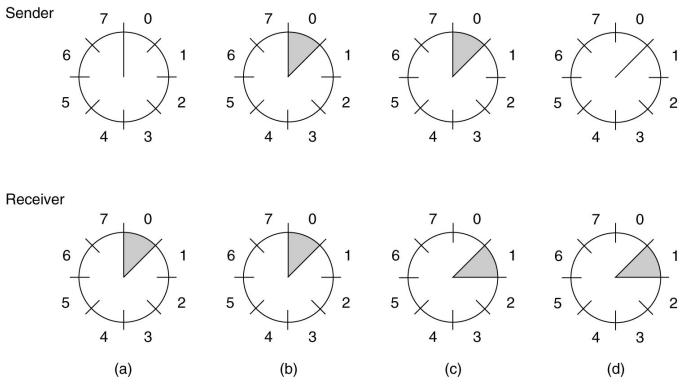


Figura 10: Esempio sliding window: (a) Inizialmente, (b) Dopo l'invio del frame 0, (c) Dopo la sua ricezione, (d) Dopo l'invio dell'ACK.

2.4.1 Protocollo sliding window a 1 bit

La finestra di controllo ha dimensione 1 e viene utilizzato il metodo stp-and-wait. Quando il mittente invia un frame resta nella finestra finché non viene

ricevuto l'ACK corrispondente prima di aggiornare la finestra. I frame inviati sono numerati con 1 o 0. Quando il destinatario riceve il frame controlla che il numero sia uguale a quello che aspettava, se si invia l'ACK. Se l'ACK contiene il numero che la sorgente si aspettava allora continua a inviare un nuovo pacchetto altrimenti reinvia quello segnato nel buffer. Si può utilizzare anche la tecnica **pipelining**, ovvero vengono inviati più frame contemporaneamente prima di entrare in attesa. Il destinatario aggiorna la finestra non appena riceve il frame e invia l'ACK. Esistono 2 approcci per contro i problemi di trasmissione durante il pipelining: uno chiamato **go back n** che semplicemente scarta tutti i frame dopo quello danneggiato, l'altra è la **ripetizione selettiva** nel quale vengono tenuti i frame buoni e non scartati e vengono messi in un buffer. Quando la sorgente va in timeout solo quello senza ACK viene rispedito. La destinazione quando trova un errore non sta ferma, ma invia un NAK in modo da stimolare la ritrasmissione prima del timeout.

2.4.2 Protocollo che usa go back n

Qui la finestra di invio ha dimensione > 1 mentre quella di ricezione è uguale a 1. I pacchetti arrivano uno alla volta e su di essi viene fatto il checksum, se si trovano errori vengono segnalati alla sorgente indicando il numero del pacchetto danneggiato, per questo la finestra sorgente deve essere capiente. Se la finestra sorgente si riempie prima che il timer scatti, la pipeline viene svuotata. Le conferme vengono sempre mandate in piggybacking. La destinazione nel frattempo scarta i pacchetti successivi a quello avente l'errore. Questo approccio può far perdere molta banda se la frequenza degli errori è alta.

2.4.3 Protocollo che usa ripetizione selettiva *(selective repeat)*

In questo caso il buffer della destinazione deve essere più capiente. Nel caso di errori viene inviato alla sorgente un NACK (*Not ACK*) indicandone il pacchetto. Finché il pacchetto errato non arriva nuovamente dalla sorgente i pacchetti successivi vengono posti nel buffer. Una volta arrivato tutto viene passato allo strato Network. *Nota*: la sorgente dispone di timer per cui se non arrivasse il NACK il pacchetto viene reinviato comunque.

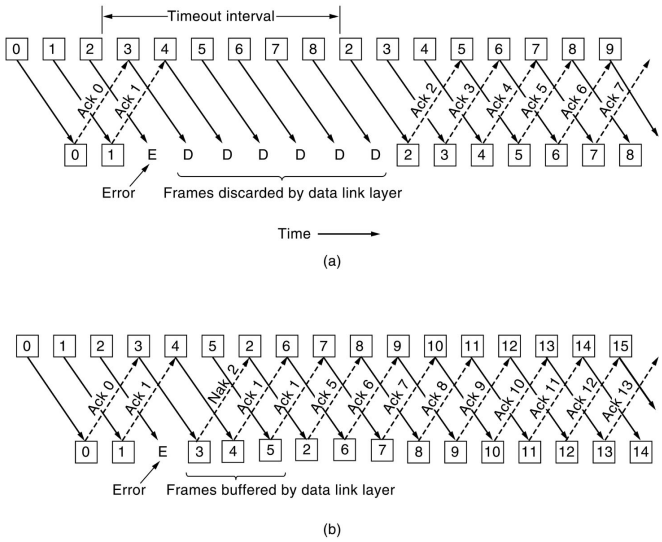


Figura 11: Esempio sliding window con *go back n*: effetti del pipelining.

2.5 Protocolli data link

2.5.1 HDLC (High-level Data Link Control)

Deriva dal protocollo SDLC della IBM con le varianti LAP e LAPB. E' un protocollo orientato ai bit e usa il bit stuffing. La struttura dei frame di questo protocollo è: FCS indica la parte di checksum eseguita con CRC, Flag(7E) è il flag 01111110, mentre il Control con le Sliding Window e determina 3 tipi di frame:

1. **Information**: formato da uno 0 seguito dal numero del frame. Poi p/f è utile per le conversazioni multiple e infine ci sono gli ACK (piggybacking sul frame successivo).

2. **Supervisory**: formato da uno 10 seguito dal campo type che può assumere 4 valori:

   (a) Type 0: Frame ACK (Receive Ready)

   (b) Type 1: Frame NACK (Reject)

   (c) Type 2: Canale congestionato (Receive NOT Ready)

   (d) Type 3: NACK selettivo (Selected Reject)

   Il resto come il precedente.

3. **Unnumbered**: formato da uno 11. Viene usato nel caso di perdita di frame.

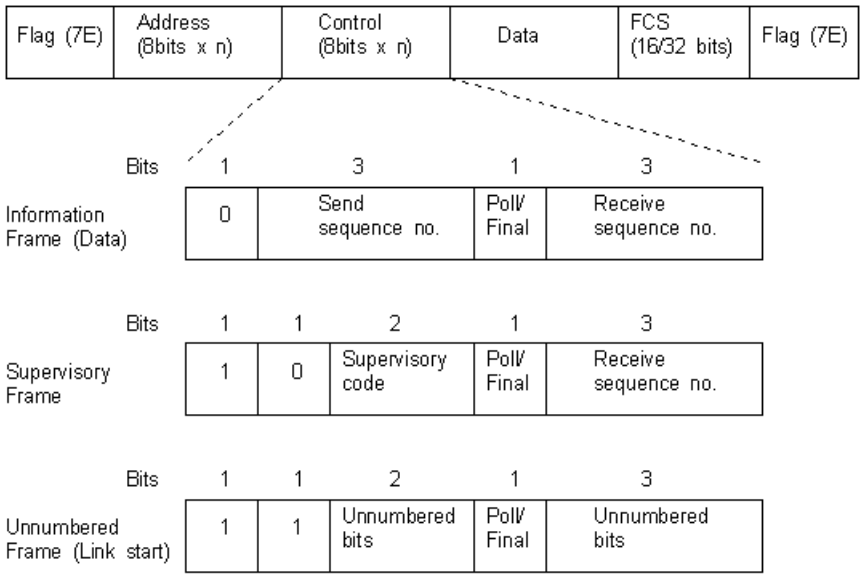


Figura 12: Struttura frame HDLC

I principalei comandi sono:

- DISC: blocca la connessione
- SNRM: nuova macchina on-line
- SAMB: crea una connessione bilanciata
- FRMR: rifiuta un frame di controllo

2.5.2 PPP *(Point-to-Point Protocol)*

Usato nei collegamenti punto a punto tra router e utente, ovvero in internet. Tra le varie funzioni di PPP troviamo: rilevazione degli errori, supporto per più protocolli, possibilità di negoziazione IP, possibilità di effettuare autenticazione. Tra le caratteristiche di questo protocollo meritano menzione:

1. Metodo di framing che permettere di limitare i vari frame in modo non ambiguo e il formato del frame permette la rilevazione degli errori
2. Protoccolo di collegamento per gestire la connessione, i test, le negoziazioni e la gestione pulita della disconnessione. (LCP)
3. Modalità per negoziare le opzioni relative allo strato Network.

Utilizza 2 protocolli speciali:

1. **LCP** *(Link Control Protocol)*: insieme di comandi per la gestione del flusso e della comunicazione
2. **NCP** *(Network Control Protocol)*: si occupa del dialogo con lo strato 3

La differenza più importante dall'HDLC è l'utilizo del byte stuffing. Il frame PPP è così composto:

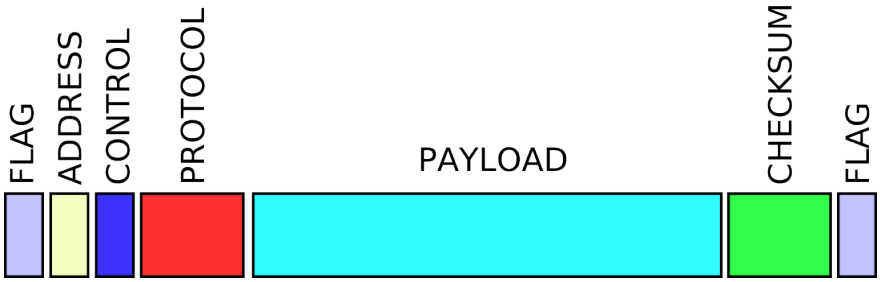


Figura 13: Struttura frame PPP

- **FLAG**: flag byte tipico 01111110
- **Address**: non viene usato, tutti 1
- **Control**: posto per default a 00000011, ovvero niente controllo
- **Protocol**: segnala il tipo di pacchetto che si invia
- **Payload**: campo dati
- **Checksum**: usa CRC
- **FLAG**: di nuovo flag byte tipico

3 Il sottostrato MAC *(Medium Access Control)*

Il MAC si occupa di assegnare l'uso di un canale di comunicazione multiaccesso nelle reti broadcast. Per fare ciò i tradizionali metodi FDM o TDM non servono poiché sono statici. In questo particolare tipo tipo di reti dobbiamo tener presente 5 fondamentali premesse:

1. **Modello della stazione**: dette anche terminali, si occupano dell'invio dei frame. Una volta generato il frame la stazione resta bloccata fino a che il frame non arriva a destinazione con successo.

2. **Canale singolo**: esiste un solo canale che assicura la comunicazione.

3. **Collisioni**: 2 frame trasmessi contemporaneamente si sovrappongano distorcendo il segnale finale. Tutte le stazioni possono rilevare una collisione. Un frame soggetto a collisione deve essere ritrasmesso.

4. (a)**Tempo continuo**: non esiste temporizzazione tra le stazioni, ognuna può trasmettere quando vuole.
   (b)**Tempo diviso in intervalli**: il tempo è diviso in intervalli, detti slot, e ogni frame viene trasmesso all'inizio dell'intervallo.

5. (a)**Occupazione del canale verificabile**: *(Carrier sense)* prima di tentare la trasmissione, la stazione controlla che il canale sia libero.
   (b)**Occupazione del canale non verificabile**: le stazioni non sono in grado di controllare il canale quindi si limitano a trasmettere.

3.1 Il protocollo ALOHA

3.1.1 ALOHA puro

Sistema a contesa ideato nelle isole Hawaii per trasmettere via radio segnali broadcast a varie isole. L'idea di base è che ognuno trasmette frame ogniqualvolta ha bisogno di farlo. Una volta inviato il frame la sorgente si mette in ascolto attendendo un feedback, ovvero, attende di capire se il frame è arrivato a destinazione altrimenti aspetta un periodo di tempo casuale e lo ritrasmette. Solitamente si attua utilizzando un meccanismo di back-off, secondo il quale la ritrasmissione viene effettuata dopo un ritardo selezionato casualmente compreso tra 0 e (K-1)T, dove T è il tempo di trasmissione del messaggio e K può eventualmente dipendere dal numero di collisioni già avvenute. Ora bisogna chiedersi quale sia la probabilità di collisione utilizzando

questa tecnica. Supponendo che la generazione di frame sia in accordo con la distribuzione di Poisson come lo è anche la probabilità di k tentativi di trasmissione per tempo di frame e vale:

$$Pr[k] = \frac{G^k e^{-G}}{k!}$$

perciò la probabilità di 0 frame è e^{-G}. Il numero medio di frame generati è 2G. La probabilità che nessuno trasmetta mentre è in circolo il frame è:

$$S = Ge^{-2G}$$

Si può quindi utilizzare al massimo circa il 18% della banda. Non molto incoraggiante.

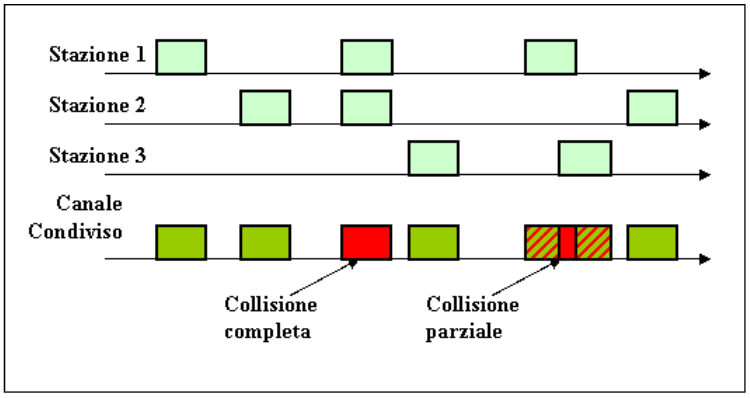


Figura 14: Aloha Puro: collisione dei frame

3.1.2 Slotted ALOHA

Questo metodo prevede che il tempo venga diviso in intervalli discreti che rappreentano un frame, detti slot. Gli utenti quindi devono concordarsi sui limiti degli intervalli in modo da sincronizzarsi, questo può essere fatto con una particolare stazione che segnala l'inizio di ogni intervallo, tipo un orologio. Così facendo il periodo vulnerabile è dimezzato, poichè ho i frame collidono completamente oppure non collidono. Per cui ora si ha:

$$S = Ge^{-G}$$

L'efficienza è quindi raddoppiata portando l'utilizzo del canale a circa un 36

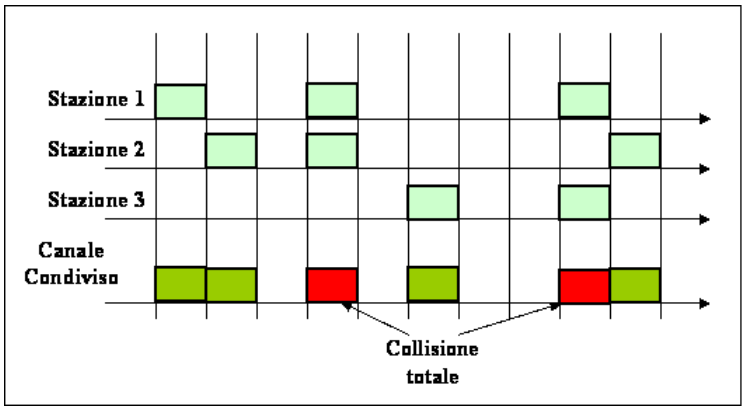


Figura 15: Slotted Aloha: collisione dei frame

3.2 CSMA *(Carrier Sense Multiply Access)*

E' un **protocollo con rilevamento della portante** e ne esistono diverse varianti.

3.2.1 CSMA 1-persistente

Quando una stazione deve trasmettere dei dati, prima di farlo ascolta se il canale è occupato. Se si la stazione aspetta che si liberi altrimenti invia un frame; in caso di collisione, la stazione resta in attesa per un tempo casuale prima di ritentare la trasmissione. Il nome 1-persistente viene dal fatto che la probabilità di trasmissione a canale libero è 1. Purtroppo le collisioni continuano a sussistere a causa del ritardo di propagazione. A dire il vero anche con tempo di propagazione 0 le collisioni ci sarebbero comunque. Nonostante questo il protocollo risulta migliore dell'Aloha puro.

3.2.2 CSMA non persistente

L'approccio di questo protocollo è leggermente diverso. La stazione se trova il canale libero invia una serie di frame altrimenti al posto di rimanere all'ascolto per approfittare della liberazione del canale attende per un tempo casuale dopo il quale controlla il canale. Si ha un miglior utilizzo del canale ma con maggior ritardo.

3.2.3 CSMA p-persistente

Si applica ai canali divisi in intervalli di tempo e funziona così: ogni stazione quando vuole trasmettere controlla il canale. Se lo trova libero trasmette con

una probabilità p e rimanda all'intervallo successivo con probabilità q = *1-p*. Il processo si ripete fino alla trasmissione del frame o qualche altra stazione trasmette, in questo caso attende un tempo casuale come se ci fosse stata una collisione.

3.2.4 CSMA con rilevamento delle collisioni

Sicuramente i metodi appena presentati sono delle migliorie all'ALOHA, ma si può ottenere risultati ancora migliori se le stazioni una volta accorte di una collisioni fermano bruscamente la trasmissione ormai divenuta inutile. Si ha un risparmio di tempo e banda. Questa idea sta alla base del protocollo **CSMA/CD** *(CSMA Collision Detection)*. Come già detto in caso di collisioni le stazioni fermano la trasmissione e attendono un tempo casuale prima di riprovare. Esistono così 3 stati: trasmissione, inattivo e attesa. Trasmissione quando si è inviato un frame nel canale, attesa quando c'è stata una collisione e inattivo quando non deve trasmettere.

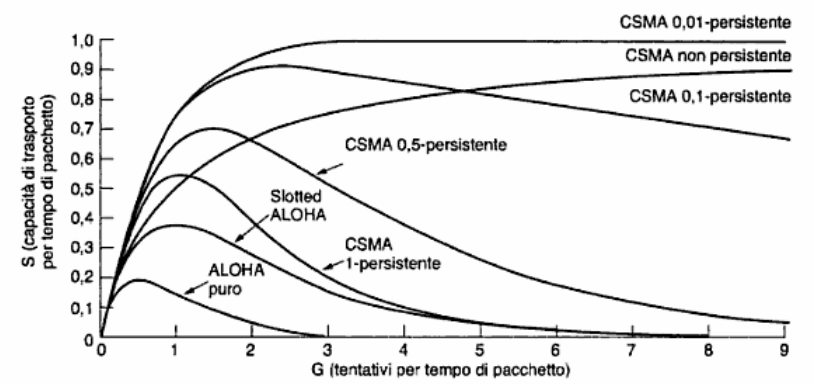


Figura 16: Confronto tra gli utilizzi del canale per protocolli ad accesso casuale.

3.3 Protocolli senza collisione

Questi protocolli partono da 2 presupposti importanti:

1. Ognuna delle N stazioni è etichettata con un indirizzo che va da *0 a N-1*
2. Il ritardo di propagazione è trascurabile

3.3.1 Protocolli a mappa di bit

Protocollo a mappa di bit elementare

In questo protocollo ogni periodo di contesa e diviso in N intervalli. Se la stazione 0 deve inviare un frame, trasmette un bit 1 durante l'intervallo 0. Durante questo intervallo solo la stazione 0 può trasmettere. In generale quindi la stazione i-esima invia un 1 nell'intervallo i-esimo quando ha un frame accodato da inviare, incurante del fatto che ci siano altre stazioni pronte a farlo. Una volta trascorsi gli N intervalli tutti sanno chi deve trasmettere e in ordine numerico iniziano a farlo. Una volta finite le trasmissioni tutto ricomincia daccapo. Questo evita completamente le collisioni! Questo tipo di protocolli è chiamato **a prenotazione**.

Conteggio Binario

Il problema del protocollo precedente è che in reti con migliaia di stazioni la gestione del bit di controllo non è semplice. Un alternativa è che quando una stazione vuole trasmettere invia il proprio indirizzo sotto forma di stringa binaria partendo dal bit più significativo. Questi bit vengono uniti in OR, quindi dopo ogni serie di bit si ha un risultato che ha la funzione di fermare nella contesa tutte quelle stazioni avente il bit in quella posizione minore. Alla fine si avrà il vincitore che coincide con la stazione con indirizzo più alto. Per evitare monopolizzazione del canale si possono utilizzare priorità a rotazione.

3.3.2 Protocolli a contesa limitata

Sono protocolli che che combinano le proprietà dei protocolli a contesa per ritardare il ritardo e i protocolli senza collisione per aumentare l'uso del canale.

Adaptive Tree Walk

Questo protocollo si basa su un idea tanto particolare quanto interessante. Pensiamo alle stazioni come foglie di un albero binario. Ora al primo intervallo di contesa dopo una trasmissione, istante 0, tutti i nodi cercano di trasmettere. Se uno ci riesce tutto ok, altrimenti se c'è una collisione all'istante 1 provano a trasmettere solo quelle stazioni sotto al nodo 2 (figlio sx della radice). Se una di queste trasmette allora all'istante 2 tocca alle stazioni sotto il nodo 3 (figlio dx della radice), altrimenti si continua sul nodo 4 (figlio sx del nodo 2) e così via.

3.4 Protocolli LAN Wireless

In quest tipi di protocollo esistono 2 problemi da considerare:

1. **Problema della stazione nascosta**: quando qualcuno controlla se nel suo raggio di portata non c'è trasmissione in corso, trasmette lui ma lo fa verso una destinazione già occupata da un terzo utente che non stava nel raggio d'azione
2. **Problema della stazione esposta**: quando una volta controllato si trova nel raggio d'azione una trasmissione anche se non verso il destinatario prescelto.

3.4.1 MACA e MACAW

Uno dei primi protocolli progettati per le LAN wireless è stato il MACA *(Multiple Access with Collision Avoidance)*. L'idea di questo protocollo è semplice: il trasmettitore incita il ricevente a trasmettere un piccolo frame in modo da scoraggiare le stazioni nelle vicinanze. La figura qui di sotto aiuta a comprendere il funzionamento. Supponiamo che A invii un frame a B. A

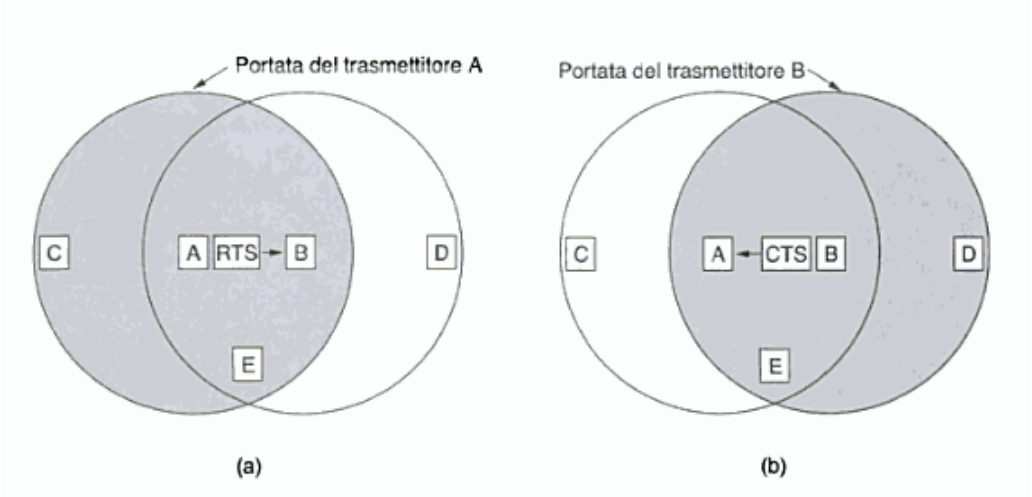


Figura 17: (a) A invia un frame RTS a B. (b) B risponde con un frame CTS.

invia prima di tutto un frame **RTS** *(Request to Send)* e B, che contiene la lunghezza del frame di dati che verrà inviata successivamente. B risponde con un **CTS** *(Clear to Send)* che contiene la lunghezza copiata dal frame RTS, A inizia a trasmettere. Le stazioni vicine ad A stanno in silenzio quando ricevono RTS mentre quelle a B quando ricevono CTS. Esiste una forma migliorata del protocollo MACA chiamato **MACAW** *(MACA per Wireless)*

che sostanzialmente utilizza un frame ACK dopo ogni trasmissione che ha avuto successo.

3.5 Ethernet

E' lo standard delle reti locali e metropolitane *(IEEE 802.3)*. In base al cablaggio esistono diversi tipi di Ethernet:

- **10Base5**: *(thick Ethernet)* cavo coassiale molto grosso, con connessioni generalmente effettuate con spine a vampiro. Il 10 indica che opera a 10 Mbps, la parola 'Base' indica che la trasmissione in banda base. può supportare segmenti lunghi fino a 500 m (il numero 5 indica). Ai cavi è fissato saldamente un **tranceiver** utile pr rilevare le collisioni e per mantenere un stabile contatto con il nucleo del cavo. Un **cavo tranceiver** collega il trasmettitore all'interfaccia installata nel pc ed è costituito di 5 doppini. 1 per i dati in ingresso e 1 per quelli in uscita, 2 per il controllo IN/OUT e 1 per l'alimentazione.

- **10Base2**: *(thin Ethernet)* cavo coassiale più sottile del precedente. I connettori sono BNC standard, che formano giunzioni a T, sono più affidabili e facili da utilizzare. Molto più economico e semplice da installare, ma ogni segmento può essere lungo al massimo 185 m e può supportare non più di 30 macchine. Per trovare guasti in questi mezzi è usata la tecnica **TDR** (*Time Domain Reflectory*) che sostanzialmente misura il ritardo dell'eco dell'impulso immesso nel cavo.

- **10Base-T**: ogni stazione è collegata direttamente a più **hub** con doppini telefonici.

- **10Base-F**: usa fibre ottiche. E' un alternativa costosa ma buona per l'immunità alle interferenze consentendo di collegare edifici/hub molto distanti

3.5.1 Codifica Manchester

Questo tipo di codifica nasce dalla necessità di dover determinare senza ambiguità il punto iniziale, finale e centrale di ogni bit senza impulsi esterni. Questa decodifica divide il periodo di bit in 2 intervalli uguali. Se si deve inviare 1 si tiene un livello di tensione alto nel primo intervallo e basso nel secondo, mentre viceversa se si invia uno 0 (lo standard direbbe esattamente il contrario!!). Questa tecnica aiuta non poco la sincronizzazione di trasmettitore e ricevitore. Esiste inoltre una variante di questa codifica

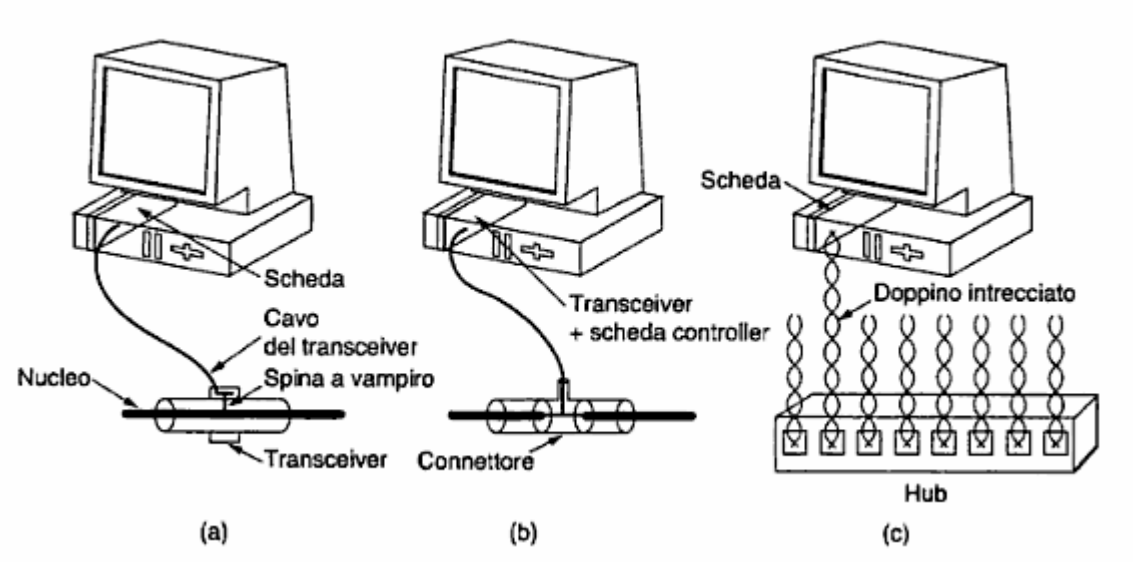


Figura 18: 3 Tipi di cavi Ethernet. (a) 10Base5, (b) 10Base2, (c) 10Base-T.

Nome	Cavo	Lunghezza max. del segmento	Nodi/ segmento	Vantaggi
10Base5	Coassiale spesso (thick Ethernet)	500 m	100	Cavo originale, ora obsoleto
10Base2	Coassiale sottile (thin Ethernet)	185 m	30	Non occorre un hub
10Base-T	Doppino intrecciato	100 m	1.024	Il sistema più economico
10Base-F	Fibra ottica	2.000 m	1.024	Il migliore fra edifici

Figura 19: Tipi di cavi Ethernet

detta **differenziale** che si basa sul cambio di transizione tra intervalli. Se è uno 0 avviene un cambio di transizione altrimenti no. Questo metodo è più immune al rumore anche se più complesso.

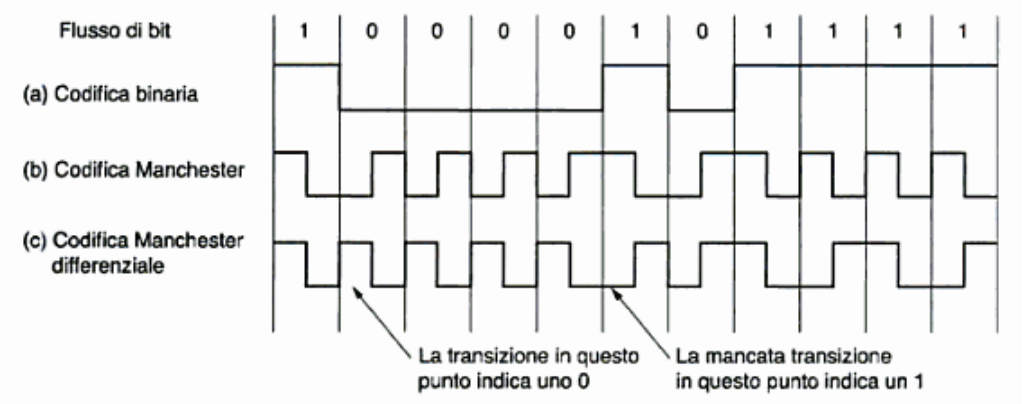


Figura 20: Codifica Manchester

3.5.2 Struttura del frame DIX

Ogni frame DIX inizia con un preambolo di 8 byte, ognuno del tipo 10101010, che in codifica Manchester è un onda quadra. Subito dopo ci sono 6 byte di indirizzo destinatario, il quale primo bit se 0 indica un indirizzo ordinario altrimenti di gruppo se 1. Se l'indirizzo è composto da soli 1 allora la comunicazione è **broadcast**. Subito dopo ci sono altri 6 byte di indirizzo sorgente. I primi 2 bit, il primo con la funzione descritta poco fa e il secondo per distinguere gli indirizzi locali da quelli globali. I restanti 46 sono di indirizzo MAC. I 2 bit di *type* vengono utilizzati dal destinatario per capire cosa farne del frame. Poi ci sono i dati seguiti da un campo pad che serve per riempimento nel caso la parte dei dati abbia un numero di byte inferiore a 46 (utile per il collision detection). Infine c'è la parte di checksum, un codice hash dei dati di 32 bit. In particolare è un CRC.

3.5.3 Indirizzo MAC

MAC-48 gestito da IEEE è formato così: i primi 3 byte identificano il produttore, il cosiddetto OUI *(Organization Unique Identifier)*, e gli ultimi 3 sono lo spazio di indirizzi dell'organizzazione.

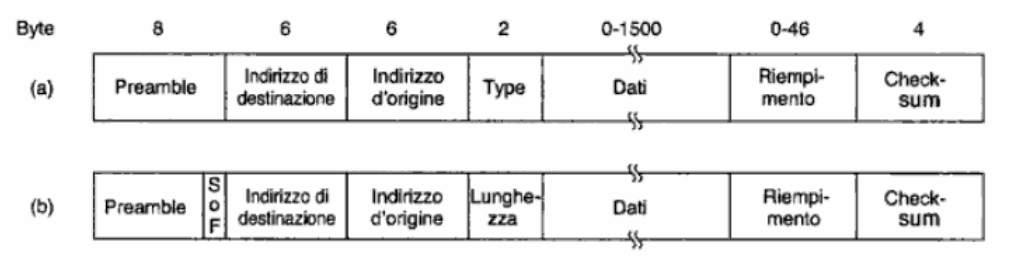


Figura 21: Struttura del frame DIX vs IEEE 802.3

3.5.4 Algoritmo di backoff esponenziale

Questo è un algoritmo che descrive come viene scelto (dinamicamente) il tempo di attesa casuale dopo una collisione. Dopo una collisione il tempo viene discretizzato in intervalli di lunghezza pari al massimo tempo di propagazione sul mezzo di trasmissione. Ogni stazione può decidere se ritrasmettere sull'intervallo 0 o 1. Nel caso di una nuova collisione tra 0 e 3, poi 0 e 7 in generale dopo i collisioni tra 0 e $2^i - 1$. Questo fa diminuire bruscamente la probabilità di collisione aumentando a ogni iterazione il tempo di attesa. Questo può essere ovviato se si tronca l'algoritmo a una decina di iterazioni cosicché la probabilità di collisione è trascurabile e il ritardo accettabile.

3.5.5 Fast Ethernet *(IEEE 802.3u)*

Nata per la necessità di velocizzare le reti è basata sullo standard 802.3 preesistente ottimizzata, mantenendo la retro-compatibilità. Approvato ufficialmente nel 1995 da IEEE, col nome 802.3u, l'idea alla base è semplice: mantenere tutto com'era prima aumentato il tempo di bit da 100 nsec a 10 nsec. Un problema non banale fu il capire il tipo di cavi supportati. I doppini cat3 erano ottimi in quanto tutti gli uffici occidentali ne disponevano ma non arrivano a velocità troppo elevate, non quanto i cat5. Si pensò allora di supportare più cavi, come fatto per Ethernet. Visti i vantaggi dei cavi 10Base-T fu scelta questa architettura.

Nome	Cavo	Lunghezza max. segmenti	Vantaggi
100Base-T4	Doppino intrecciato	100 m	UTP di categoria 3
100Base-TX	Doppino intrecciato	100 m	Full duplex a 100 Mbps (UTP di categoria 5)
100Base-FX	Fibra ottica	2.000 m	Full duplex a 100 Mbps; distanze elevate

Figura 22: I cavi Fast Ethernet

- **100Base-T4**: utilizza una velocità di segnale di 25 MHz con cavo di categoria 3. Per raggiungere la banda necessaria è richiesto l'uso di 4 doppini per raggiungere la banda necessaria. Uno dei 4 doppini trasmette sempre all'hub e no riceve e gli altri sono commutabili. Si abbandona la codifica Manchester. Grazie a 3 doppini dedicati alla trasmissione si invia un segnale ternario a ogni ciclo di clock avendo valori tra 0 e 2. In questo modo si inviano 4bit di informazione a ogni ciclo arrivando a 100Mbps con canale inverso a 33Mbps (schema **8B/6T**). Poco elegante ma funzionante con i cavi esistenti. Lunghezza fino a 100 m.
- **100Base-TX**: utilizza una velocità di segnale di 125 MHz con cavo di categoria 5. Per raggiungere la banda necessaria è richiesto l'uso di soli 2 doppini, uno verso e uno dall'hub. Anche qui è abbandonata la codifica binaria e si utilizza uno schema chiamato 4B/5B: ogni 5 cicli di clock si hanno 32 combinazioni, le prime sedici trasmettono i 4 gruppi di bit e i rimanenti per funzioni di controllo. Lunghezza fino a 100 m.
- **100Base-FX**: utilizza fibre ottiche multimodali raggiungendo in full duplex una velocità di 100 Mbps. Lunghezza fino a 2 Km.

3.5.6 Gigabit Ethernet *(802.3z)*

L'idea principale è di rendere Ethernet 10 volte più veloce mantenendo la retro-compatibilità. Tutte le configurazioni Gigabit sono punto-punto, supportando 2 modalità: full duplex se collegati a switch (che offre anche un buffer di memorizzazione) e half duplex se collegati ad hub. In full duplex non esistono più le collisioni così si abbandona l'uso del CSMA/CD, in hal duplex invece viene utilizzato. Siccome per raggiungere velocità elevate si doveva però ridurre la distanza notevolmente (distanza massima 25 m!!) furono introdotte alcune funzionalità chiamate **carrier extension** e **frame bursting**.

Carrier extension

Essenzialmente dice all'hardware di aggiungere byte al pacchetto fino a raggiungere i 512 byte. Questo aumenta l'efficienza di circa il 9%. Siccome è tutto sull'hardware non c'è bisogno di modifica al software esistente.

Frame bursting

Concatena più frame in in una singola trasmissione riempendoli per raggiungere i 512 byte se necessario. Se i dati sono minori di 512 byte vengono

aggiunti per riempire direttamente dall'hw. Se molti frame sono in attesa questo schema è più efficiente del precedente.

Queste 2 funzionalità estendono la rete per un raggio di circa 200 m. I cavi supportati da questo standard sono riportati di seguito:

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

Figura 23: I cavi Gigabit Ethernet

3.6 LAN Wireless

3.6.1 Pila di protocolli 802.11

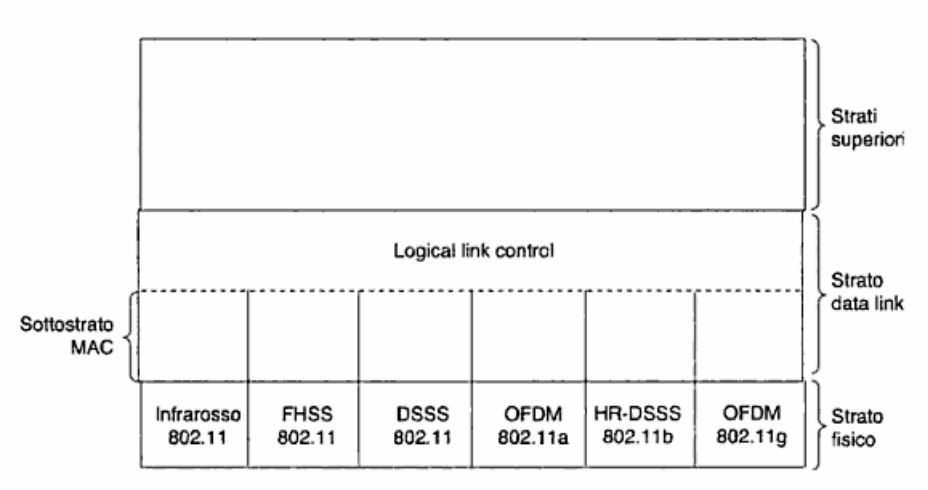


Figura 24: Parte della pila di protocolli 802.11

Qui sopra si può vedere parte della pila dei protocolli 802.11: lo strato fisico è più o meno corrispondente a quello dello strato fisico OSI, lo strato Data link è divo in sottostrato MAC che stabilisce il metodo di allocazione del anale e **LLC** (*Logical Link Control*) che ha il compito di rendere indistinguibili le differenze tra l varianti di 802.

3.6.2 Strato fisico di 802.11

Esistono 5 tecniche diverse per l'invio di un frame MAC da una stazione all'altra. La variante a infrarossi utilizza una trasmissione diffusa (non in linea retta) e supporta le velocità 1-2 Mbps. Ad 1 Mbps si usa la codifica **Gray**. **FHSS** (*Frequency Hopping Spread Spectrum*) utilizza 79 canali da 1MHz Si usa un generatore di numeri pseudo casuali per produrre la sequenza di frequenze che si susseguono. Tutte le stazioni salteranno sulle stesse frequenze se usano lo stesso seme generatore e restano sincronizzate. Il **tempo di rotazione** (tempo per ogni frequenza) è arbitrario ma inferiore a 400 msec. Grazie alla sua casualità FHSS è abbastanza sicuro e buono per allocare lo spettro. Il terzo metodo di modulazione **DSSS** (*Direct Sequence Spread Spectrum*) è limitato a 1 o 2 Mbps. Simile a CDMA ogni bit è trasmesso come 11 chip usando la cosiddetta **sequenza Barker**. Un'altro metodo adottato dalla prima LAN wireless ad alta velocità, 802.11a, si chiama **OFDM** (*Orthogonal Frequency Division Multiplexing*) che distribuiva fino a 54Mbps

nella banda dei 5 GHz. Utilizza molte frequenza 48 per i dati e 4 per la sincronizzazione. Per raggiungere alte velocità vengono utilizzate complicate tecniche di modulazione di fase. Altra tecnica a diffusione di spettro come la precedente è l' **HR-DSSS** (*High Rate DSSS*)raggiunge velocità di 11 Mbs utilizzando 11 milioni di chip al secondo. Chiamato **802.11b**, alle velocità più basse viene utilizzata una modulazione di fase per mantenere la compatibilità con DSSS. A quelle più alte viene invece utilizzata una codifica con codici Walsh/Hadamard. L'evoluzione fu **802.11g** che utilizza OFDM ma opera in una banda più ristretta.

3.6.3 Il protocollo del sottostrato MAC di 802.11

Per gestire il problema della trasmissione senza collisioni ed evitare il problema della stazione esposte e nascosta si sono ideate 2 modalità operative. La prima chiamata **DCF** (*Distribuited Coordination Function*) che non utilizza alcun controllo centrale e 'altra chiamata **PCF** (*Point Coordination Function*) che usa la stazione centrale per controllare la cella. Adottando DCF si utilizza un protocollo chiamato **CSMA/CA** (*CSMA/Collision Avoidance*) vengono controllati sia il canale fisico che quello virtuale. Supporta 2 modalità operative: uno si basa sull'attesa del canale libero per trasmettere e in caso di collisione si usa l'exponential backoff per attendere. La seconda modalità si basa su MACAW e viene controllato il canale virtuale. Il protocollo entra in azione in una situazione come in figura quando A vuole trasmettere a B. A invia a B un frame RTS per richiedere di trasmettere, se B glielo

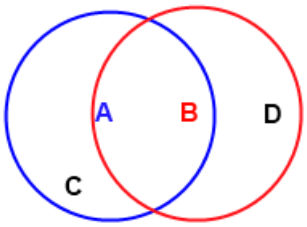


Figura 25: Situazione ad hoc per CSMA/CA (1).

consente invia di ritorno un frame CTS e successivamente A invia i dati e fa partire il suo timer. Intanto C, nel raggio di azione di A, vede il frame RTS e quindi stima il tempo della comunicazione e si alloca un **NAV** (Network Allocation Vector), una sorta di canale virtuale. D invece riceve il CTS di B e analogamente a C alloca un NAV stimato leggermente più corto. Vedi figura. Per ovviare ai problemi dei canali rumorosi 802.11 permette la divisione dei

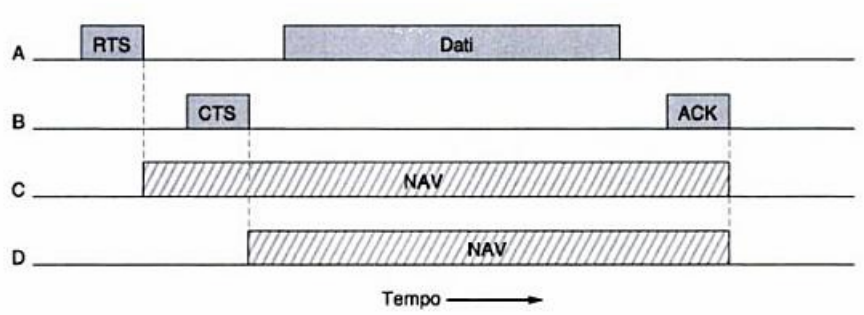


Figura 26: Situazione ad hoc per CSMA/CA (2).

frame in frammenti più piccoli aggiungendo il loro checksum e il numero di sequenza con ACK associato e utilizzando un protocollo stop-and-wait. Dopo lo scambio RTS-CTS si possono inviare più frammenti consecutivi (**burst di frammenti**). Quando si utilizza la tecnica PCF c'è bisogno di un controllo centralizzato per vedere se le stazioni devono trasmettere e questo viene fatto con un **frame di segnalazione** che si occupa di gestire vari problemi. Inoltre la stazione base si occupa di memorizzare i frame indirizzati alle stazioni che si sono disattivate. PCF e DCF possono coesistere grazie a un ingegnosa suddivisione di intervalli dopo l'invio di un frame come si vede in figura.

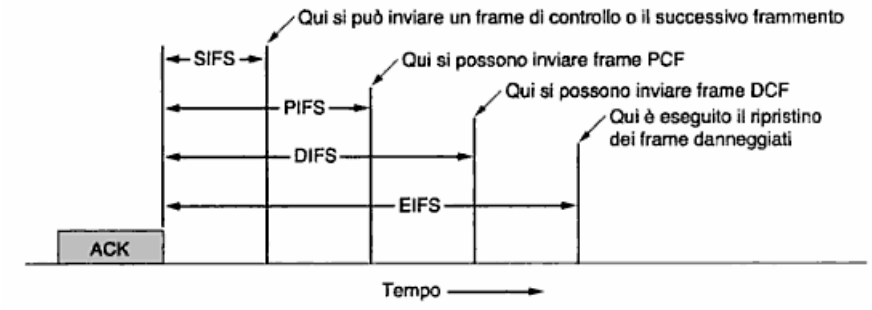


Figura 27: Intervalli tra frae in 802.11.

3.7 Servizi

Ogni LAN Wireless deve servire 9 servizi fondamentali:

1. **Associazione**: Utilizzato per effettuare la connessione alla stazione base. Al suo arrivo la stazione annuncia la propria identità e le funzionalità. Può o meno accettare stazioni mobili in caso affermativo c'è bisogno di autenticazione.

2. **Separazione**: gestisce la disconnessione (separazione) tra stazione mobile e stazione base.
3. **Riassociazione**: servizio che serve per sganciarsi da una stazione e riagganciarsi ad un altra.
4. **Distribuzione**: servizio che descrive il tipo di instradamento dei frame verso la stazione base.
5. **Integrazione**: servizio che si occupa della traduzione dal formato 802.11 a quello richiesto.
6. **Autenticazione**: servizio che richiede l'autenticazione alla stazione mobile, la quale deve cifrare il frame di challenge e reinviarlo alla base.
7. **Invalidamento**: quando una stazione vuole lasciare la rete deve essere invalidata.
8. **Riservatezza**: servizio che cifra le informazioni trasmesse..
9. **Trasferimento dati**: servizio che permette la trasmissione dei dati.

3.8 Bluetooth

Standard nato dal consorzio SIG attorno al 1999.

3.8.1 Architettura Bluetooth

L'unita base di un sistema Bluetooth è il **piconet**, composto da un nodo master e diversi nodi slave (≤ 7) situati in un ragio di circa 10m. Più piconet possono connettersi tra loro formando uno **scatternet**. La rete inoltre può contenere fino a 255 nodi sospesi, ovvero in bassa alimentazione. In questo stato un nodo può solamente rispondere a una richiesta di attivazione o a un segnale del nodo master. Il cuore della piconet è costituito da un sistema TDM centralizzato: il nodo master controlla il clock e decide chi può comunicare a ogni intervallo. Non sono ammesse comunicazioni dirette tra nodi slave.

3.8.2 Applicazioni Bluetooth

La secifica Bluetooth nomina 13 applicazioni specifiche, detti **profili**, da supportare con i rispettivi protocolli (Vedi figura).

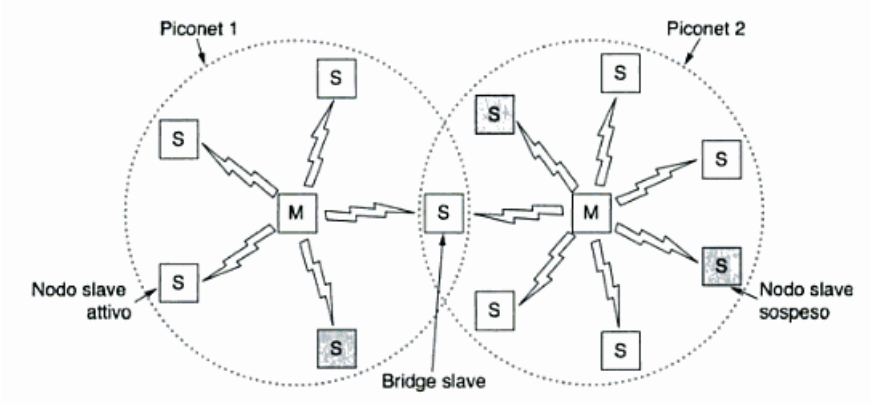


Figura 28: Due piconet: Scatternet.

Nome	Descrizione
Accesso generico	Procedure per la gestione del collegamento
Scoperta del servizio	Protocollo per scoprire i servizi offerti
Porta seriale	Sostituzione del cavo seriale
Scambio di oggetti generico	Definisce la relazione client/server per lo spostamento degli oggetti
Accesso LAN	Protocollo tra un computer portatile e una LAN fissa
Accesso dial-up	Premette a un computer portatile di chiamare attraverso un telefono mobile
Fax	Permette a un apparecchio fax mobile di comunicare con un telefono mobile
Telefonia cordless	Collega un telefono alla sua base
Intercomunicanti	Ricetrasmittenti digitali
Auricolare wireless	Comunicazione vocale senza mani
Invio oggetto	Permette di scambiare semplici oggetti
Trasferimento file	Gestisce il trasferimento di file
Sincronizzazione	Permette a un PDA di sincronizzarsi con un altro computer

Figura 29: i profili Bluetooth.

3.8.3 Lo strato radio di Bluetooth

Questo strato si occupa di spostare i bit dal nodo master allo slave e viceversa. La banda è divisa in 79 canali di 1MHz. La modulazione è FSK (Frequency Shift Keying) con 1 bit per Hz quindi una velocità di circa 1Mbps, ma gran parte dello spettro è utile per il checksum. Per assegnare i canali si usa una tecnica a spettro distribuito a frequenza variabili con 1600 cambi al secondo.

3.8.4 Lo strato baseband di Bluetooth

Trasforma il flusso di bit grezzi in frame. Il nodo master definisce una serie di intervalli temporali di 625 microsec e lui trasmette negli intervalli pari e gli slave in quelli dispari. Un multiplexing a divisione di tempo. I frame possono occupare 1,3 o 5 intervalli. Ogni frame è trasmesso attraverso un canale logico chiamato **link** stabilito tra un nodo master e uno slave. Il primo collegamento, chiamato **ACL** (Asynchronous ConnectionLess), è utilizzato per i dati a commutazione di pacchetto. I dati sono trasmessi in modo **best effort**, cioè senza alcuna garanzia di consegna. L'altro collegamento si chiama **SCO** (Synchronous Connection Oriented)ed è utilizzato per i dati in tempo reale. I frame inviati attraverso questi canali non vengono mai ritrasmessi, ma si utilizza un meccanismo di correzione degli errori. A differenza del precedente che permetteva un solo collegamento slave-master lo SCO ne permette 3.

3.8.5 La struttura del frame Bluetooth

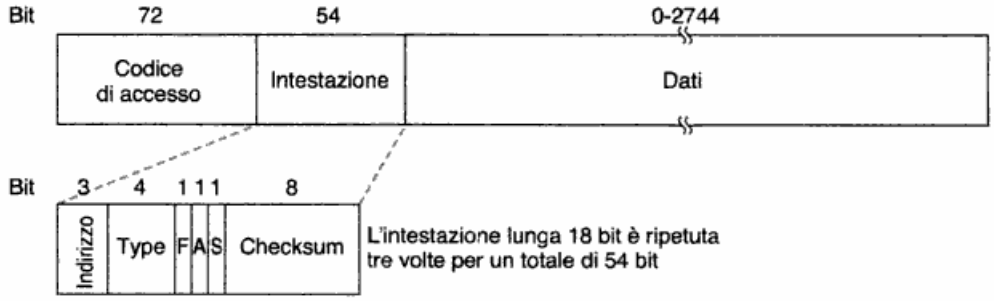


Figura 30: Frame Bluetooth

Inizia con un codice d'accesso che solitamente identifica il nodo master. Segue un intestazione di 54 bit contenente i classici campi del sottostrato MAC. Poi il campo dati che può arrivare a 2744 bit. Il campo *indirizzo* identifica il destinatario. Il campo *type* identifica il tipo di frame

(ACL—SCO—interrogazione—nullo). F sta per *Flow* ed è un bit attivato da uno slave quando ha il buffer pieno. A sta per *Acknowledgement* e aggiunge semplicemente un ACK al frame. S invece sta per *Sequence*, utilizzato per numerare i frame. Seguono gli 8bit di checksum.

3.9 Commutazione nello strato data link

[Da integrare]

3.10 Tabella riassuntiva

Metodo	Descrizione
FDM	Dedica una banda di frequenza a ogni stazione
WDM	Schema FDM dinamico per la fibra
TDM	Dedica un intervallo temporale a ogni stazione
ALOHA puro	Trasmissione non sincronizzata che può iniziare in qualunque istante
ALOHA slotted	Trasmissione casuale in intervalli di tempo ben definiti
CSMA 1-persistente	Capacità standard di rilevamento della portante ad accesso multiplo
CSMA non persistente	Ritardo casuale quando il canale è occupato
CSMA P-persistente	CSMA, ma con una probabilità p di persistenza
CSMA/CD	CSMA, con interruzione in caso di collisione
Mappa di bit	Pianificazione a turno mediante una mappa di bit
Conteggio binario	Seleziona la stazione pronta associata al numero più alto
Tree walk	Contesa ridotta mediante attivazione selettiva
MACA, MACAW	Protocolli LAN wireless
Ethernet	CSMA/CD con backoff esponenziale binario
FHSS	Spettro distribuito a frequenza variabile
DSSS	Spettro distribuito a sequenza diretta
CSMA/CA	Capacità di rilevamento della portante ad accesso multiplo con annullamento delle collisioni

Figura 31: Sommario cap.3

4 Lo strato Network

4.1 Servizio senza connessione e orientato alla connessione

In un servizio senza connessioni i pacchetti, detti **datagrammi** sono inviati singolarmente senza alcuna configurazione anticipata e ognuno segue la propria strada. Questi tipo di sottoreti sono dette **sottoreti a datagrammi**. Se il servizio è orientato alla connessione prima di inviare i pacchetti bisogna che ci sia un collegamento tra i 2 router. Questa connessione chiamata **CV** (*Circuito virtuale*) e la sottorette è detta **sottorete a circuito virtuale**. Queste 2 sottoreti hanno vantaggi e svantaggi come indicati nella tebella seguente:

Problema	Sottorete a datagrammi	Sottorete a circuito virtuale
Impostazione circuito	Non è necessaria	È richiesta
Indirizzamento	Ogni pacchetto contiene l'indirizzo completo di destinazione e quello di origine	Ogni pacchetto contiene un numero CV corto
Informazioni sullo stato	I router non conservano informazioni sullo stato delle connessioni	Ogni CV richiede spazio nella tabella del router per la connessione
Routing	Ogni pacchetto è instradato indipendentemente	Il percorso è scelto durante l'impostazione del CV. Tutti i pacchetti seguono lo stesso percorso
Effetto dei guasti nei router	Nessuno, tranne per i pacchetti perduti durante il guasto	Tutti i CV che passano attraverso il router guasto sono terminati
Qualità del servizio	Difficile	Facile se è possibile assegnare in anticipo abbastanza risorse a ogni CV
Controllo della congestione	Difficile	Facile se è possibile assegnare in anticipo abbastanza risorse a ogni CV

Figura 32: Confronto tra le 2 sottoreti.

4.2 Algoritmi di Routing

Questi algoritmi rappresentano la parte software dello strato network che si preoccupano di scegliere lungo quale strada vanno instradati i pacchetti. Questascelta viene fatta una sola volta per le sottoreti che utilizzano un

circuito virtuale mentre viene fatta per ogni datagramma in una sottorete a datagrammi. Gli algoritmi di routing si posso raggruppare in 2 categorie principali: **algoritmi non adattivi** che non basano le loro decisione su stime del traffico o sulla topologia della rete (**routing statico**), mentre gli **algoritmi adattivi** cambiano le loro decisioni in base alle modifiche topologiche della rete e a volte anche in base al traffico. Il problema del routing è fondamentale nella gestione di reti. Bisogna decidere che strada devono fare i pacchetti per arrivare a destinazione. Un metodo semplice è contare il numero di **hops** (stazioni passate) e scegliere quella che ne ha di meno. Un altro metodo semplice, se la rete e statica, è utilizzare un algoritmo per il cammino minimo in un grafo come Dijkstra che si basa sul **principio di ottimalità** che afferma che se il router J si trova sul percorso ottimale tra I e K allora anche il percorso ottimale da J a K segue la stessa rotta.

4.2.1 Flooding

Algoritmo statico in cui ogni pacchetto che arriva a una stazione, viene rimandato in tutte le direzioni esclusa quella da cui è arrivato. Un metodo potente, ma a rischio di congestione. Si può migliorare utilizzando un metodo di **hop counting**, che semplicemente conta le stazioni percorse e la trasmissione si ferma fino al raggiungimento di un hopmax. Un altro modo è tener traccia dei pacchetti già arrivati (nei pacchetti c'è un numero di serie) e non ritrasmetterli più, richiedendo però molta memoria (a ogni router serve un buffer per ricordare i numeri di serie dei pacchetti e la loro origine), un compromesso è tenerne traccia fino a un certo punto. Questo metodo funziona bene in comunicazioni P2P, broadcast e multicast. Robusta rispetto alle modifiche della rete. Una variante di questa tecnica è il **flooding selettivo**, ogni router non trasmette in tutte le direzioni ma solo in quelle che approssimativamente vanno nella direzione giusta. Nella maggior parte dei casi comunque questo algoritmo non è molto utilizzato.

4.2.2 Distance vector routing

Questo a differenza dei precedenti è un algoritmo dinamico poiché tiene conto del carico instantaneo della rete. Ogni router possiede una mappa di tutte le distanze e le connessioni con ogni altro router. Per trovare la via migliore ogni router chiede la mappa (vettore) ai router vicini e grazie a quest'ultime e al tempo di risposta costruisce la propria. Ogni voce in questa tabella contiene 2 parti: la linea di trasmissione preferita e la stima del tempo o della distanza associata a quella destinazione. Le metriche utilizzate sono il numero di hop, la lunghezza nella coda oppure il ritardo che si può calcolare

grazie a speciali pacchetti ECHO. Ogni T msec ogni router invia ai propri vicini i ritardi segnati nella propria tabella così da poter tenere la rete sempre aggiornata. Il problema di questa tecnica si ha quando un nodo sparisce o diventa lentissimo(Vedi sezione successiva). Per ovviare al problema si utilizza un algoritmo un po diverso: il Linkstate routing. Ogni router conserva una tabella di routing formata da 2 colonne, una che indica la linea di trasmissione preferita per quella destinazione e il ritardo espresso in hop o in msec di ritardo. Il ritardo viene calcolato tramite speciali pacchetti ECHO.

Problema del conto all'infinito

In termini teorici l'algoritmo appena descritto converge alla soluzione, non è garantito però che avvenga in tempi brevi. C'è una differenza sostanziale nella propagazione di buone e cattive notizie. Partiamo vedendo come si propagano le buone notizie: se in una rete composta dai nodi B, C, D, E che distano rispettivamente 1 dal successivo, se si aggiunge A prima di B lo scambio di informazioni tra i router avviene in un numero di passaggi pari al loro numero (vedi figura). Una cattiva notizia invece, A si spegne brutalmente, ha effetti disastrosi. B una volta che si accorge che non arrivano i pacchetti ad A vede che passando per C ci vogliono 3 passi e quindi aggiorna la sua tabella. Poi C fa lo stesso con D e così via. Il problema sta nel fatto che le strade migliori passano per i router che invece non hanno alcun collegamento effettivo con A. (vedi figura)

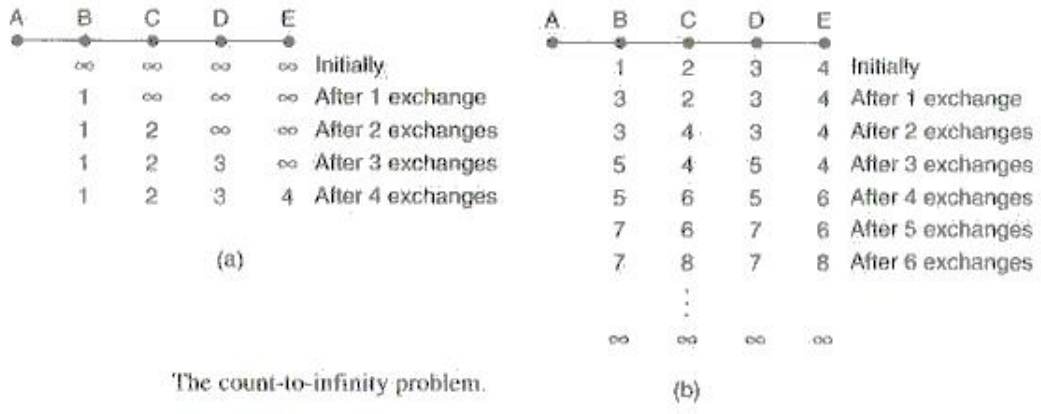


Figura 33: Il problema del conteggio a infinito

4.2.3 Linkstate routing

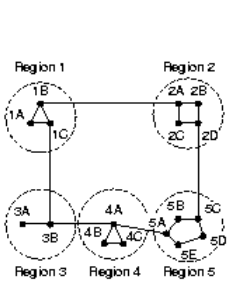
Questo algoritmo è basato sullo stato dei collegamenti e può essere riassunto in 5 punti:

1. Scoprire i propri vicini e il loro indirizzo di rete
2. Misurare il ritardo dai vicini
3. Costruire un pacchetto con le informazioni raccolte
4. Inviare pacchetti agli altri router
5. Elaborare il percorso più breve dai router.

Inizialmente, come l'algoritmo precedente, avviene il passaggio di pacchetti HELLO che indica al vicino di inviare il proprio indirizzo di rete dicendogli anche la propria identità. Successivamente ogni router deve capire il ritardo da ogni vicino, grazie a un pacchetto ECHO che indica al ricevente di rispondere immediatamente per capire il ritardo dalla sorgente. Poi vengono costruiti i pacchetti con le informazioni sullo stato dei collegamenti e inviati in broadcast utilizzando il flooding. Questa costruzione è molto semplice: il pacchetto inizia con la propria identità seguita da un numero di sequenza, dall'età e da una lista di vicini con affiancato il ritardo rilevato. Se il numero del pacchetto ricevuto rispetto al precedente è più recente viene memorizzato e inoltrato alle altre stazione, se è meno recente viene ritrasmesso alla sorgente, se invece è lo stesso viene scartato. Per costruire infine il percorso più breve tra router viene utilizzato Dijkstra.

4.2.4 Hierarchical routing (Routing Gerarchico)

A volte la rete è talmente estesa che avere informazioni riguardo a tutti gli altri router diventa impossibile. In questi casi la rete viene divisa in **regioni**: ogni router conosce solo i router della propria regione. Quando la rete è molto grande le regioni vengono raggruppate in claster, poi in zone poi in gruppi ecc... . Così facendo Quando si connettono diverse sottoreti ogni regione vede un altra come un semplice nodo. La figura di qui sotto mostra la riduzione della tabella da routing normale a gerarchico. Questo risparmio di spazio però ha un lato negativo: non sempre il percorso è il più breve, sfortunatamente spesso non lo è.



Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Figura 34: Routing gerarchico

4.3 Routing broadcast

Alcune applicazioni a volte hanno la necessità di dialogare con tutti i router della rete, si ha la necessità così di trasmettere in broadcast. Un modo semplice e inviare un pacchetto a tutti. Un metodo molto semplice quanto oneroso perché cobsuma tantissima banda. Un altro metodo abbastanza semplice ma inefficace è il flooding. Ha lo stesso problema del precedente anche se in maniera minore. Un approccio buono è il **multidestination routing** che assegna a ogni pacchetto una lista di destinazioni desiderate. Poi c'è l'algoritmo di **spanning tree**, che crea un albero al di sopra del grafo (ovvero la rete), in modo da non avere cicli. Ogni router dovrà conoscere questo albero (è questo il problema!) così sa su che linee inviare i vari pacchetti.Infine c'è l'algoritmo di **reverse path forwarding**. Funziona come il flooding solo che vengono considerati solo i pacchetti che arrivano dal cammino migliore (quello che arriva dalla sorgente). Se arriva da una linea non considerata migliore, il pacchetto è visto come duplicato, quindi scartato.

4.4 Algoritmi per il controllo della congestione

Quando troppi pacchetti sono presenti in una sottorete si crea **congestione**. Una congestione può essere causata da molti fattori: improvvisamente molti flussi di pacchetti vengono incanalati in poche linee di trasferimento, la memoria può non essere sufficiente, alcune stazioni possono non avere microprocessori veloci, o le linee possono avere banda stretta. Una differenza fondamentale è il controllo di flusso dal controllo della congestione. Il primo si limita al controllo del traffico P2P. Il suo compito è evitare che un trasmettitore sia troppo veloce rispetto al ricevitore. Il secondo invece deve garantire che la sottorete sia in grado di trasportare il traffico immesso.

4.4.1 Choke packet

Il router invia all'host sorgente un **choke packet** dandogli la destinazione trovata nel pacchetto. Il pacchetto viene etichettato in modo che non venga 'richokkato' e viene inoltrato come sempre. Quando riceve il choke la sorgente deve ridurre il traffico verso la destinazione specificata dell'X % indicato. La sorgente ignorerà le richieste di choke provenienti dai pacchetti inviati nel frattempo. Per evitare che la congestioni si ripeti, il ripristino della velocità viene fatto con incrementi più piccoli.

4.4.2 Choke packet hop-by-hop

Ad alte velocità e su lunghe distanze l'approccio precedente ha problemi di ritardo. Qui il choke ha effetto immediato poiché non appena arriva il choke al router, lui stesso riduce la velocità nel mandare la segnalazione alla sorgente.

4.4.3 Load shedding

Questo metodo è davvero banale: quando un router è troppo carico scarta alcuni pacchetti, ma lo fa con un po' di criterio. Se per esempio si sta trasferendo dei semplici file i pacchetti più vecchi hanno più importanza di quelli nuovi quindi sono quest'ultimi a essere scartati. Nel caso di trasmissioni multimediale invece è il contrario. Questi 2 approcci sono chiamati rispettivamente **wine** a **milk**. Per migliorare l'algoritmo di decisione i pacchetti vengono contrassegnati da un livello di priorità.

4.4.4 RED (*Random Early Detection*)

L'idea alla base è simile a quella del Load shadding con la sola differenza che al posto di aspettare che la congestioni fermi tutto lo scarto dei pacchetti viene fato prima che il buffer sia pieno. Si avvia lo scarto anticipato quando le code delle linee superano una certa soglia limite prestabilita. La sorgente non viene avvisato della cancellazione del pacchetto per evitare traffico inutile. Semplicemente la sorgente non vedendo l'ACK prende provvedimenti.

4.4.5 Controllo del Jitter

La variazione di tempo con cui arrivano i pacchetti è detta **jitter**. In trasmissioni audio/video un jitter elevato causa una qualità variabile del media. Quando un pacchetto arriva al router viene controllato il suo anticipo/ritardo e viene inserita questa info nel pacchetto stesso. Se il pacchetto è in anticipo

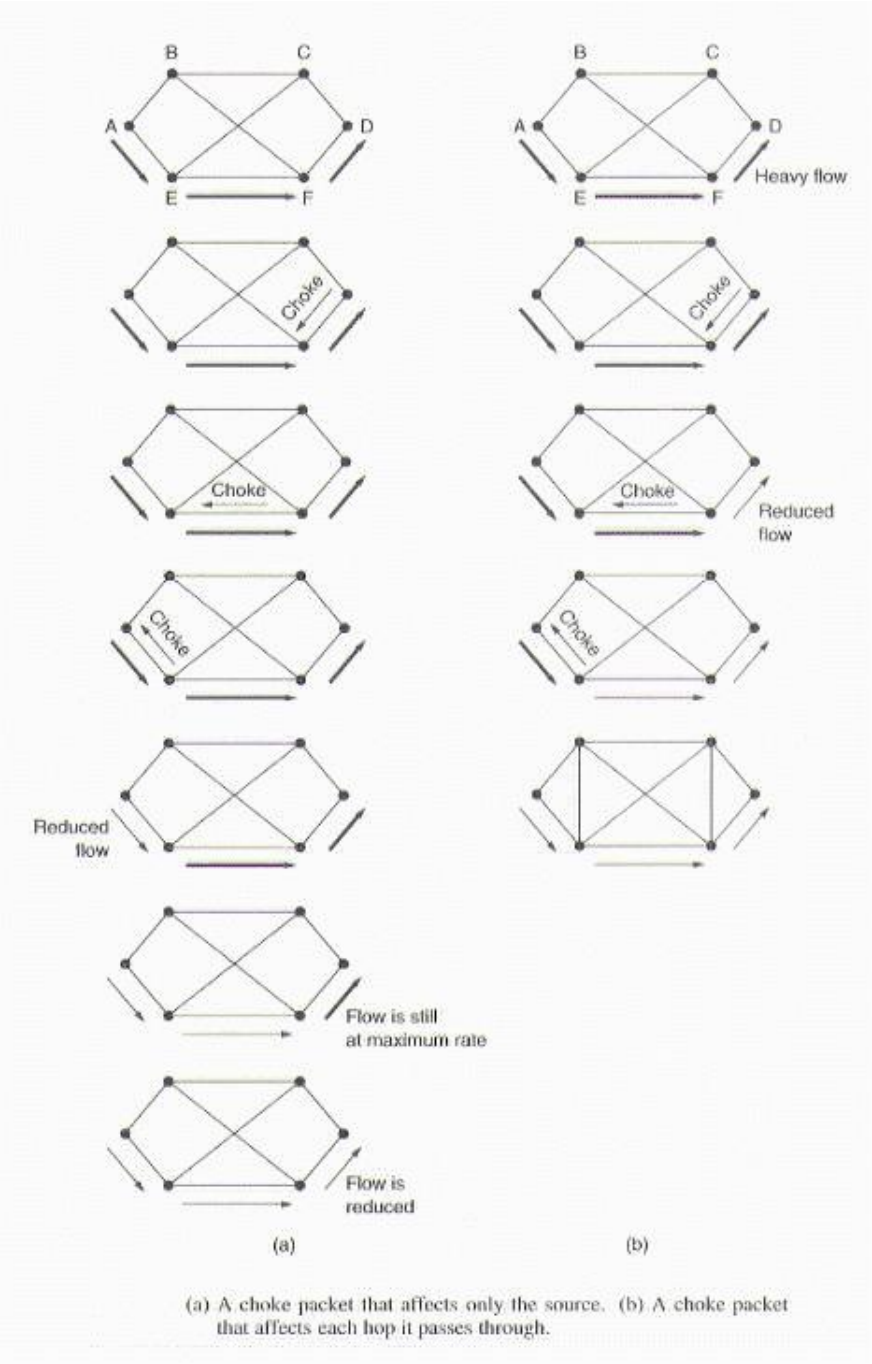


Figura 35: (a) Choke packet; (b) Hop-by-Hop

viene trattenuto per il tempo necessario, se in ritardo viene ritrasmesso il prima possibile.

4.4.6 Qualità del servizio

Un flusso di pacchetti di retto da una sorgente a una destinazione è chiamato semplicemente **flusso**. L'esigenza di ogni flusso possono essere caratterizzate da 4 parametri primari: affidabilità, ritardo, jitter(cambiamento delle caratteristiche del segnale) e banda. Questi parametri assieme determinano la **QoS** *(Quality of Service)*.

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Figura 36: QoS

4.4.7 Ottenere una buona qualità di servizio

Sovradimesionamento

Una semplicissima soluzione è fornire al router tanta capacità e spazio di buffer. Il problema è ovviamente il fatto che è costoso.

Utilizzo del buffer

I flussi si possono memorizzare in buffer senza avere conseguenze sull'affidabilità o sulla banda. Eliminando così il jitter causando però ritardi. Per trasmissione audio e video questa tecnica è molto utile. Non risolve la congestione.

Traffic shaping

Questa tecnica si basa sull'idea di rendere uniforme il traffico per evitare congestioni, regolando la velocità media della trasmissione. In questo modo il cliente chiede all'operatore se è in grado di gestire un certo modello di

trasmissione (**service level agreement**). Se può allora viene monitorato per vedere se mantiene le sue promesse *(traffic policing)*. Questa supervisione è molto più semplice nelle sottoreti a circuito virtuale rispetto a quella a datagrammi.

Leaky bucket

L'idea che sta alla base è questa: si pensi a un secchio con dell'acqua avente un piccolo foro. Per quanta acqua ci sia dentro la velocità di fuori uscita dal foro è costante. Ecco, ogni host si interaccia alla rete con un **leaky bucket**, ovvero un buffer sottoforma di coda. Quando arriva un pacchetto a coda piena viene subito scartato. L'host trasmette un pacchetto ogni ciclo di clock. Questi comportamenti possono essere gestiti sia dal OS che a livello HW. Questo algoritmo viene applicato così come appena descritto quando i pacchetto hanno dimensione costante. In caso contrario viene gestito un contatore inizializzato a n a ogni ciclo di clock. Se si trasmette un pacchetto di dimensioni inferiore si ha l'opportunità di trasmetterne altri restando però all'interno della dimensione n. Il **leacky bucket a conteggio di byte** funziona analogamente a quello appena descritto.

Token bucket

E' la versione dinamica del precedente. Qui il leaky bucket contiene dei token generati da un clock. Perché un pacchetto possa essere trasmesso deve prendere e distruggere un token. Se i token finiscono i pacchetti devono attendere la nuova generazione. Se un host resta molto inattivo il numero di token posseduto non è infinito ma limitato a un certo n. Una variante di questo algoritmo consiste in token che permettono l'invio non di un pacchetto ma di un certo quantitativo di byte.

4.5 Lo strato network in internet

La rete internet può essere vista come un insieme di sottoreti o di **Autonomous System** *(AS)* interconnessi. Non esiste una vera e propria sruttura ma solo dorsali principali formate da linee a banda larga a cui le reti regionali si collegano (Vedi Immagine). Internet poi è retto dal protocollo IP pensato da subito per la comunicazioni tra reti. La comunicazione in Internet funziona così: lo strato trasporto prende i flussi di dati e li divide in datagrammi, ognuno di questi è trasmesso attraverso Internet e può essere frammentato in unità più piccole.

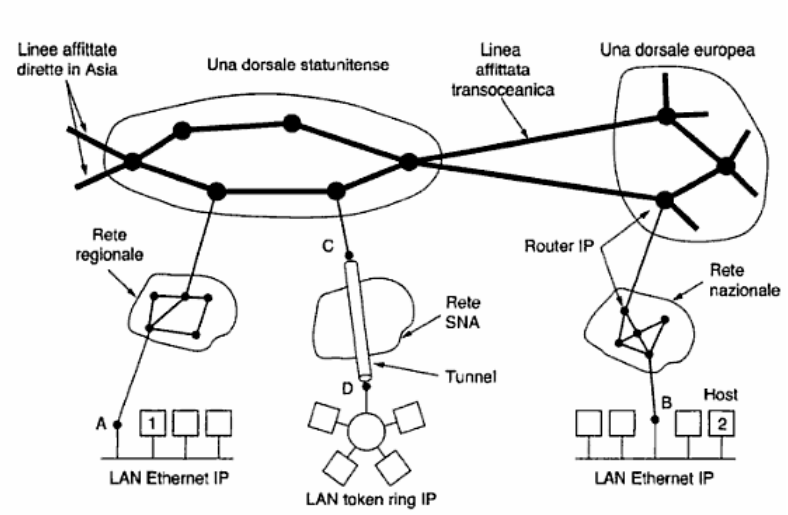


Figura 37: Internet è un insieme di reti interconnesse.

4.5.1 Protocollo IP

Prima di tutto esaminiamo il datagramma IP.

- **Version**: indica la versione del protocollo utilizzato dal datagramma (IPv4/IPv6).
- **IHL**: indica la lunghezza dell'intestazione espressa in parole da 32 bit, quindi può essere al massimo 60 byte.
- **Type of Service**: indica il tipo di servizio utilizzato (QoS).
- **Total lenght**: tiene conto di tutto il contenuto del datagramma (255 hops).

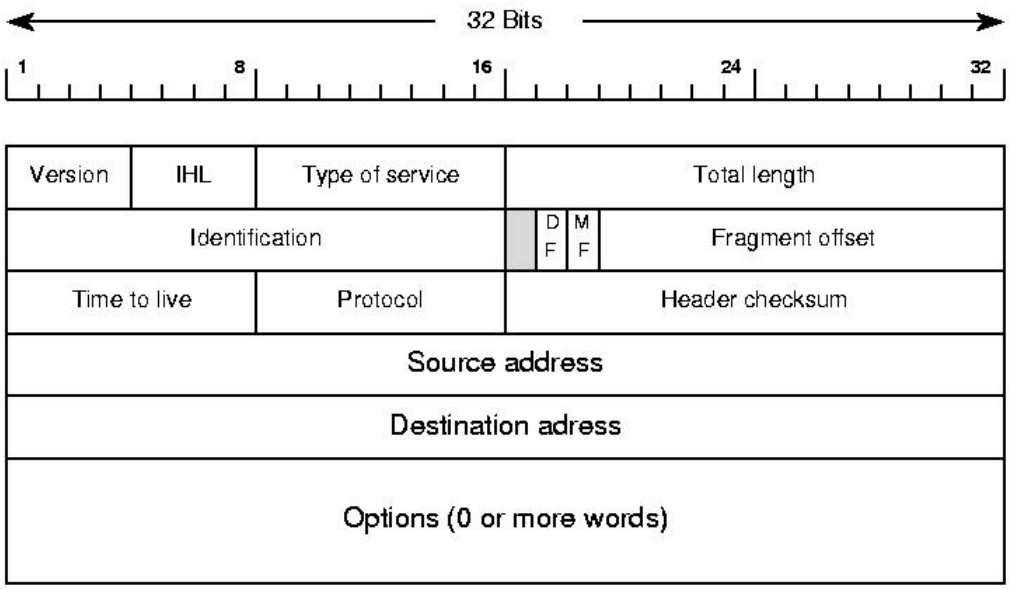


Figura 38: Datagramma IP

- **Identification**: identifica il datagramma di appartenenza del frammento.
- **DF**: *(Don't Fragment)* indica di non frammentare il datagramma.
- **MF**: *(More Fragments)* indica l'esistenza di altri frammenti del datagramma.
- **Time to live**: limita la vita del pacchetto.
- **Protocol**: indica quale processo di trasporto attende quei dati.
- **Header checksum**: verifica l'header. E' creato con somme in complemento a 2.
- **Source Address**: indirizzo della sorgente (32 bit - Pochi!!).
- **Destination Address**: indirizzo del destinatario (32 bit - Pochi!!).
- **Options**: campi opzionali.

Le opzioni sono di lunghezza variabile, ognuna inizia con 1 byte che la identifica e alcune sono seguite da un campo *option lenght*. Il campo option è riempito con multipli di 4 byte. Le opzioni principali sono elencate in figura.

Significato	Descrizione
Sicurezza	Specifica il livello di segretezza del datagramma
Instradamento strettamente definito dall'origine	Definisce il percorso completo da seguire
Instradamento lascamente definito dall'origine	Elenca i router che non devono essere mancati
Registra il percorso	Fa sì che ogni router aggiunga il proprio indirizzo IP
Contrassegno temporale	Fa sì che ogni router aggiunga indirizzo e ora

Figura 39: Alcune opzioni di IP.

4.5.2 Indirizzi IP

Ogni host e router possiede un indirizzo IP che codifica il suo indirizzo di rete e il suo numero host. Un indirizzo IP per la precisione si riferisce a una scheda di rete non ad un host. In una rete internet non possono esistere 2 indirizzi uguali su macchine diverse. Tutti gli indirizzi sono lunghi 32bit e sono utilizzati nei campi source e destination address del pacchetto IP. Gli indirizzi di rete sono gestiti da un azienda no profit **ICANN**. Esistono varie classi di indirizzi IP (**indirizzamento per classi**):

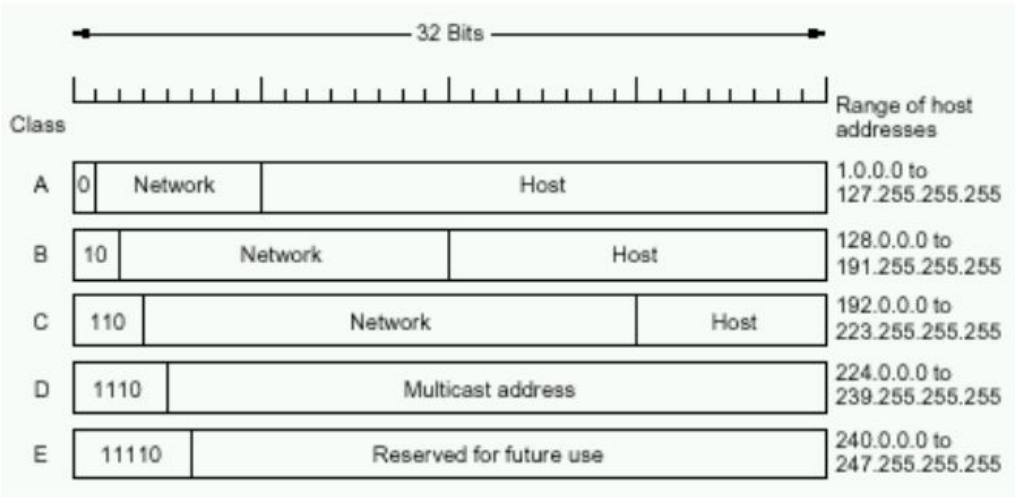


Figura 40: Classi di indirizzo IP

- **A**: 0 + 7 bit di rete + 3 byte di host
- **B**: 10 + 14 bit di rete + 2 byte host
- **C**: 110 + 22 bit di rete 1 byte host
- **D**: 1110 + 28 bit indirizzamento multicast
- **E**: 1111 + ... Riservato per usi futuri

Esistono degli indirizzi speciali:

- Tutti 0: il nostro indirizzo
- Tutti 1: indirizzo di broadcast
- 0 nella parte rete + host: identifica un host della nostra rete

4.5.3 Sottoreti

Quando una rete comincia a diventare troppo vasta l'assegnazione degli indirizzi IP tra varie sottoreti diventa complicato. Si pensi a un infrastruttura universitaria avente un unico inndirizzo di rete da condividere tra molte facoltà. Creerebbe non pochi problemi un assegnazione fatta senza criterio, per questo si è pensati di tenere alcuni bit degli indirizzi host per identificare la sottorete. Per fare questo serve una **subnet mask** che indichi il punto di demarcazione tra l'indirizzo della rete e quello dell'host. Questa suddivisione in sottoreti non è visible all'esterno.

4.5.4 CIDR *(Classless InterDomain Routing)*

L'idea alla base è semplice: gli indirizzi rimanenti vengono assegnati in base alla necessità senza tenere conto delle classi. Questa tecnica però crea problemi per quanto riguarda l'inoltro. Ora non c'è più una categorizzazione ben definita delle classi, ma ogni router dovrà tener traccia per ogni indirizzo anche la maschera a cui si riferisce. Un modo abbreviato per indicare una mask è la (**voce aggregata**) che postpone a ogni indirizzo un */N* dove N è il numero di bit di rete. Il CIDR funziona in questo modo: quando arriva un pacchetto indirizzato a una certa rete (divisa in sottoreti) viene preso e messo in AND con le mask delle sottoreti. Si assegna questo pacchetto alla sottorete che coincide con più bit.

4.5.5 NAT *(Network Address Translation)*

Soluzione che si è dovuta adottare, almeno fino alla completa adozione dell' IPv6, per risolvere il problema dell'esaurimento degli indirizzi IP. L'idea alla base è molto semplice: viene 'simulata' una rete in un unico indirizzo IP. Percui ogni azienda ha un solo indirizzo IP (o comunque pochi) per il traffico internet. Dentro all'azienda ogni pc riceve un indirizzo univoco per instradare i dati nella rete aziendale. Quando il traffico invece esce da questa rete avviene una traduzione dell'indirizzo a quello di Internet, effettuato dal **dispositivo NAT**. Sono stati riservati 3 intervalli di indirizzi che si possono utilizzare internamente:

- 10.0.0.0 - 10.255.255.255/8 (16.777.216 host)
- 172.16.0.0 - 172.31.255.255/12 (1.048.576 host)
- 192.168.0.0 - 192.168.255.255/16 (65.536 host)

Solitamente il dispositivo NAT è abbinato a un firewall, e a volte è integrato nel router. Il problema di questo protocollo sta a chi inviare internamente la risposta arrivata dal Web che come indirizzo ha quello generico aziendale! Quello che succede è questo: quasi la totalità dei pacchetti IP trasporta carichi utili TCP o UDP. Qui è segnata la porta sorgente e di destinazione del pacchetto e tramite la sorgente il NAT, grazie alla mappatura, riesce a risalire all'indirizzo dell'host che lo ha inviato. Questo trucco però è soggetto a varie critiche riguardo alla stratificazione dei protocolli, al legame con TCP/UDP e alla violazione del modello gerarchico IP.

4.5.6 Protocolli di controllo Internet

ICMP *(Internet Control Message Protocol)*

E' un sistema di messaggistica che viene attivato quando accade qualcosa di imprevisto o per test. Ogni messaggio è incapsulato in un pacchetto IP.

Tipo di messaggio	Descrizione
Destinazione irraggiungibile	Il pacchetto potrebbe non essere inoltrato
Tempo superato	Il campo 'Time to live' ha raggiunto valore 0
Problema di parametri	Campo dell'intestazione non valido
Spegnimento della sorgente	Pacchetto di interruzione
Reindirezzamento	Insegna al router la geografia
Eco	Chiede a una macchina se è viva
Risposta eco	Si, sono vivo
Richiesta di contrassegno temporale	Come eco ma con contrassegno temporale
Risposta di contrassegno temporale	Come eco ma con contrassegno temporale

ARP *(Address Resolution Protocol)*

Un problema che nasce con tutti questi indirizzi è l'associazione tra indirizzi IP e quelli degli altri strati. Questo viene effettuato tramite il protocollo ARP utilizzato in internet. L'idea alla base è questa: quando un host vuole sapere a che host corrisponde un certo indirizzo non fa altro che chiederlo in broadcast e ovviamente risponde solo la macchina che ha quell'indirizzo. Quindi il sistema non deve far altro che assegnare l'indirizzo IP, ARP si occuperà del resto. L'host1 prepara il pacchetto e inserisce l'indirizzo del destinatario. L'host2 riceve il frame vede che è indirizzato a lui lancia un interrupt e lo passa allo strato superiore. ARP intanto ha memorizzato nella cache l'indirizzo della sorgente nel qualcaso servisse successivamente. Un miglioramento a questa tecnica è che all'accensione ogni host invia la sua associazione in broadcast. Un problema nasce se 2 host non sono nella stessa rete poiché non riceve la trasmissione broadcast. Una soluzione potrebbe essere che il router sia configurato per rispondere alle richieste ARP dalla rete dell'host sorgente, così tutto il traffico diretto verso quella rete passa per il router che ha mappato IP e indirizzo Ethernet. Questa tecnica è detta **proxy ARP**. Un altro approccio si basa sul fatto che l'host1 si accorge subito che il pacchetto è destinato a una rete remota e allora il traffico viene trasmesso a un indirizzo Ethernet.

4.5.7 DHCP *(Dynamic Host Configuration Protocol)*

Protocollo che permette l'assegnazione manuale o automatica dell'indirizzo IP. L'idea alla base è semplice: esiste un server speciale che assegna gli indirizzi alle macchine che lo richiedono. Questo server può non trovarsi sulla stessa LAN del richiedente. E' buona norma comunque che in ogni LAN ci sia un **agente di inoltro DHCP**. Quando una macchina vuole il proprio indirizzo, all'accensione manda un pacchetto DHCP DISCOVER che viene catturato dall'agente. Quest'ultimo lo invia al server DHCP. Questa allocazione automatica per evitare di 'perdere' indirizzi viene fatta sottoforma di **leasing**, ovvero, l'indirizzo è associato temporaneamente e poi deve essere rinnovato.

4.5.8 OSPF *(Open Shortest Path First)* - Routing in internet

Esistono 2 tipi di routing: quello all'interno dell'AS (**interior gateway protocol**) e quello tra AS (**exterior gateway protocol**). Inizialmente il protocollo di routing per i gateway interni era sostanzialmente basato sul vettore delle distanze fondato sull'algoritmo Bellman-Ford, ma soffriva del problema del conto all'infinito. Nasce così OSPF che si basa sualcuni principi fondamentali:

1. Essere libero (aoerto).
2. Suppotro di diverse metriche di distanza
3. Dinamicità
4. Il routing doveva basarsi sul tipo di servizio (*type of service*)
5. Deve bilanciare il carico (quindi non sempre usare la strada migliore)
6. Supporto di sistemi gerarchici
7. Sicurezza

OSPF supporta 3 tipi di connessione e di rete:

1. Linee punto-punto tra 2 router
2. Reti multiaccesso con trasmissione broadcast
3. Reti multiaccesso senza trasmissione broadcast

Una **rete multiaccesso** è una rete con più router ognuno in grado di comunicare con gli altri. OSPF opera riassumendo tutta la rete in un grafo orientato con archi pesati, per poi calcolarne il percorso più breve. OSPF permette la divisione di AS grandi in piccole **aree** che rappresentano un loro grafo di rete. Ogni AS ha un area **dorsale** chiamata area 0. Tutte le aree sono collegate alla dorsale. Esiste poi in ogni area un router speciale che si collega alla dorsale. Con questa topologia abbiamo quindi 3 tipi di routing: intra-aree, tra aree e tra AS. Il primo è semplice poiché ogni router conosce la strada migliore per arrivare a un altro nella stessa area. Il secondo passa sempre 3 fasi: dalla sorgente alla dorsale, dalla dorsale all'area destinazione, infine al router destinazione. OSPF distingue 4 tipi di router:

1. i router interni che sono completamente dentro un area
2. i router di confine che collegano 2 o più aree
3. i router di dorsale che sono sulle dorsali
4. i router sul confine dell'AS che comunano con i router di altri AS.

Ogni router comunica con i router adiacenti che non significa che sono direttamente collegati. I tipi di messaggi che si inviano sono:
Tra AS invece non è più questo protocollo ad avere il controllo ma il BGP.

4.5.9 BGP *(Border Gateway Protocol)*

Il bisogno di cambiare protocollo per il routing fra AS è dovuto al fatto che si devono applicare molti più vincoli, del tipo:

- Nessun traffico di passaggio attraverso certi AS
- Mai mettere un certo stato x su un percorso che inizia da un altro y
- Non utilizzare un certo stato x per arrivare a y
- Passare per x solo se non esistono alternative
- Il traffico generato o in arrivo da x non dovrebbe passare per y

Dato lo speciale interesse per BGP al traffico di passaggio, le reti sono raggruppate in 3 categorie:

1. **Stub networks**: hanno una sola connessione al grafo BGP e non sono di interesse per il traffico di passaggio perché non portano in nessuna rete

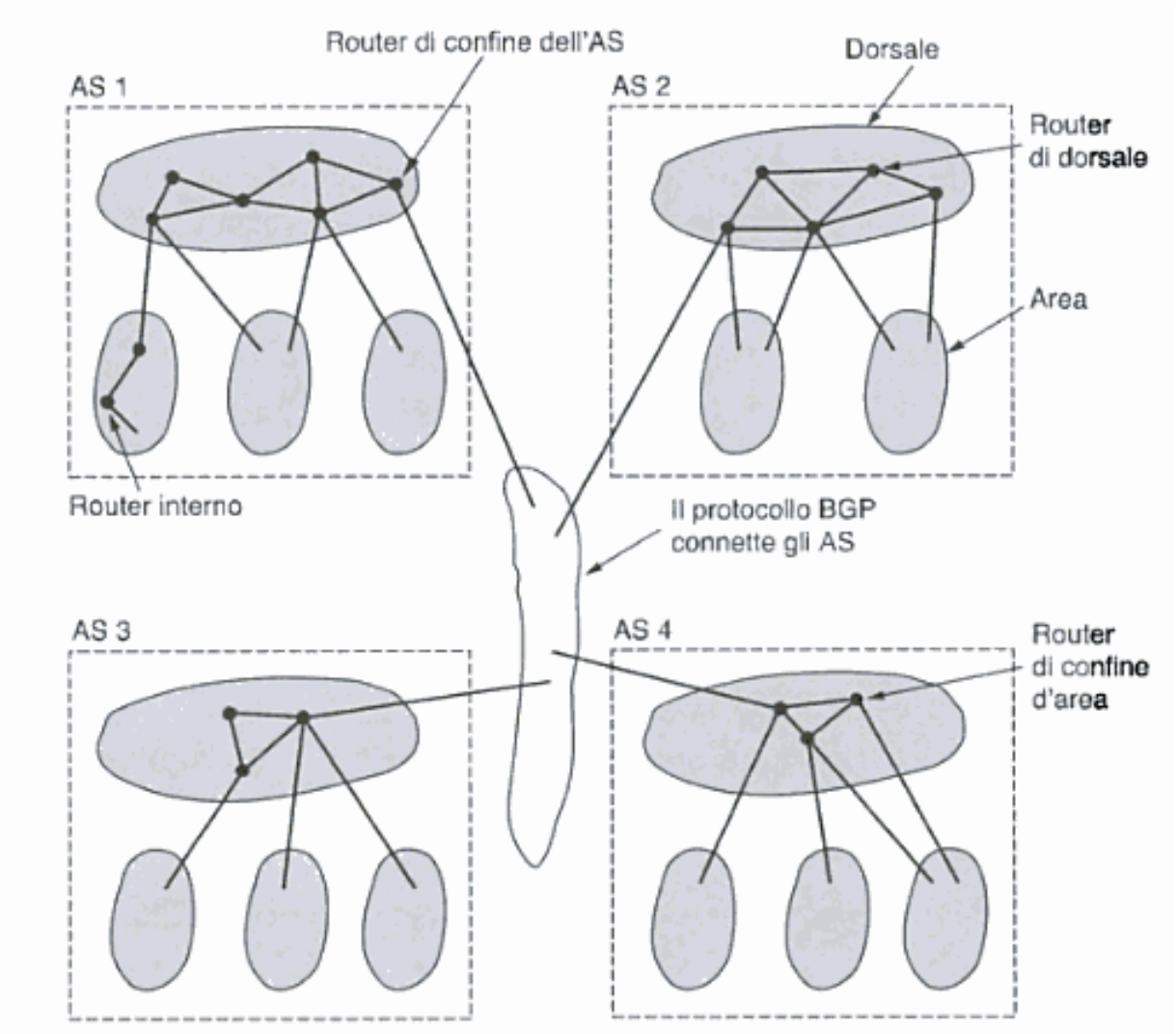


Figura 41: Realazione tra AS, dorsali e aree in OSPF.

Tipo di messaggio	Descrizione
Presentazione	Utilizzato per scoprire chi sono i vicini
Aggiornamento stato del collegamento	Comunica ai vicini i costi del trasmittente
Conferma stato del collegamento	Conferma l'aggiornamento dello stato del collegamento
Descrizione del database	Annuncia gli aggiornamenti posseduti dal trasmittente
Richiesta stato del collegamento	Richiede informazioni al partner

Figura 42: I tipi di messaggi OSPF.

2. **Multiconnected networks**: queste possono essere utilizzate a tale scopo, se lo permettono.
3. **Tranist networks**: che fungono da dorsale tra AS

Coppie di router BGP comunicano tra loro stabilendo connessioni TCP. Questo protocollo fondamentalmente è basato sul distance vector. Ogni router BGP oltre al costo di ogni destinazione tiene in memoria anche il percorso ed è questo che comunica ai router vicini periodicamente.

4.5.10 IPv6

Nonostante ci siano ancora degli escamotage per evitare la inevitabile fine di IPv4 nel 1993 l'IEEE si portò avanti con il protocollo che lo sostituirà definitivamente, ovvero l'IPv6. Quest'ultimo migliora il precedente sotto vari aspetti:

- Indirizzi più lunghi: IPv6 utilizza indirizzi da 16byte e non 4
- Semplificazione dell'intestazione: dai 13 campi del precedente ai 7 di IPv6
- Migliore supporto per le opzioni: dovuto al fatto che i campi tolti ora sono opzionali
- Sicurezza: autenticazione e riservatezza sono le chiavi del nuovo IP
- Riduzione delle tabelle di routing

- Lasciare spazio a evoluzioni future
- Permettere ai protocolli vecchi di coesistere con quello nuovo

Intestazione di IPv6

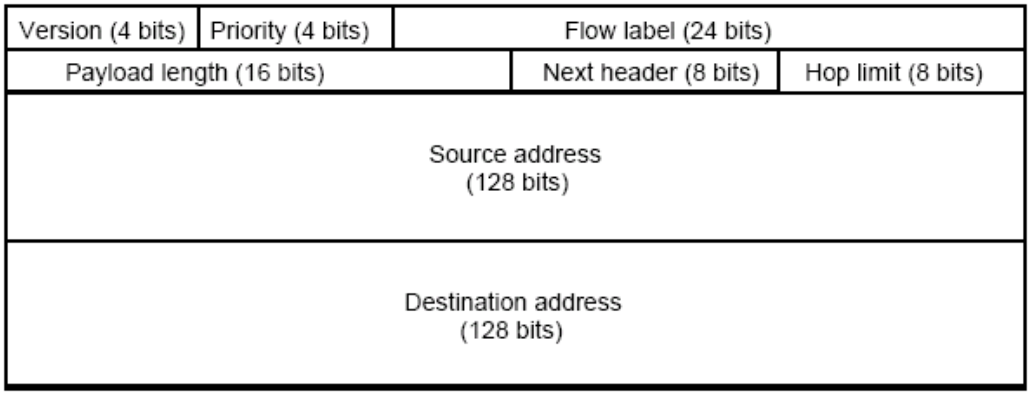


Figura 43: Intestazione IPv6

- **Version**: contiene ovviamente il valore 6.
- **Traffic class**: utile per distinguere i pacchetti con diversi requisiti di distribuzione.
- **Flow label**: in fase sperimentale, servirà per impostare pseudoconessioni con particolari proprietà.
- **Payload lenght**: indica il numero di byte che seguono l'intestazione.
- **Next header**: questo campo indica quale intestazione (tra 6 scelte, per ora) segue quella corrente.
- **Hop limit**: come il Time to live.
- **Source Address**: indirizzo sorgente.
- **Destination Address**: indirizzo destinazione.

Per scrivere gli indirizzi a 16 byte è stata pensata una particolare notazione: otto gruppi di 4 cifre esadecimali divise da ':'. Se un gruppo è di soli zeri si semplifica omettendoli. Gli indirizzi IPv4 sottoforma di IPv6 possono essere scritti in forma '::indirizzoIPv4'.

5 Lo strato trasporto

5.1 Stabilire una connessione

5.1.1 Handshake a tre vie

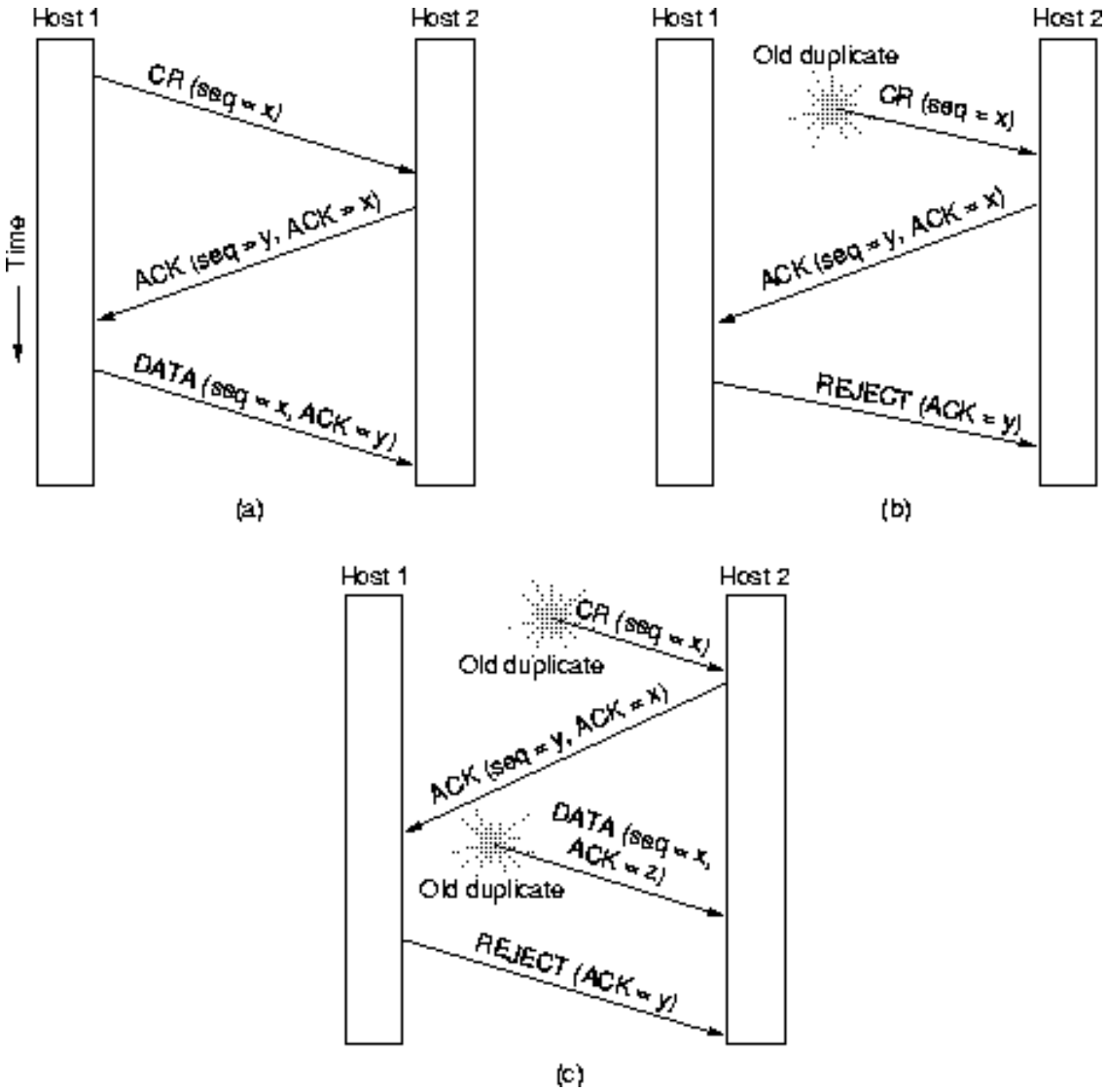


Figura 44: Handshake a 3 vie

E' un protocollo che non richiede sincronizzazione tra le parti ne con orologio di tipo locale ne globale. Funziona così: L'host1 sceglie un numero di sequenza x e invia una CONNECTION REQUEST (CR) con x all'host2.

L'host2 risponde con un ACK che riconosce x e annuncia il suo numero di sequenza y. Infine host1 riconosce y indicando sul suo primo invio dati con DATA. Nel caso di duplicati ecco cosa succede: mettiamo che una CR venga duplicata. L'host2 invia un ACK. L'host1 riconosce la situazione e invia un REJECT all'host2 che capisce di essere stato ingannato. Il caso peggiore avviene quando sia CR che ACK sono duplicati: come prima host2 riceve un CR duplicato e risponde a questo. Ora quando riceve il DATA l'host2 si accorge che è un duplicato grazie al numero di sequenza che non corrisponde a quello appena inviato!

5.2 Rilascio della connessione

Esistono 2 modi per terminare la trasmissione: il rilascio simmetrico e quello asimmetrico. Quello asimmetrico è quello utilizzato dal sistema telefonico, quando 1 dei 2 interlocutori riattacca cade la connessione. Quello simmetrico invece tratta la connessione come se fossero 2 unidirezionali. Il rilascio asimmetrico essendo improvviso può portare alla perdita di dati. Per evitarlo viene data la possibilità all'host che richiede di disconnettersi di ricevere ancora dati. Si può utilizzare un Handshake a 3 vie anche per la disconnessione ma se si esigesse di essere sicuri che anche l'altro interlocutore è pronto a disconnettersi non sarebbe possibile. Nella disconnessione bisgona accettare il rischio e l'handshake a 3 vie è adeguato.

5.3 UDP *(User Datagram Protocol)*

E' un protocollo di trasporto senza connessione. Trasmette **segmenti** costituiti da un intestazione di 8 byte seguita dal carico utile. L'intestazione UDP è formata da 2 byte di *Source port* seguiti da altri 2 di *Destination port* che indicano le porte a cui sono associati i processi di destinazione. Poi c'è il campo *UDP lenght* che include l'intestazione e i dati. Infine c'è il *Checksum* che è facoltativo e settato a 0 se non utilizzato. Un applicazione che fa uso di UDP è DNS, tipico esempio di situazione client/server. L'UDP predilige la velocità al controllo. Questo protocollo è utilizzato dal DNS.

5.4 TCP *(Trensmission Control Protocol)*

TCP è stato progettato per riuscire a garantire solide prestazioni nache in presenza di molti errori di vario genere. Un entità TCP accetta dai processi locali il flusso di dati degli utenti e li suddivide in pezzi di dimensione non superiore ai 64 KB e li invia in un datagramma IP autonomo. E' un protocollo orientato alla connessione. Sia il ricevente che il mittente devono creare dei

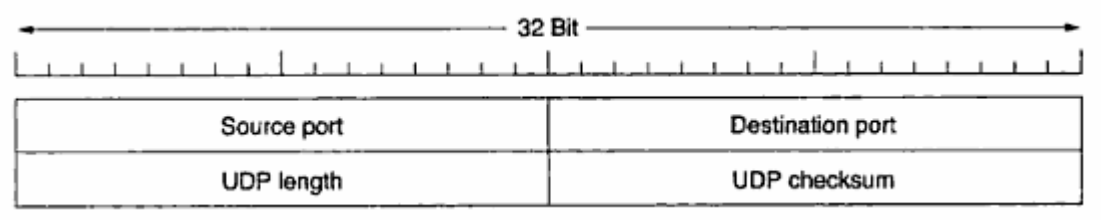


Figura 45: Intestazione UDO.

socket i quali possiedono un indirizzo (**porta**). Un singolo socket supporta più connessioni contemporanee. Le connessioni si identificano dalla coppia di porte dei socket (socketS, socketD). I numeri di porta minori di 1024 identificano le **well-known ports** riservate a servizi standard. Ogni byte in

Porta	Protocollo	Utilizzo
21	FTP	Trasferimento di file
23	Telnet	Login remoto
25	SMTP	Posta elettronica
69	TFTP	Trivial file transfer protocol
79	Finger	Ricerca di informazioni su un utente
80	HTTP	World Wide Web
110	POP3	Accesso remoto alla posta elettronica
119	NNTP	News di USENET

Figura 46: Alcune porte dei servizi standard.

una connessione TCP ha un numero di sequenza di 32 bit. Le entità TCP si scambiano dati sottoforma di segmenti. Un **segmento TCP** consiste in una intestazione di 20 byte seguita dai dati. Ogni segmento può essere al massimo 65536 byte. Inoltre ogni rete ha un **MTU** *(Maximum Transfert Unit)* e ogni segmento deve essere contenuto in questo. Il protocollo di base utilizzato è sliding window.

5.4.1 Intestazione TCP

- **Source port**: porta della sorgente
- **Destination port**: porta della destinazione
- **Sequence number**: numero di sequenza

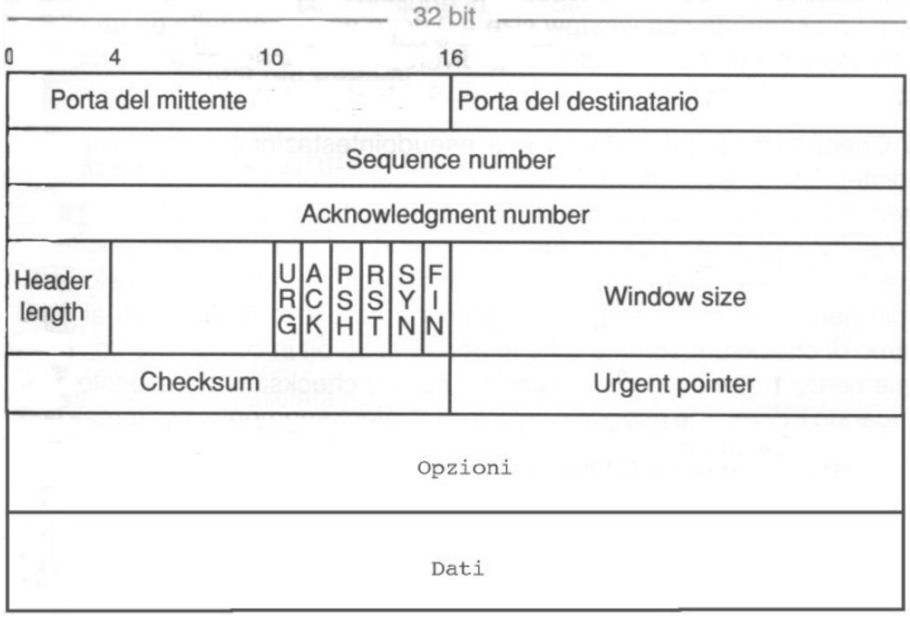


Figura 47: Intestazione TCP.

- **Acknowledgment number**: numero di ACK
- **TCP header length**: indica quante parole di 32 bit sono contenute nell'intestazione
- **URG**: flag impostato a 1 quando si usa l' *urgent pointer*
- **ACK**: flag impostato a 1 se l'ACK number è valido. Se a 0 il segmento non contiene ACK
- **PSH**: flag impostato a 1 se c'è presenza di dati PUSH.
- **RST**: flag impostato a 1 se si vuole resettare la connesione a causa di errori
- **SYN**: flag utilizzato per stabilire la connessione
- **FIN**: flag utilizzato per rilasciare la connessione
- **Window size**: indica la dimensione della finestra
- **Checksum**: che contiene la somma di controllo dell'intestazione, dei dati e della pseudo intestazione dei dati (vedi figura)
- **Urgent pointer**: indica l'inizio dei dati urgenti

- **Options**: utile per aggiungere informazioni extra
- **Dati**: contiene i dati veri e propri

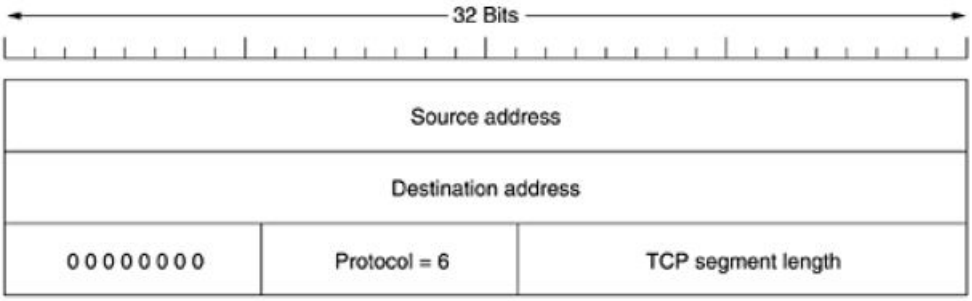


Figura 48: Pseudointestazione TCP

5.4.2 Connessione TCP

Il server resta in attesa con le primitive LISTEN e utilizza ACCEPT per accettare le connessioni in ingresso. Per connettersi un client esegue una primitiva CONNECT con SYN=1 e ACK=0 . Il server riceve l'entità TCP e controlla se sulla porta indicata esiste un processo e in caso negativo invia una risposta con RST=1. Altrimenti se si accetta viene restituito un segmento di ACK. In caso di richiesta contemporanea di connessione ne viene presa in considerazione solo una. Quando la connessione viene rilasciata si invia un segmento TCP con FIN=1. Quando si riceve l'ACK del FIN il flusso in quella direzione viene chiusa ma può continuare nel verso opposto.

5.4.3 Rilascio della connessione TCP

Per interrompere una connessione entrambe le parti possono inviare un segmento TCP con il bit FIN impostato in modo da segnalare che non hanno più nulla da trasmettere. Quando il FIN riceve l'ACK la connessione in quella direzione viene chiusa. Quando entrambe saranno chiuse allora la connessione viene rilasciata. Per evitare il **problema dei 2 eserciti** si usano dei timer in modo da rilasciare comunque la connessione anche in mancanza di un ACK di un FIN.

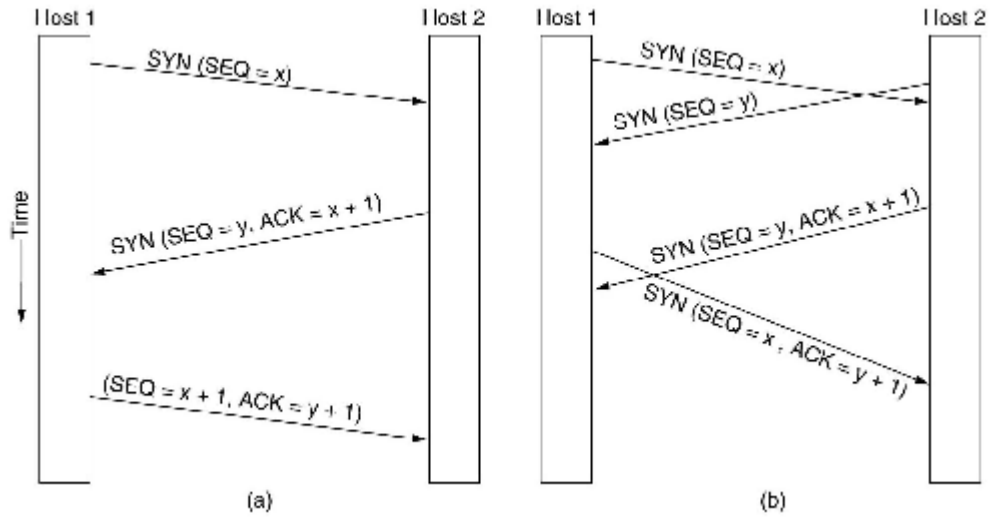


Figura 49: Connessione TCP

6 Lo strato Applicazione

6.1 DNS: il sistema dei nomi di dominio

Concettualmente internet è divisa in oltre 200 **domini** di primo livello. Ogni dominio è diviso in sotto domini e così via. Una struttura ad albero dove le foglie possono contenere un singolo host o può rappresentare una società e contenerne migliaia. I domini di primo livello sono di 2 tipi: generici e per nazioni. Inizialmente i generici erano *.com, .edu, .gov, .int, .mil, .net e .org* . Quelli per nazione avevano le iniziali caratteristiche della nazione. Successivamente vennero approvati altri domini di primo livello. Queste assegnazioni sono molto complesse mentre per i domini di secondo livello le cose sono più semplici. L'indirizzo è del tipo *nome.com*, per averlo è sufficiente contattare un **register** per il dominio di primo livello corrispondente, che controlla la disponibilità e quindi si paga una tariffa tipicamente annuale e il gioco è fatto. I nomi posso avere sio a 63 caratteri e il percorso completo non deve superare i 255.

6.1.1 Record delle risorse

A ogni dominio può essere associato un insieme di **resource records**. Per un semplice host il record di risorse è l'indirizzo IP, ma posso esisterne molti altri. Il DNS quindi deve occuparsi di associare il nome dei domini ai record.

Un record delle risorse è una quintupla:

NomeDominio TempoDiVita Classe Tipo Valore

- **NomeDominio**: indica il dominio cui si riferisce il record
- **TempoDiVita**: è un indicatore della stabilità del record
- **Classe**: per le informazioni di internet è sempre *IN*
- **Tipo**: specifica il tipo di record
- **Valore**: può essere una stringa ASCII o un dominio, la semantica dipende dal tipo

6.1.2 I server dei nomi

Lo spazio dei nomi DNS è diviso in **zone** non sovrapposte. Ogni zona contiene alcune parti dello spazio totale. Quando un risolutore ha un interrogazione su un nome di dominio, passa per un server dei nomi locali, il quale se trova la corrispondenza ritorna un **record autorevole**. Quest'ultimo è un record fornito dall'autorità che gestisce il record. Se invece non si trova in quel server quest'ultimo invia un messaggio ai server di primo livello per il dominio richiesto. Questo metodo è chiamato **interrogazione ricorsiva**.

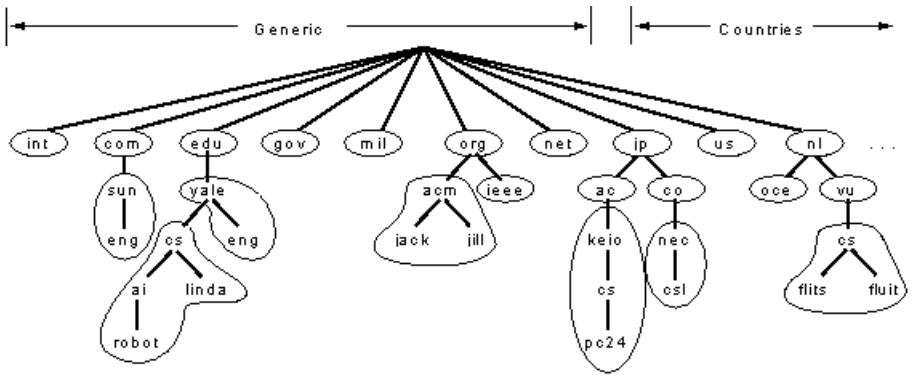


Figura 50: Spazio dei nomi DNS

7 Sicurezza

7.1 Crittografia

Bisogna innanzitutto distinguere 2 cose: **cifrario** e **codice**. Il primo è una trasformazione carattere per carattere. La seconda si intende un rimpiazzo di una parola con un altra.

7.1.1 Introduzione

I messaggi da decifrare sono detti **testo in chiaro**, sono trasformati tramite una funzione parametrizzata da una **chiave**. L'output è il **testo cifrato**. L'arte di decifrare i messaggi criptati è detta **criptoanalisi**, mentre l'arte di inventarli **crittologia**. Per cui **decriptare** è l'attività di **decifrare** non lecita.

$$C = E_K(P)$$

Si intende la cifratura di un messaggio P utilizzato una chiave K. Da cui:

$$P = D_K(C)$$

Quindi:

$$D_K(E_K(P)) = P$$

E e D sono quindi delle funzioni matematiche. Secondo il **principio di Kerckhoff** si deve supporre che il criptoanalista conosca gli algoritmi di decrittazione e criptazione (pubblici) e quindi il segreto resta esclusivamente nella chiave. Tenere segreto l'algoritmo è detto anche **sicurezza per occultamento**. Il segreto sta quindi in un buon algoritmo con chiavi 'lunghe' per aumentare il **fattore lavoro**.

7.1.2 Cifrari a sostituzione

Uno tra i cifrari più semplici, attribuito a Cesare, si basa sul principio di sostituzione di lettere con altre (**sostituzione monoalfabetica**), o più semplicemente 'shiftare' l'alfabeto di un numero k di caratteri (k in questo caso diventa la chiave). Nonostante la semplicità si può pensare che provare tutte le combinazioni sia proibitivo. Vero, ma non è l'approccio corretto per risolvere l'enigma. In questi casi si usa un approccio statistico che si basa sul fatto che le singole lettere, i digrammi (coppie di lettere) e i trigrammi (terne di lettere) in ogni lingua hanno una certa frequenza. Un altro approccio è quello di provare con una parola che dato il contesto ha una buona probabilità di essere nel testo e da li ricavare le varie lettere.

7.1.3 Cifrari a trasposizione

Questi cifrari semplicemente mascherano l'apparenza del testo in chiaro senza modificarlo. Si basano su 2 principi: il primo riguarda la scrittura in righe di N caratteri del messaggio, una sotto l'altra, in modo da creare colonne di M caratteri. Poi una chiave che ha il solo scopo di dare un ordine a come vengono prese tali colonne di caratteri per formare la parola criptata. Per attaccare un cifrario di questo tipo si vede se le frequenze delle lettere corrisponde e così si capisce che è a trasposizione poi bisogna capire di quante colonne è formato e questo la maggior parte dei casi è fatta grazie all'intuizione di una parole che ha molte probabilità di comparire nel testo. Infine bisogna trovare l'ordine delle colonne. Per farlo solitamente si prendono coppie di colonne e si controllano le frequenze dei digrammi e quella più corrispondente viene presa come la coppia giusta e poi si continua così con 3,4 fino alle k colonne. Questo metodo solitamente ha successo o al massimo si usano piccoli accorgimenti.

7.1.4 Blocchi monouso *(One-Time Pad)*

Un metodo imbattibile è prendere una stringa di bit, prendere il messaggio e fare uno XOR bit a bit con la stringa. Gli svantaggi evidenti sono: la chiave non può essere imparata, limita la lunghezza dei dati da trasmettere, impossibilità di capire eventuali errori.

7.2 Principi Crittografici fondamentali

Esistono 2 principi crittografici alla base dello studio dei sistemi di crittografia:

1. **Ridondanza**: tutti i messaggi cifrati devono contenere informazioni ridondanti, ovvero non necessarie alla comprensione del messaggio. Questo per evitare che intrusi generino dati casuali interpretati come validi messaggi.

2. **Attualità**: ogni messaggio deve avere il modo di vedere se è attuale. Questo per evitare che intrusi inviino messaggi vecchi spacciandoli per nuovi.

7.3 Algoritmi a chiave simmetrica

Dal nome si intuisce che per decifrare e cifrare si usa la stessa chiave. Un esempio ne sono i **cifrari a blocco** che prendono n bit dal testo in chiaro e li trasformano utilizzando una chiave a n bit. Questi algoritmi possono essere realizzati sia via sw che hw. Spesso questi algoritmi utilizzano metodi semplici come trasposizione e sostituzione con qualche accorgimento. Per permutare si usa una cosiddetta **P-box** che prende in input dei bit e li permuta, senza violare il principio di Kerckhoff, infatti si sa che è una permutazione ma non quale sia. Per la sostituzioni invece si usa una **S-box** che compie le sostituzioni. Collocando questi 2 dispositivi in cascata si ha il cosiddetto **cifrario prodotto**. Ogni iterazione di sostituzione utile per produrre un cifrario prodotto è detta **round**.

7.3.1 DES *(Data Encryption Standard)*

Il testo in chiaro viene cifrato con blocchi di 64 bit, con chiave a 56 bit e 19 passaggi. Il primo e l'ultimo sono trasposizioni, una il contrario dell'altra. Il penultimo consiste nello scambiare i 32 bit di destra con quelli di sinistra e i passaggi intermedi sono funzionalmente uguali ma parametrizzati con con diverse funzioni della chiave. E' strutturato in modo che la decifrazioni usi la stessa chiave di cifratura. Uno stadio intermedio funziona così (Vedi figura (b)): vengono presi in ingresso 2 blocchi da 32 bit e in output si ha come primi 32 bit i 32 bit di destra dell'ingresso e come secondi 32 sono il risultato dei 32 di sinistra in XOR con una funzione dei 32 di destra e della chiave dello stato in esame. La funzione finale è divisa in più passaggi:

1. Si espandono i 32 bit di destra in 48 (E) usando una regola fissa di trasposizione e duplicazione
2. Si esegue una XOR tra E e i 32 bit di destra
3. Si divide l'output in 8 gruppi da 6 e ognuno di questi viene dato in pasto a una S-box che restituisce 4 bit
4. Ognuno degli 8 gruppi di 4 bit viene dato in pasto a una P-box

Si usa una chiave diversa per ogni iterazione (16). Per rendere ancora più sicuro DES viene usata a volte la tecnica dello **sbiancamento** che consiste nel cifrare il testo in chiaro preliminarmente tramite una serie di 2 XOR tra blocchi di 64 bit del testo in chiaro con 2 chiavi a 64 bit.

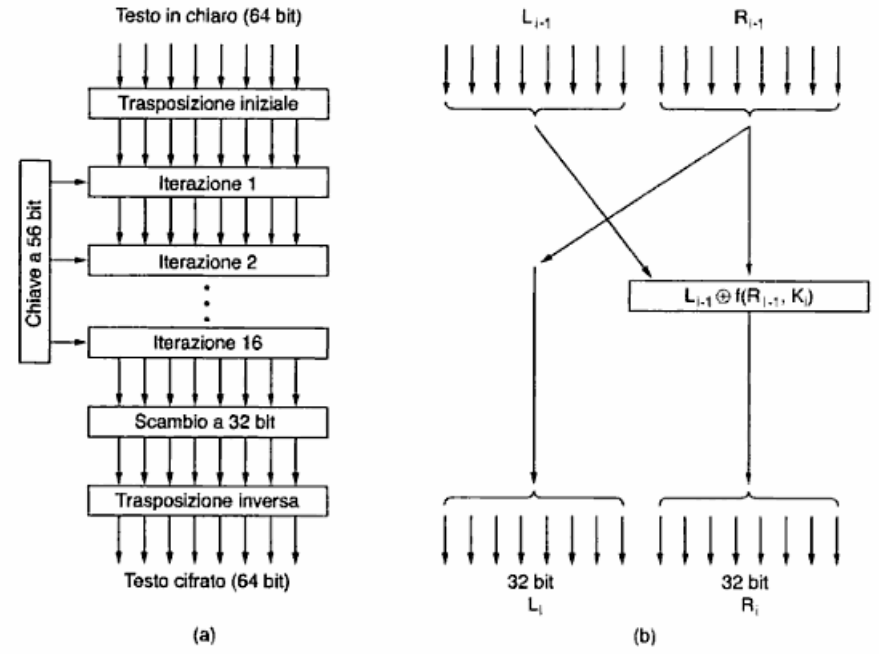


Figura 51: DES.

7.3.2 Triplo DES

Il triplo DES ha un funzionamento molto semplice: è diviso in 3 fasi, nella quale la prima è un DES in criptazione con chiave K1, poi viene usato un DES in decrittazione con chiave K2 e poi ancora in criptazione con K1 (Vedi figura). Si usano 2 chiavi e non 3 per una questione di risparmio. Si usano 2 criptazione e una decrittazione al posto che 3 criptazioni per una questione di retrocompatibilità con DES semplice.

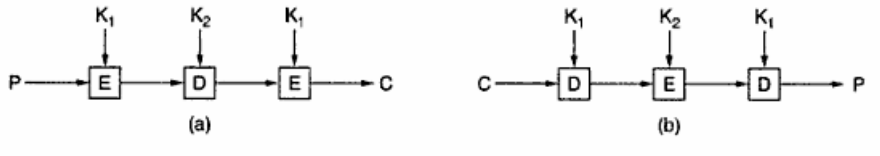


Figura 52: Triplo DES.

7.3.3 Blowfish

Il Blowfish è un cifrario a blocchi a chiave simmetricae si basa fondamentalmente su di un Feistel network, per chi no lo sapesse una **Rete Feistel** che

itera una certa funzione di encrypt un certo numero di volte (round) normalmente vengono fatti 16 rounds. Inoltre si basa su delle S-Box indipendenti da chiavi, e su una funzione F oneway, in altre parole non invertibile. La grandezza dei blocchi è di 64 bit, mentre la chiave può essere di una qualsiasi lunghezza purché non superi i 448 bits. Fino a questo momento non si conoscono tecniche di attacco sicure, inoltre blowfish è più veloce di IDEA e DES. L'algoritmo gestisce due tipi di liste di sottochiavi: una lista di 18 elementi,

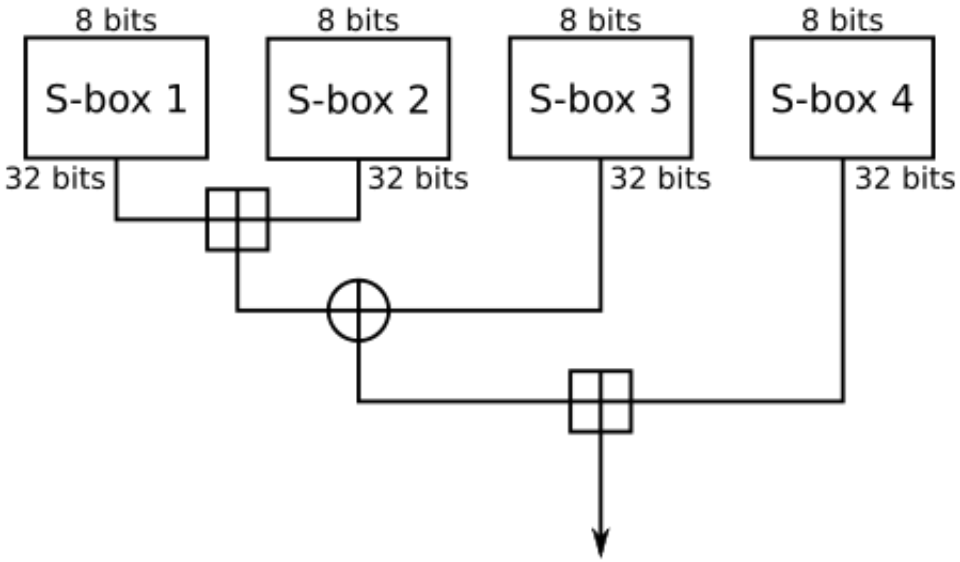


Figura 53: La funzione F di Blowfish.

definita P-array, e quattro lista da 256 elementi ciascuna di S-Box. Ciascuna S-Box ha in ingresso 8 bit di informazioni, e produce in uscita 32 bit. A ogni ripetizione del ciclo si usa un elemento diverso del P-array, e dopo l'ultimo ciclo a ogni metà del blocco di dati viene moltiplicata in XOR con uno dei due elementi inutilizzati del P-array. Dato che Blowfish è una rete di Feistel, può essere invertito semplicemente mettendo in XOR gli elementi 17 e 18 del P-array, e quindi utilizzando gli elementi del P-array in ordine inverso.

7.4 Modalità di cifratura

7.4.1 Modalità cipher block chaining

E' un metodo per evitare che vengano invertite parti di testo anche senza decifrare. Si basa sull'idea che ogni blocco ha un legame col successivo in modo che un loro spostamento fa perdere di significato al tutto. Questo metodo in particolare prende ogni blocco di testo e lo mette in XOR col precedente. Per il primo blocco lo XOR viene fatto con un blocco casuale detto **IV** (Initialization Vector) che è trasmesso insieme al testo cifrato (Vedi figura).

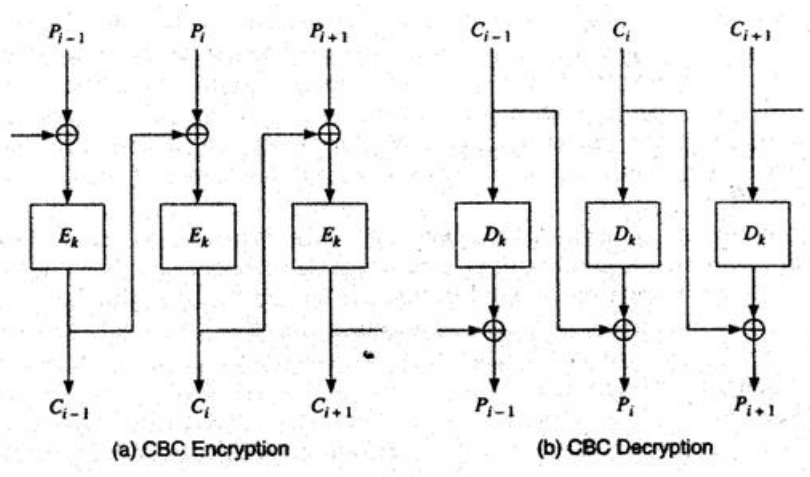


Figura 54: Cipher block chaining.

7.4.2 Modalità cypher feedback

Il metodo precedente ha lo svantaggio che per decifrare ha bisogno di avere l'intero blocco di 64 bit già scaricato. Questa tecnica (con triplo DES) invece una cifratura byte a byte e quindi più parallela della precedente. Per effettuare la criptazione viene utilizzato un registro di shift da 64 bit (se si usa il DES). Quando arriva un nuovo byte il DES agisce sul registro per generare 64 bit di testo cifrato. Si prende il byte più a sinistra e si mette in XOR con il nuovo blocco di testo in chiaro, ottenendo C. Il registro nel frattempo viene spostato di un byte facendo entrare in ultima posizione il risultato dell'ultima criptazione (C). Quindi ogni cifratura dipende dalle precedenti, 2 blocchi di testo uguali saranno cifrati in maniera differente.

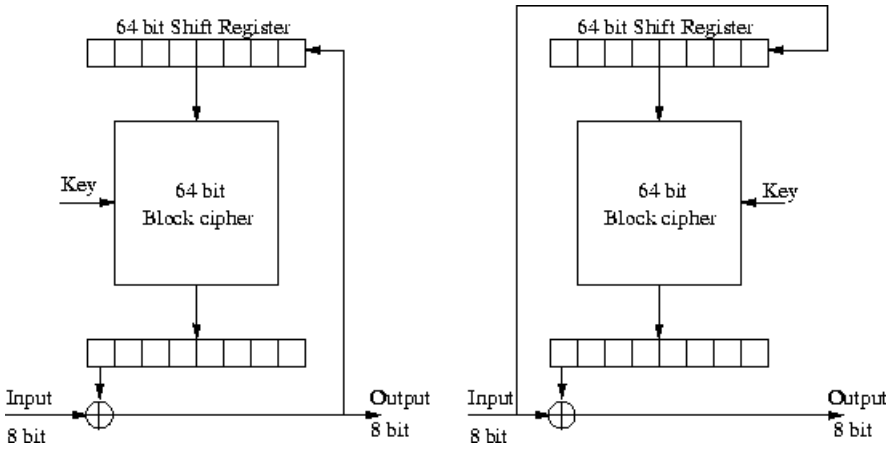


Figura 55: Cipher feedback.

La decifrazione è molto simile: il contenuto del registro di shift viene cifrati (non decifrato) così il byte viene messo in XOR con il byte C_i per ottenere P_i. Un problema si può avere se un bit viene invertito poiché entrerebbe nel registro di shift e quindi gli 8byte cifrati non saranno corretti.

7.4.3 Modalità stream cipher

Per ovviare al problema appena descritto viene utilizzato un metodo di cifratura diverso lo stream cipher. Questa funzione non fa altro che criptare un IV con una chiave crittografica ottenendo un blocco di uscita, R, il quale viene cifrato per ottenere il successivo blocco di uscita e cos' via. La sequenza di blocchi cifrati in uscita è chiamata **keystream**. L'IV viene utilizzato solo per la prima iterazione, poi sono i keystream a essere decifrati. L'uso di diversi keystream evita gli **attacchi di tipo keystream riutilizzato**.

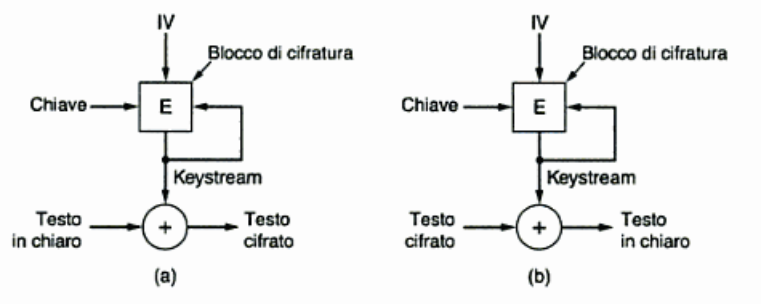


Figura 56: Stream cipher.

7.4.4 Modalità contatore

Questa modalità permette l'accesso casuale a dati criptati, quello che gli altri metodo non permettono. Con questo metodo il testo in chiaro non viene cifrato direttamente ma viene cifrato un IV con una chiave crittografica. Questo blocco cifrato viene messo in XOR con il testo in chiaro. A ogni blocco l'IV viene incrementato di 1. Questo sistema però è esposto ad attacchi di tipo keystream riutilizzati.

7.5 Algoritmi a chiave pubblica

Si basano su questi presupposti: sorgente e destinatario hanno 2 chiavi diverse (K, J) e usano rispettivamente E e D per criptare e decrittare. Vengono resi pubblici K, E_K eE_J, mentre sono segreti D_K, D_J. Gli algoritmi devono soddisfare queste proprietà:

- D(E(P)) = P
- Deve essere difficile ricavare D da E
- E non può essere forzato con un attacco di tipo 'testo in chiaro a scelta'

7.5.1 RSA

Richiede chiavi di almeno 1024 bit, per questo è abbastanza lento e rappresenta il suo maggior svantaggio. L'algoritmo funziona così:

1. Si scelgono 2 numeri primi molto grandi p e q (almeno 1024 bit)
2. Si calcoli $n = pq$ e $z = (p-1)(q-1)$
3. Scegliamo un numero coprimo con z, detto d
4. Troviamo e tale che $ed = 1 \ mod \ z$

Dividiamo in blocchi il testo in chiaro in modo che ogni messaggio P sia tale che $0 \leq P < n$. Per farlo basta raggruppare il testo in chiaro in blocchi di k bit con k tale che sia il massimo intero per cui vale $2^k < n$. Ora per cifrare il messaggio P calcoliamo $C = P^e(modn)$ mentre per decifrare c si calcola $P = C^d(modn)$. Questo vale perché $P = (x^e modn)^d modn = P^e dmodn$.

7.6 Firma digitale

E' equivalente alla firma 'analogica', ma in formato digitale. La firma può essere verificata e in più è vincolante.

7.6.1 Firme a chiave simmetrica

Si supponga l'esistenza di una autorità che conosce tutte le chiavi (BB). Ogni utente sceglie una chiave segreta e la comunica a BB in maniera sicura (a mano è meglio). Quindi Mimmo ha la propria chiave K_M che conosce lui e BB. Quando Mimmo vuole inviare un messaggio P firmato genera $K_M(B, R_M, t, P)$ dove B è il destinatario, R_M è un numero casuale usato

come token per evitare gli **attack replay**, t è un timestamp e P il testo in chiaro. Il messaggio viene inviato a BB, il quale lo decifra e lo invia a B con in più la firma ovvero $K_{BB}(A, t, P)$ con K_{BB} chiave dell'autorità. La debo-

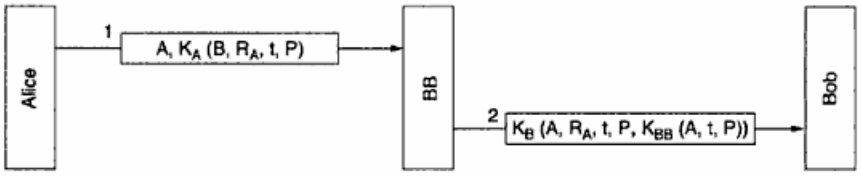


Figura 57: Firma digitale a chiave simmetrica.

lezza di questa firma sta negli attacchi a ripetizione. Per questo motivo sono usati i timestamp e il numero casuale più utile per le ripetizioni istantanee.

7.6.2 Firme a chiave pubblica

A volte non ci si vuole sempre fidare di BB e per quello si è pensato ad un modo senza intermedi. Gli algoritmi di criptazione e decrittazione devono essere tali che D(E(P)) = P e E(D(P)) = P. Per cui A decripta P con K_A e poi lo cripta con K_B. B riceve $E_{K_A}(D_{K_B}(P)) = P$ al quale applica D_{K_A} così ottiene $D_{K_B}(P)$ e assieme a P che ottiene decriptando forma la firma digitale. Il problema principale di questo metodo è che la chiave di chi firma deve rimanere segreta altrimenti non vale più nulla. L'algoritmo più utilizzato per questo tipo di firma è l'RSA. Un altro algoritmo buono per questo scopo è El Gamal che sfrutta la difficoltà di calcolare il logaritmo discreto di un numero e non la fattorizzazione come fa l'RSA.

7.6.3 Message digest

La funzione Hash MD ha queste proprietà:

- Dato P è facile calcolare MD(P)
- Dato MD(P) è quasi impossibile ricavare P
- Dato P, non si può trovare P' tale che MD(P')=MD(P)
- Se l'input cambia anche di un bit, l'output diventa completamente diverso

Per fare questo l'hash deve almeno essere di 128 bit e deve modificare radicalmente i bit in input. BB al posto di firmare con $K_{BB}(A, t, P)$, inserisce come quinto elemento di K_B $K_{BB}(A, t, MD(P)$ che viene mandato a B.

7.6.4 MD5

Prima di tutto l'algoritmo riempe il messaggio fino a fargli raggiungere una lunghezza di 448 bit. Il valore della lunghezza originale del messaggio viene aggiunta sottoforma di intero a 64 bit in modo da ottenere 512 bit, e si inizializza a un valore fissato un buffer di 128 bit. Ogni iterazione prende un blocco di 512 dall'input che viene modificato grazie ali 128 bit del buffer. Infine viene aggiunta una tabella costruita con la funzione seno. La scelta del seno non è casuale, in questo modo si è voluto scongiurare ogni sospetto riguardo all'uso di black door da parte dell'autore.

7.6.5 SHA-1 (*Secure Hash Algorithm*)

Lavora con blocchi da 512 ma restituisce un MD di 160 bit. Un esempio di utilizzo: il messaggio M viene usato come input per SHA-1 il quale viene firmato con la chiave privata RSA e trasmette a B sia M che l'hash. Ricevuto l'hash B lo ricalcola con M e applica la chiave pubblica di A all'hash firmato per ottenre l'hash originale H.

7.6.6 Birthday Attack

Questo attacco riduce il numero di operazioni medie per forzare un MD a m bit da 2^m a $2^{m/2}$. Se c'è una funzione fra input e output con n valori di input e k possibili calori di output, ci sono n(n-1)/2 coppie di input. Se queste coppie sono > k la possibilità di avere una coppia con lo stesso output è molto buona. Quindi approssimativamente basta avere un n > della radice di k. Quindi con 64 bit si ha una buona probabilità generando 2^{32} messaggi di trovarne 2 con lo stesso MD.

7.7 Gestione delle chiavi pubbliche

In base agli algoritmi descritti precedentemente A e B devono conoscere le loro rispettive chiavi. Ma qual'è il modo migliore per trasmettersi le rispettive chiavi?

7.7.1 Certificati

I Certificati è un documento che lega una certa chiave al suo proprietario. Questi certificati sono rilasciati da un organizzazione pubblica chiamata **Certification Authority**. Esistono certificati che non legano una chiave a una persona ma a un **attributo**. per es. questa chiave pubblica appartiene a

un maggiorenne. Questi tipi di certificati mantengono segreta l'identità del possessore.

7.7.2 X.509

E' uno standard per i certificati approvato dall'ITU ed è largamente utilizzato in internet. La sua funzione principale è quella di descrivere i certificati. I campi primari di un certificato sono riassunti in figura:

Campo	Significato
Version	Numero della versione di X.509
Serial number	Il certificato è univocamente identificato da questo numero più il nome della CA
Signature algorithm	Algoritmo usato per firmare il certificato
Issuer	Nome X.500 della CA
Validity period	Inizio e fine del periodo di validità
Subject name	L'entità proprietaria della chiave da certificare
Public key	La chiave pubblica del soggetto e l'ID dell'algoritmo che la usa
Issuer ID	Facoltativo: identificativo univoco di chi emette il certificato
Subject ID	Facoltativo: identificativo univoco del soggetto del certificato
Extensions	Sono state definite molte estensioni
Signature	La firma del certificato (firmata dalla chiave privata della CA)

Figura 58: I campi essenziali di un certificato.

7.7.3 Infrastruttura a chiave pubblica

Esiste un metodo per certificare le chiavi pubbliche chiamato **PKI** (*Public Key Infrastucture*) ed è composto da diversi componenti: gli utenti, la CA, i certificati e le directory. La funzione della PKI è di fornire una struttura per queste componenti e definire gli standard e i protocolli. Un esempio di PKI è la gerarchia delle CA. Per scoprire la validità di un certificato bisogna risalire lungo l'albero fino alla radice che si suppone tutti conoscano la sua chiave.

7.8 Sicurezza delle comunicazioni

7.8.1 IPsec (*IP Security*)

Si è deciso di rendere la cifratura una cosa non opzionale, ma dando la possiilità a chi non la vuole di utilizzare un algoritmo di criptazione nullo. IPsec offre molti servizi: segrettezza, integrità dei dati e protezione dagli attacchi a ripetizione. Questi servizi sono basati sulla crittografia a chiave simmetrica per le sue ottime performance. Offre inoltre molti algoritmi in modo

da poterli cambiare. Inoltre IPsec è orientato alla connessione, in questo contesto una connessione è definita come **SA** (*Secutiry Association*). Una SA rappresenta una connessione simplex quindi per avere una comunicazione sicura in entrambe le direzioni servono 2 SA. L'IPsec può essere utilizzato in 2 modalità:

1. **Modalità trasporto**: l'intesazione IPsec viene inserita subito dopo quella IP. Il campo protocol viene modificato per indicare che si sta utilizzando IPsec. L'intestazione IPsec contiene principalmente l'identificazione del SA, un nuovo numero di sequenza e eventualmente un controllo di integrità del payload.

2. **Modalità tunnel**: il pacchetto IP viene completamente incapsulato in nuovo pacchetto IP con una nuova intestazione. Lo svantaggio di questa modalità è il fatto che viene sostanzialmente modificata la dimensione del pacchetto originario.

La prima ulteriore intestazione è detta **AH** (*Authentication Header*) e garantisce il controllo di integrità e la sicurezza dagli attacchi. L'intestazione AH in IPv4 è messa tra l'header IP e TCP. Al suo interno ci sono vari parametri tra cui l'identificatore della connessione, il numero di sequenza e l'authentication data che contiene la firma digitale del payload. L'algoritmo di firma è scelto tra le parti tramite crittografia a chiave simmetrica. Un metodo semplice consiste nel calcolare un hash dei contenuti del pacchetto e della chiave senza trasmettere la chiave condivisa. Questa tecnica è detta **HMAC** (*Hashed Messsage Authentication Code*). Esiste un intestazione alternativa all'AH ovvero l'**ESP** (*Encapsulating Security Payload*) che ha funzioni simili (Vedi figura).

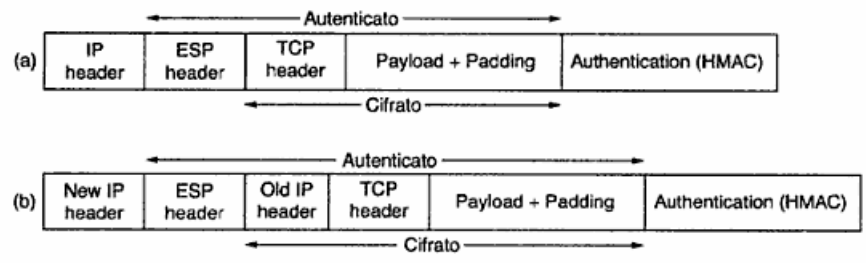


Figura 59: Intestazione ESP (a) modalità trasporto, (b) modalità tunnel.

7.8.2 Firewall

I Firewall hanno la funzione di filtrare il traffico in entrata e in uscita. Questo filtraggio è fatto grazie a 2 router e un gataway aplicativo. Esistono configurazioni più semplici ma questa è più sicura. Questi router sono detti **packet filter** e sono semplici router con qualche funzionalità extra. Il pacchetto in ingresso viene prima ispezionato dal primo router secondo alcuni criteri poi passa al gateway e infine al secondo router. Il gateway applicativo non ispeziona il pacchetto quanto tale ma in base a quello che rappresenta. I Firewall non riescono però ad evitare tutti gli attacchi per esempio un attacco **DoS** (*Denial of Services*) che ha l'intento di bloccare un certo servizio semplicemente sovraccaricandolo. Questo lo si può fare inviando richieste multiple di SYN senza poi rispondere. Una variante di questa tecnica è il **DDoS** (*Distribuited DoS*) che è uguale al precedente con la differenza che l'intruso è entrato in migliaia di macchine sparse per il mondo aumentando così il suo raggio d'azione.

7.8.3 Sicurezza di 802.11

Il protocollo di sicurezza prescritto dallo standard 802.11 è il **WEP** (*Wired Equivalent Privacy*). Quando in 802.11 la sicurezza viene attivata ogni stazione deve stabilire una chiave segreta con la base, il modo in cui questo avviene non è specificato. Un'altra possibilità è che la stazione base generi numeri causali e li invia con una trasmissione wireless cifrata usando la chiave pubblica del ricevente. La cifratura WEP usa uno stream cypher basato sull'algoritmo RC4 che genera un keystream che viene messo in XOR col testo in chiaro per produrre il cifrato. Per prima cosa viene calcolato il checksum del payload usando CRC-32 polinomiale. Il checksum viene aggiunto al payload per formare il testo da cifrare. Il testo viene messo in XOR con una parte del keystream. L'IV utilizzato per inizializzare il keystream viene inviato insieme al messaggio. La decriptazione è semplice: viene estratto il payload cifrato poi si genera il keystream con l'IV ricevuto e la chiave segreta condivisa, infine si calcola lo XOR con il testo cifrato. Lo standard WEP raccomanda che l'IV sia cambiato ad ogni invio anche se questa tecnica non evita gli attacchi (si pensi all'attacco del compleanno). Oltre a questa debolezza c'è la vulnerabilità dell'algoritmo RC4.

7.8.4 Sicurezza del Bluetooth

Bluetooth a 3 modalità di sicurezza che vanno dalla sicurezza 0 a controllo e integrità. Bluetooth presenta soluzioni di sicurezza a più strati: in quello fisico viene usato il salto di frequenza (frequency hopping). La vera sicurezza

si ha comunque quando uno slave comunica con il master. Si suppone che i dispositivi si siano scambiati le chiavi, a volte già inserita nei dispositivi. Questa chiave è detta **passkey**. Per stabilire una connessione si controlla che l'altro conosca la passkey. In caso affermativo viene stabilito se il canale deve essere cifrato e viene scelta una chiave di sessione casuale a 128 bit. La cifratura usa uno stream cypher detto E_0 e il controllo di integrità usa **SAFER+**.

7.9 Protocolli di autenticazione

Bisogna prima di tutto distinguere **autenticazione** che è la tecnica usata per verificare l'identità di un processo da autorizzazione che invece si occupa di stabilire cosa può o non può fare tale processo. Il modello generale utilizzato sa tutti i protocolli di autenticazione è del tipo: A invia un messaggio a B e a una terza persona fidata, detta **KDC** (*Key Distribution Center*). A e B inoltre stabiliscono una chiave segreta (**chiave di sessione**) da usare per continuare la conversazione. Tutto il traffico è cifrato con crittografia a chiave simmetrica, per motivi di prestazioni. Mentre per stabilire la chiave di sessione viene utilizzata quella a chiave pubblica.

7.9.1 Autenticazione basata su un segreto condiviso

Supponiamo che A e B conoscano la chiave segreta K_{AB}. Uno dei 2 invia un numero casuale all'altro che lo trasforma con un particolare algoritmo e lo ritrasmette (**challenge response**). Si veda la figura: A invia a B la sua

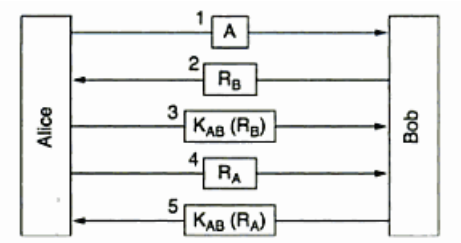


Figura 60: Autenticazione a 2 vie.

identità. B non essendo sicuro invia un numero casuale ad A. Questi numeri casuali usa e getta sono detti **nonce**. A cifra il messaggio di B con la chiave condivisa e glielo invia. Ora è A ad avere sospetti e invia un numero casuale a B. B lo cifra e glielo rimanda analogamente a prima. Questo protocollo può essere semplificato come in figura. Questo secondo metodo ha vantaggi ma anche svantaggi ovvero gli **attacchi per riflessione** (*reflection attack*).

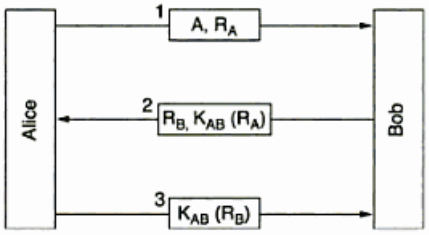


Figura 61: Autenticazione a 2 vie ridotto.

Reflection Attack

Un attacco per riflessione funziona così: indichiamo con T l'impostore. T finge di essere A e invia un numero casuale R_T a B che risponde con il suo numero casuale R_B e cripta R_T con la chiave comune. Ora T dovrebbe rispondere con criptando il numero casuale ma non conosce la chiave. Per riuscire a ricavarla T può riaprire una nuova sessione e mandare come numero casuale R_B che B gli ritorna criptato. Ora T ha quello che voleva e può completare la prima sessione. Questo attacco riesce a sorprendere anche l'autenticazione a 2 vie non ridotta: se A vuole dialogare con B invia la propria identità e T la intercetta. T fingendo di essere B apre una nuova sessione mandando la sua identità e A risponde con il suo numero casuale R_A. Ora T può continuare la sessione 1 inviando l'R_A appena ricevuto al quale A risponde con $K_{AB}(R_A)$ proprio quello che serve a T per continuare la sessione 2. Ora T può abbandonare la sessione 1 e dialogare come se fosse B nella 2 oppure può ripetere il procedimento appena descritto e avere così 2 sessioni autenticate.

7.9.2 Autenticazione con HMAC

A invia a B il suo R_A. B gli risponde con R_B e in più aggiunge un HMAC formato costruedo una struttura dati comprendente R_A, R_B le 2 identità A e B e la chiave comune K_{AB}. HMAC è calcolato facendo l'hash di questa struttura per esempio con SHA-1. A poi risponderà indipendentemente da B con un suo HMAC comprendente R_A, R_B e K_{AB}. Questo protocollo non è forzabile da T. Un alternativa all'HMAC è l'utilizzo del cipher block chaining per cifrare gli oggetti.

Replay Attack

E' una forma di attacco di rete che consiste nell'impossessarsi di una credenziale di autenticazione comunicata da un host ad un altro, e riproporla

successivamente simulando l'identità dell'emittente. In genere l'azione viene compiuta da un attaccante che s'interpone tra i due lati comunicanti. A differenza dell'attacco di tipo *man in the middle* che opera sempre in tempo reale, il replay attack può operare anche in modo asincrono quando la comunicazione originale è terminata. Si verifica un replay-attack quando Mallory intercetta la comunicazione tra Alice, che si sta autenticando con Bob, e si spaccia, agli occhi di Bob, per Alice. Quando Bob chiede a Mallory (convinto di parlare con Alice) una chiave d'autenticazione, Mallory pronta invia quella di Alice, instaurando così la comunicazione. Gli attacchi di tipo replay si evitano con l'uso di token di sessione generati pseudocasualmente: Bob invia ad Alice uno di questi token usa e getta, che Alice utilizza per criptare la propria chiave da inviare a Bob. Bob effettua lo stesso calcolo e controlla che il suo risultato corrisponda con quello di Alice. Un'altra contromisura è quella di utilizzare una marca temporale e di far sì che questa sia inserita nel corpo del messaggio criptato.

7.9.3 Lo scambio di chiavi di Diffie-Hellman

E' stato sempre supposto che le parti A e B avessero una chiave segreta condivisa ma come fanno a scambiarsela in modo veloce e sicuro? con il protocollo di **scambio di chiavi di Diffie-Hellman**. Questo protocollo funziona così: A e B si mettono d'accordo su 2 numeri grandi n e g dove n è primo e anche (n - 1)/2 è primo e g deve soddisfare certe particolari condizioni. A sceglie un altro numero segreto x che tiene per se. B analogamente sceglie y. A invia a B un messaggio contenente (n,g, g^x mod n) e B risponde con $g^y mod n$. Ora A calcola

$$(g^y \mod n)^x \mod n = g^{xy} \mod n$$

mentre B calcola:

$$(g^x \mod n)^y \mod n = g^{xy} \mod n$$

Ecco che le 2 parti hanno una chiave comune g^{xy} mod n. Il problema di questo protocollo è che è debole all'attacco noto come **man in the middle** o **attacco bucket brigade**.

Man in the middle attack

E' un attacco nel quale l'attaccante è in grado di leggere, inserire o modificare a piacere, messaggi tra due parti senza che nessuna delle due sia in grado di sapere se il collegamento sia stato compromesso. L'attaccante deve essere in grado di osservare e intercettare il transito dei messaggi tra le due vittime.

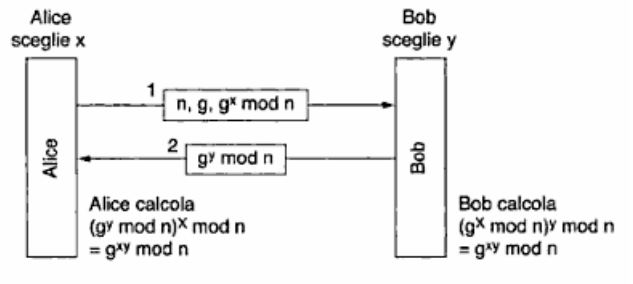


Figura 62: Protocollo Diffie-Hellman.

Supponiamo che Alice voglia comunicare con Bob, e che Giacomo voglia spiare la conversazione, e se possibile consegnare a Bob dei falsi messaggi. Per iniziare, Alice deve chiedere a Bob la sua chiave pubblica. Se Bob invia la sua chiave pubblica ad Alice, ma Giacomo è in grado di intercettarla, può iniziare un attacco Man in the middle. Giacomo può semplicemente inviare ad Alice una chiave pubblica della quale possiede la corrispondente chiave privata. Alice poi, credendo che questa sia la chiave pubblica di Bob, cifra i suoi messaggi con la chiave di Giacomo ed invia i suoi messaggi cifrati a Bob. Giacomo quindi li intercetta, li decifra, ne tiene una copia per sè, e li re-cifra (dopo averli alterati se lo desidera) usando la chiave pubblica che Bob aveva originariamente inviato ad Alice. Quando Bob riceverà il messaggio cifrato, crederà che questo provenga direttamente da Alice. Un simile attacco è possibile, in teoria, verso qualsiasi messaggio inviato usando tecnologia a chiave pubblica, compresi pacchetti di dati trasportati su reti di computer.

7.9.4 Autenticazione con crittografia a chiave pubblica

A chiede alla PKI, che contiene una Directory server, il certificato di B. La risposta è un certificato X.509 che contiene la chiave pubblica di B. A verifica la correttezza della firma e invia a B un messaggio che contiene la sua identità e un nonce. B ora non sa se A è davvero lei e ripete i primi 2 passi di A a ruoli invertiti. Quindi B invia ad A il suo nonce R_B quello che gli aveva inviato (R_A) e una proposta per la chiave di sessione K_S. Se A è daccordo invia in risposta a B $K_S(R_B)$. Qui T non riesce a fare nulla.

7.10 Sicurezza del Web

7.10.1 DNS spoofing

T, in grado di entrare nel sistema DNS, cambia l'IP di B con il proprio. Ora A chiede l'indirizzo di B, lo ottiene e gli chiede la sua home page ottenendola.

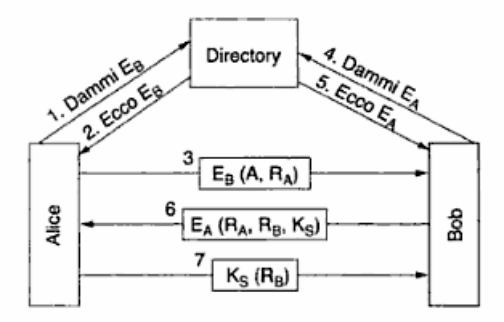


Figura 63: Autenticazione con chiave pubblica.

Avendo modificato l'IP T si finge B e A non lo sa. Ingannare il DNS è abbastanza semplice: T invia una query per l'IP di B al DNS, il quale usando UDP non ha modo di sapere chi fornisce la risposta a tale query. T falsifica la risposta e inserisce un falso IP nella cache del DNS (**poisoned cache**).