Appunti di Reti

Giacomo Manzoli, Antonio Cavestro, Simone Magagna

2 aprile 2014

Indice

Capitolo 1

Livello Fisico

1.1 Basi teoriche della comunicazione dati

1.1.1 Banda base e banda passante

Nessun mezzo di trasmissione è in grado di trasmettere segnali senza perdere parte dell'energia durante il processo. Ciò introduce delle **distorsioni**, a causa dell'attenuazione non uniforme, da parte del mezzo trasmissivo, delle armoniche di Fourier che compongono il segnale inviato. Ad esempio, di solito su un cavo le ampiezze sono trasmesse senza distorsioni da 0 fino a una certa frequenza f. L'intervallo delle frequenze trasmesse senza una forte attenuazione è chiamato **banda**. La **larghezza di banda** è la larghezza della banda delle frequenze. L'informazione trasportata dipende solo da questa larghezza e non dalle frequenze di inizio e fine.

I segnali che partono da 0 fino a una frequenza massima prendono il nome di **banda base**, mentre i segnali che vengono traslati per occupare una gamma di frequenze più alta vengono chiamati segnali in **banda passante**. In questo caso, il ricevente può traslarlo in banda base per elaborarlo, in modo che il riconoscimento dei simboli sia più semplice.

1.1.2 Baud rate e bit rate

Il **baud rate** indica il numero di simboli trasmessi in un secondo. Il **bit rate**, invece, indica il numero di bit trasmessi in un secondo. Possono sembrare la stessa cosa, dato che entrambi definiscono quanta informazione passa nel canale in un dato intervallo di tempo, ma ad un simbolo possono corrispondere più bit nel caso si usino tecniche di modulazione di ampiezza, frequenza o fase. In questo caso il bit rate è un multiplo del baud rate.

1.1.3 Teoremi di Nyquist e Shannon

Henry **Nyquist** dimostrò che anche un canale perfetto, senza rumore, ha una capacità di trasmissione limitata. Se si trasmette un segnale arbitrario ad ampiezza di banda B limitata (Nyquist parla di un segnale che passa attraverso a un filtro passa basso), **il segnale ricevuto può essere ricostruito completamente prendendo solo $2B$ campioni al secondo**. Se V è il numero di livelli che può assumere il segnale, il teorema di Nyquist afferma che:

$$velocità\ massima\ di\ trasmissione = 2B \log_2 V\ bit/s$$

Claude **Shannon** portò poi avanti il lavoro di Nyquist, estendendolo al caso in cui il canale trasmissivo è soggetto a rumore casuale, misurato facendo il **rapporto tra la potenza del segnale e la potenza del rumore** (SNR). Di solito si misura in decibel: $10 \log_{10} S/N$.

Dunque, il massimo tasso trasmissivo in un canale rumoroso con rapporto segnale-rumore pari a S/N e ampiezza di banda B è pari a:

$$velocità\ max = B \log_2 (1 + S/N)$$

1.2 Mezzi trasmissivi

Menzione semi-seria meritano i **mezzi magnetici**, ossia vedere come un canale di trasmissione l'uso di un nastro magnetico in cui salvare i dati che poi viene fisicamente trasportato dal ricevente. Effettivamente, trasportare scatola di nastri da 800 GB fino a una destinazione distante un'ora significa avere un'ampiezza di banda di che supera i **1700 Gbps**. In questo caso, però, l'ora di strada percorsa rappresenta la latenza, decisamente elevata. Vale la pena di citare il Tanenbaum:

> mai sottovalutare l'ampiezza di banda di una station wagon piena
> di nastri lanciata a tutta velocità lungo l'autostrada.

1.2.1 Mezzi Wired

Doppino

È composto da due cavi di rame **attorcigliati tra loro a forma elicoidale**, in modo da limitare le interferenze tra loro. Se non fossero intrecciati formerebbero un'eccellente antenna, mentre così i campi magnetici generati da entrambi i conduttori si annullano a vicenda. Si possono usare per trasmettere segnali sia analogici che digitali; l'ampiezza di banda dipende dal diametro del cavo e dalla distanza percorsa. In generale, un doppino può raggiungere una **lunghezza di alcuni chilometri** senza bisogno di amplificazione.

Cavo coassiale

Cavo composto da un nucleo di rame rivestito da materiale isolante, a sua volta rivestito da una calza metallica. Il tutto è poi ricoperto da una guaina protettiva. Presenta un'**eccellente immunità al rumore** grazie alla quale può estendersi per **distanze maggiori rispetto ai doppino**. Riesce ad avere una banda passante fino ad 1 GHz, dipendente da lunghezza, qualità e rapporto segnale-rumore.

Fibra Ottica

Un sistema ottico è composto da tre parti: sorgente luminosa, mezzo di trasmissione e rilevatore di luce.

- Sorgente di luce: è costituita da un LED o da un semiconduttore laser;
- mezzo trasmissivo: è la fibra, costituita da un nucleo di vetro avvolto in un rivestimento (cladding) anch'esso di vetro che ha un indice di rifrazione più basso rispetto al core. La luce che attraversa la fibra viene riflessa al suo interno fino ad arrivare all'altra estremità del cavo, in quanto, una volta immesso nel core, il segnale luminoso incontra in cladding che, avendo appunto un indice di rifrazione più basso, lo riflette completamente;
- rilevatore: è composto da un fotodiodo che produce un impulso elettrico quando colpito dalla luce. Il tempo di risposta dei fotodiodi limita la velocità di trasmissione dei dati, costituendo il **limite di banda** della fibra.

In una fibra possono coesistere più raggi che si riflettono in essa, caratterizzati ognuno da un angolo di riflessione diverso dagli altri. Questo tipo di fibre sono chiamate **multimodali**, mentre quelle che trasmettono un singolo raggio luminoso, in linea retta, sono dette **monomodali**.

Le fibre soffrono di un particolare fenomeno detto **dispersione cromatica**, che espande il segnale nella lunghezza d'onda durante la propagazione. Per evitare ciò, si è scoperta l'esistenza di impulsi di una particolare forma, detti **solitoni**, grazie ai quali è possibile annullare quasi tutti gli effetti della dispersione e trasmettere per migliaia di chilometri senza che la forma del segnale subisca modifiche sensibili.

Confrontando la fibra con i cavi in rame, **a vantaggio della fibra** troviamo:

- maggiore ampiezza di banda;
- immunità alle interferenze elettro-magnetiche;
- distanza massima maggiore;

- difficoltà nell'intercettazione dei dati, quindi più sicurezza;
- caratteristiche che la rendono più sottile e leggera.

A vantaggio del doppino, invece, troviamo una maggiore robustezza in quanto le fibre si possono danneggiare se eccessivamente piegate. Inoltre, la fibra richiede personale più qualificato per l'installazione e la manutenzione.

Per le loro caratteristiche, i doppini vengono usati nelle reti locali o nei local-loop delle linee telefoniche. Le fibre vengono maggiormente impiegate per le trasmissioni a lunga distanza, rimpiazzando via via i cavi coassiali. Inoltre, stanno iniziando ad essere utilizzate anche nel contesto locale, coprendo gli utenti fino all'armadio di quartiere (FttC) o direttamente fino a casa (FttH).

1.2.2 Trasmissioni satellitari

LEO

Sono i satelliti con l'**orbita più bassa**, per cui per comunicare con essi basta una parabola di piccole dimensioni, comportando un minor consumo energetico. Inoltre il ritardo del segnale dovuto al tempo di propagazione è minimo, nell'ordine di **pochi millisecondi**, e pure il costo del lancio è nettamente inferiore ad altre tipologie di satelliti. Essendo bassi è però necessario un **elevato numero di satelliti per coprire l'intera superficie** terrestre. Esempi di satelliti LEO sono i sistemi **telefonici** Iridium e Globalstar.

MEO

L'orbita MEO è situata **tra le due fasce di Van Allen**. In circa sei ore percorrono l'intera circonferenza terreste e la comunicazione con questi satelliti ha un ritardo che si aggira sui 100 ms. Il **primo satellite** della storia, lo Sputnik, volava in quest'orbita. Attualmente, viene principalmente utilizzata dai satelliti per la **navigazione GPS**, come il sistema del dipartimento della difesa statunitense Navstar e il russo Glonass.

GEO

Questa tipologia di satelliti si trovano su orbite alte, a circa 35mila Km d'altitudine. Essi orbitano attorno alla terra sull'equatore con una velocità tale da **rimanere sempre sopra la stessa area geografica**, da qui il termine di satelliti **geostazionari**.Possono essere presenti **solo 180 satelliti in quest'orbita**, sia per evitare interferenze sia per avere un certo spazio di manovra quando ne viene lanciato uno nuovo. L'allocazione degli slot orbitali è gestita dall'ITU.

Sono in grado di raggiungere dimensioni enormi (alcuni pesano 5 tonnellate) e consumare diversi kilowatt di energia, prodotta da pannelli solari.

Gravità solare, lunare e planetaria tendono ad allontanarli dagli slot e dagli orientamenti assegnati, ma l'effetto è contrastato dai motori a razzo installati a bordo. Dopo una **decina d'anni** il propellente dei motori si esaurisce e il satellite va alla deriva precipitando.

Per evitare interferenze con i sistemi a microonde a terra, l'ITU ha assegnato alle applicazioni satellitari alcune **bande di frequenza apposite**: C, L, S, Ku e Ka. I primi satelliti GEO con una singola emissione coprivano un'area grande circa un terzo della terra, chiamata **impronta**. Successivamente, con il progresso tecnologico, si è potuto concentrare i raggi trasmissivi in aree geografiche più piccole.

La trasmissione con questi satelliti richiede **maggiore potenza e la latenza è tra i 250 e i 300 ms**. Nonostante ciò possono essere usati come satelliti meteorologici, spia o per le trasmissioni broadcast come la **televisione satellitare**. Hotbird e Astra sono esempi di sistemi satellitari televisivi, in uso anche da colossi come Sky.

Un recente sviluppo del settore è rappresentato dalle microstazioni a basso costo chiamate **VSAT**. Possiedono piccole antenne del diametro di circa un metro e consumano pochi watt di potenza. A causa delle loro ridotte dimensioni, non sempre riescono a comunicare tra di loro (via satellite, logicamente). Per questo motivo, è necessario installare speciali stazioni terrestri chiamate **hub** che, dotate di grosse antenne ad alto guadagno, trasmettono il traffico attraverso le varie stazioni VSAT. Il compromesso per avere stazioni terminali utente **più economiche** è quello di un maggior ritardo di propagazione, ma questa tecnologia ha un **grande potenziale nelle aree rurali**.

Molniya

Sono satelliti che hanno un'**orbita ellittica**, grazie alla quale possono rimanere visibili in due punti della Terra per lungo tempo ottenendo un effetto simile al comportamento dei satelliti GEO anche al di fuori dell'equatore.

1.3 Modulazione digitale e multiplexing

1.3.1 Delta Modulation

È una tecnica di compressione di un segnale analogico. Il segnale viene campionato con una determinata frequenza e viene trasmesso un 1 se il segnale cresce, 0 se decresce. Può avere problemi con bruschi cambiamenti di livello.

1.3.2 Modulazione di ampiezza, frequenza e fase

Sono varie tecniche di trasmissione di informazioni in un segnale digitale.

Modulazione di ampiezza (ASK)

Nella modulazione di ampiezza per rappresentare simboli diversi vengono usate ampiezza d'onda diverse per rappresentare i livelli logici 0 e 1.

Modulazione di frequenza (FSK)

In questo caso, a frequenze diverse vengono associati simboli diversi. La modulazione in frequenza è complicata da gestire e inoltre nelle trasmissioni si cerca di tenere la frequenza maggiore in modo da avere un data rate più elevato.

Modulazione di fase (PSK)

Nella modulazione di fase l'onda portante viene traslata di 0 o 180 gradi all'inizio della trasmissione di ogni simbolo. La modulazione di fase funziona bene quando gli sfasamenti possibili sono pochi (Es: QPSK ha 4 sfasamenti possibili, multipli di 45 gradi), dato che se ci sono troppi simboli la distanza tra loro diventerebbe troppo piccola e basterebbe un piccolo errore per cambiare un simbolo in un altro. A livello pratico viene usata una combinazione della modulazione in fase e in ampiezza.

1.3.3 QAM

QAM, *Quadrature Amplitude Modulation*, combina l'utilizzo di PSK con la modulazione in ampiezza. Questo perché solo una tra frequenza e fase può essere modulata ogni volta, in quanto esse sono correlate: la frequenza è infatti la variazione della fase nel tempo. Quindi, sono ampiezza e fase a venire modulate, in maniera combinata.

Le modulazioni QAM vengono rappresentate mediante un grafico a coordinate polari, detto costellazione, dove l'ampiezza di ogni simbolo è associata al modulo mentre lo sfasamento all'angolo.

QPSK

Fa uso di quattro traslazioni, 45, 135, 225 e 315 gradi per trasmettere due bit d'informazione per simbolo.

QAM rettangolari e circolari

In QAM-16, vengono usate 16 combinazioni di ampiezza e fase, trasmettendo 4 bit per simbolo. In QAM-64, invece, vengono usate 64 combinazioni, codificando 6 bit per simbolo. Oltre a differire per il numero di simboli, i QAM differiscono anche per la forma delle loro costellazione, che può essere rettangolare (QAM-16-64-256) oppure circolare (QAM-8). Quelli circolari sono detti anche ottimali perché sfruttano al meglio lo spazio tra i simboli, nella

pratica però vengono usati quelli rettangolari dato che sono più semplici da gestire.

1.3.4 Multiplexing

La modulazione del segnale può avvenire anche per convogliare più conversazioni in uno stesso canale, cosa che le aziende praticano abitudinariamente per ridurre i costi ed ottimizzare le risorse.

Multiplexing a divisione di frequenza (FDM)

Lo spettro di frequenza è diviso in bande e ogni utente ne possiede una.

Multiplexing a divisione di tempo (TDM)

Gli utenti, a turno (round-robin), prendono possesso completamente del canale per un tempo limitato.

Multiplexing a divisione di codice (CDMA)

Permette la trasmissione continua attraverso l'intero spettro. Il punto centrale di tale tecnica consiste nell'essere in grado di **estrarre un particolare segnale e rifiutare tutto il resto come rumore casuale**. È l'esempio del party internazionale descritto dal prof. Marchiori a lezione.

Il tempo di trasmissione di ogni bit viene suddiviso in m intervalli più brevi chiamati **chip**. A ogni stazione viene assegnato un codice univoco di m bit chiamato **chip sequence**. Per trasmettere un bit di valore 1, una stazione invia la sua sequenza di chip, mentre per trasmettere uno 0 ne invia la negazione. Tutte le sequenze di chip di stazioni diverse sono **ortogonali a coppie**: il prodotto interno normalizzato di ogni coppia distinta di sequenze di chip S e T è pari a zero.

Sia $\overline{T}$ la negazione del vettore T. Si noti che se $S \times T = 0$ allora $S \times \overline{T} = 0$ e che $S \times S = 1$ e $S \times \overline{S} = -1$. Per recuperare la sequenza di bit generata da una certa stazione il ricevente deve conoscere in anticipo la sequenza di chip di quella stazione. Il recupero si effettua calcolando il **prodotto interno normalizzato tra la chip sequence ricevuta** (che è la somma di tutte le sequenze di chip delle varie diverse stazioni che trasmettono contemporaneamente) **e la sequenza della stazione di cui vogliamo recuperare il messaggio**. Se la sequenza ricevuta è $M = A + B + C$ e il ricevente vuole parlare con C, allora basta calcolare $M \times C = (A \times C) + (B \times C) + (C \times C) = 0 \times 0 \times 1 = 1$. Se invece $M = A + B + \overline{C}$, allora $M \times C = 0 + 0 + (-1) = -1$. Se invece $M = A + B$ e quindi C non ha trasmesso nulla, $M \times C = 0 + 0 = 0$.

Per ottenere queste particolari chip sequence, vengono usate le matrici di Hadamard quadrate (dette anche matrici di Walsh), le cui righe hanno

la caratteristica di essere formate da +1 o -1 tali che il prodotto scalare tra due righe distinte sia 0. Basta poi assegnare ad ogni stazione una riga della matrice. CDMA assume che tutte le stazioni trasmettano con la stessa potenza, perciò ognuna deve essere in grado di variare la potenza del segnale che emette basandosi sulla distanza dalle altre.

Esempio: Base trasmissiva per 18 stazioni. Basta calcolare una matrice di Hadamard 32x32 e prendere solo 18 righe, ogni riga definisce un codice di trasmissione per una stazione.

1.4 Telefonia mobile

1.4.1 Hard/Soft Handoff

L'handoff si verifica quando una cella riceve un segnale troppo debole da un cellulare. La cella interroga le celle vicine per sapere quale riceve il segnale migliore dal cellulare e ne passa ad essa il controllo.

Nella versione **hard** la vecchia cella disconnette il cellulare e quella nuova lo riaggancia, quindi se c'è una chiamata in corso cade la linea dato che questa procedura può richiedere diverso tempo, fino a 300 ms. Nella versione **soft**, invece, la nuova cella aggancia il cellulare prima che la vecchia lo molli. È però necessario che il cellulare sia in grado di gestire contemporaneamente due frequenze diverse.

1.4.2 GSM

Appartiene alla **seconda generazione** delle comunicazioni mobile. È lo standard usato in Europa e si basa sulla creazione di canali di comunicazione mediante FDM. Ci sono 124 canali full duplex da 200 KHz l'uno, realizzati mediante due canali simplex. Ogni canale è in grado di ospitare 8 slot TDM rendendo possibile tenere un elevato numero di utenti, grazie anche al fatto che il segnale trasmesso è di tipo digitale, quindi comprimibile.Ogni canale offre 270 Kbps, 33 ad utente, che però solo 13kbps sono utilizzabili, considerando l'overhead dei pacchetti. Ci sono 4 tipi di celle: Macro, Micro, Pico, Indoor.

Inoltre, questo protocollo introduce le SIM (*Subscriber Identity Module*) che, oltre a staccare il numero telefonico dal cellulare, contiene altri informazioni quali IMSI e Ki: IMSI identifica univocamente la SIM mentre Ki è una chiave usata per criptare la comunicazione tra la SIM e la cella. L'autenticazione funziona in questo modo:

1. il cellulare manda l'IMSI della sua SIM in broadcast;
2. l'operatore lo riceve e trasmette un numero casuale;
3. il cellulare lo rimanda firmato con la propria Ki;

4. l'operatore verifica l'identità del cellulare.

Nel sistema GSM vi sono più canali:

Canale di controllo broadcast È un flusso continuo di dati trasmessi dalla stazione base, che annuncia identità e stato del canale.

Canale di controllo dedicato È utilizzato per aggiornare la posizione, registrare il terminale nella rete e configurare la chiamata.

Canali di controllo comune Paging Utilizzato dalla stazione per annunciare le chiamate in arrivo.

Accesso casuale Permette agli utenti di richiedere uno slot sul canale di controllo dedicato.

Assegnazione di accesso Assegna il canale di controllo dedicato.

1.4.3 GPRS

Nasce come overlay del 2G: introduce la gestione del traffico a pacchetti, risolvendo alcuni problemi riguardanti la trasmissione dati mediante GSM. Quest'ultimo infatti era stato pensato solamente per trasmettere voce e se lo si voleva utilizzare per trasmettere dati era necessario riservare un canale, portando ad uno spreco di risorse dato che la navigazione web per sua natura ha molte pause.

Con GPRS non si spreca banda in quanto si può usare un canale condiviso per accedere a Internet e si può tariffare l'utente in base al traffico anziché in base al tempo. Le celle allocano dinamicamente i canali voce e i canali internet in base alle richieste. Ci sono varie classi di cellulari GPRS:

Classe C Si possono connettere come GSM o GPRS, l'utente deve settare manualmente quale usare

Classe B Si possono connettere come GSM o GPRS ma il cambio viene effettuato automaticamente in base al tipo di utilizzo. È la classe più comune.

Pseudo Classe A Vengono usate entrambe le modalità contemporaneamente usando una sola frequenza (*Dual Transfer Mode*). Deve essere supportato anche dalla rete.

Classe A Vengono usate contemporaneamente su due frequenze diverse.

1.4.4 UMTS (WCDMA)

WCDMA, *wideband* CDMA, è lo standard di terza generazione adottato in Europa, ribattezzato dall'UE con il nome UMTS, *Universal Mobile Telecomunication System*. Usa canali da 5MHz ed è compatibile con la rete 2G GSM.

Come suggerisce il nome, WCDMA si basa su CDMA, con alcune correzioni dovute alla natura delle comunicazioni mobili. Il problema più importante è dovuto alla **mancata sincronizzazione** tra i vari trasmettitori, cosa che il CDMA classico esige per ottenere una perfetta ortogonalità tra le varie chip sequence. Si è risolto adottando sequence che avessero alte probabilità di avere una **bassa correlazione incrociata**, cioè lunghe sequenze pseudocasuali che fossero ortogonali **a livello probabilistico**.

Naturalmente, come nel CDMA classico, **tutti gli apparecchi devono trasmettere con la stessa potenza**. È compito della stazione base regolare i trasmettitori, inviando loro informazioni sull'energia da usare per comunicare. Un'altra correzione effettuata rispetto al CDMA teorico consiste nel permettere agli utenti di usare **velocità di trasmissione distinte**: questo si ottiene fissando il tasso con cui vengono trasmesse le chip e assegnando agli utenti sequenze di chip di lunghezze differenti.

Sono tre i vantaggi principali che hanno portato all'uso di CDMA nelle reti telefoniche mobili di terza generazione:

1. è in grado di aumentare la capacità trasmissiva sfruttando i brevi periodi in cui i trasmettitori sono silenziosi (ci sono molte pause nelle comunicazioni). In CDMA meno trasmettitori significa meno interferenze;

2. ogni cella usa le stesse frequenze. Questo elimina una complicata pianificazione delle frequenze e aumenta la capacità trasmissiva, avendo più banda per canale;

3. facilità il soft handoff: il fatto di usare tutte le frequenze in ogni cella facilita il lavoro.

WCDMA non è l'unico standard 3G in circolazione: negli Stati Uniti troviamo CDMA2000, simile per certi aspetti, tranne per la mancata retrocompatibilità con GSM. Negli USA questo sistema non è molto diffuso, optando quindi per la compatibilità verso il loro IS-95.

Capitolo 2

Data Link

Le funzioni principali di questo secondo livello dell'architettura di rete sono:

- definire l'interfaccia per lo strato network;
- controllo degli errori;
- controllo di flusso.

I pacchetti che arrivano dallo strato soprastante (lo strato di Rete), vengono incapsulati in **frame** aventi un header, un corpo e una coda.

2.1 Framing

2.1.1 Conteggio caratteri

Questa tecnica prevede l'inserimento nell'header di un campo contenente la lunghezza del frame.

2.1.2 Byte Stuffing

E' una tecnica di delimitazione dei frame che consiste nell'usare un particolare byte per indicare l'inizio e la fine del frame. Può verificarsi il caso che all'interno del campo dati del frame compaia il byte delimitatore, di conseguenza è necessario utilizzare un altro byte di escape per evidenziare che il byte che lo segue non è di controllo ma è un dato. Nel caso si verifichi un errore di trasmissione si perdono al massimo 2 frame.

2.1.3 Bit Stuffing

Tecnica molto simile al byte stuffing, solo che lo stuffing viene fatto a livello di bit anziché byte. Si usa per esempio il pattern 01111110. Nel caso dentro al campo dati del frame ci siano cinque 1 consecutivi, in codifica si aggiunge uno 0 mentre in decodifica (se dopo i cinque 1 c'è uno 0) lo si toglie. In questo modo il pattern delimitatore non può comparire dentro i dati del frame.

2.2 Rilevazione e correzione degli errori

2.2.1 Codici a correzione di errore

Codice di Hamming

In questo codice, un frame consiste di m bit di dati (il messaggio) e r bit di ridondanza, cioè di controllo. Se la lunghezza del blocco è $n = m + r$, denotiamo il relativo codice come un codice (n, m).

Date due parole del codice, è possibile sapere quanti bit sono differenti calcolando l'OR esclusivo tra le due e contando il numero di bit pari a 1 nel risultato. Questo numero è detto **distanza di Hamming**. La distanza di Hamming in un intero codice, inoltre, è pari al minimo delle distanze per ogni coppia di parole. Il **significato** di questa distanza è che, se due parole sono a distanza d una dall'altra, saranno necessari d **errori su singoli bit per convertire una sequenza nell'altra**.

Questa distanza gioca un ruolo fondamentale nella gestione degli errori, infatti, per **rilevare** d errori di trasmissione è necessaria una codifica con distanza di Hamming pari a $d+1$, in quanto il codice garantisce che d errori non possano cambiare una parola di codice valida in un'altra anch'essa valida. Per poter invece **correggere** d errori è necessario un codice con distanza pari a $2d + 1$: in questo caso, le parole di codice valide sono così distanziate tra loro che anche con d cambiamenti la parola originale è sempre la più vicina secondo la distanza di Hamming.

Poiché, nel caso generale, tutte le parole devono essere valutate, tale ricerca ha un **costo di esecuzione molto alto**. Nella pratica, i codici sono progettati secondo algoritmi che sfruttano *trucchi algebrici* per velocizzare la ricerca. In un messaggio, i bit nelle posizioni che sono potenza di 2 sono **bit di controllo**, mentre gli altri vengono usati per gli m bit di dati. Ogni bit di controllo forza la somma modulo 2, o parità, di alcuni gruppi di bit incluso se stesso, a essere pari (o dispari, a seconda della convenzione). Dato un bit dati in posizione k, per conoscere quali bit di controllo influenza, riscriviamo k come somma di potenze di due. Un bit dati è controllato solo dai bit di controllo presenti nella sua espansione in somma di potenze di due.

2.2.2 Codici a rilevazione di errore

Questi codici **rilevano solamente la presenza** di errori nei dati ricevuti, ma non li correggono.

Parity Bit

E' la tecnica più semplice di error detection e consiste nell'aggiungere 1 bit di parità ogni m bit di dati e permette di rilevare errori con distanza 1. Il data rate è pari a $\frac{m}{m+1}$.

Checksum

Gruppo di bit di controllo associati al messaggio. Di solito, è posizionato alla fine del messaggio e calcolato come **complemento a uno del risultato della somma dei dati**. In questo modo gli errori possono essere rilevati sommando l'intera parola contenente sia i bit dati sia il checksum.

CRC

È una tecnica di error detection basata sull'**aritmetica polinomiale** in base 2. Una stringa di m bit viene interpreta come un polinomio di grado $m-1$ che ha come coefficienti i bit della stringa.

La sorgente e la destinazione si accordano su un **polinomio generatore** $G(x)$, che deve avere come primo ed ultimo bit 1. Il frame da controllare $M(x)$ deve essere di ordine maggiore di $G(x)$. L'idea è aggiungere una specie di checksum alla fine del frame in modo che il polinomio rappresentato dal frame, checksum compreso, sia divisibile per $G(x)$.

Il calcolo del checksum avviene in questo modo:

1. sia r il grado di $G(x)$. Si aggiungono r zeri alla fine del frame, in modo da ottenere una sequenza $x^r M(x)$ composta da $m+r$ bit;
2. si divide tale sequenza per $G(x)$ tramite la divisione in modulo 2;
3. si sottrae il resto ottenuto dalla divisione a $x^r M(x)$, sempre in modulo 2;
4. la sequenza risultante sarà il frame con il checksum pronto per la trasmissione.

La destinazione riceve il polinomio e prova a dividerlo per $G(x)$, ossia eseguo uno shift a sinistra di r bit. Se c'è **resto allora si è verificato un errore**. Dato che un errore di trasmissione può essere visto come sommare un polinomio E(x) al frame, allora è possibile fare error detection per tutti gli errori E(x) che non sono divisibili per G(x).

Esempio:

$$M(x) = 10011101$$

$$G(x) = x^4 + x + 1 = 10011$$

$$F(x) = x^4 M(x) + R(x) = 100111010000 + 1111 = 100111011111$$

2.3 Protocolli per il controllo di flusso

2.3.1 Stop-and-wait

È piuttosto elementare e si usa in canali simplex o half duplex. Il principio è il seguente: il mittente trasmette un frame e **aspetta** che chi è dall'altro capo

del canale invii una **conferma di ricezione**, detta **ACK**, *Acknowledgement*. Un chiaro svantaggio di questo protocollo è l'**elevato tempo di attesa** tra le trasmissioni dei frame: Stop-and-wait **non funziona bene in presenza di roundtrip delay elevato**.

Si possono verificare due errori:

Il frame non arriva a destinazione Il mittente si trova bloccato nell'**aspettare l'ACK, che non arriverà mai**, rendendo necessaria la presenza di un **timeout** dopo il quale il frame venga inviato nuovamente.

L'ACK non arriva al mittente Il destinatario riceve il frame ma al mittente non arriva la conferma: dopo il timeout descritto precedentemente i dati vengono ritrasmessi. Tuttavia, in questo modo **il destinatario riceve due volte lo stesso frame**. La soluzione consiste nel dotare i frame di un campo che lo etichetti con un numero; basta anche un solo bit, per indicare il precedente dal successivo.

2.3.2 I protocolli sliding window

Ogni partecipante alla conversazione deve tener sotto controllo **due 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.

Quando il mittente **invia un frame, resta nella finestra** finché non viene ricevuto il corrispondente ACK. Dopodiché la finestra verrà aggiornata. Quando il destinatario **riceve il frame**, controlla che il numero sia uguale a quello che si aspettava e ne invia l'ACK. Se quest'ultimo contiene il numero che la sorgente si aspettava, prosegue nella trasmissione inviando un nuovo frame. Altrimenti, invia nuovamente quello segnato nel buffer.

Si possono **inviare più frame contemporaneamente** prima di entrare in attesa (pipelining). Il destinatario aggiorna la finestra non appena riceva il frame e invia l'ACK. In caso di pipelining, i protocolli **go back n** e **selective repeat** permettono di gestire i problemi dovuti a questo tipo di trasmissione.

Go back n

E' il protocollo a finestra scorrevole più semplice. La finestra di chi riceve ha ampiezza 1 mentre quella di chi trasmette ha ampiezza N. Vengono inviati fino a N pacchetti alla volta con un'etichetta che indica lo slot della finestra, anche se il ricevente li gestisce comunque uno alla volta.

La destinazione, una volta accortasi che un frame è corrotto, **scarta a prescindere tutti i frame successivi** (per numero di sequenza) già ricevuti. Per i frame scartati non invia ACK, ma aspetta che scadano i timeout nel mittente, il quale provvederà a trasmetterli nuovamente.

Go back n **funziona bene quando il roundtrip delay è elevato e il canale è affidabile**. Si noti che il mittente deve essere in grado di gestire N timer per rispedire i pacchetti e avere un buffer capiente in cui tenerne una copia da ritrasmettere.

Selective repeat

Evoluzione del Go back n, nella quale anche il ricevente ha un buffer per contenere più frame contemporaneamente. Il destinatario conserva tutti i frame validi e li salva in un buffer. La sorgente continua ritrasmettere i frame per i quali, prima della scadenza del proprio timeout, non ha ricevuto l'ACK. Se la destinazione riceve un frame corrotto, invia un **NACK** (*Not ACK*) per sollecitare la ritrasmissione prima della scadenza del timeout da parte del mittente.

Si può inoltre inviare un NAK per indicare al mittente che un frame potrebbe essere stato perso e diminuire ulteriormente lo spreco di tempo (Il timer del NAK deve essere minore del timer di rinvio). Questo protocollo **sfrutta il più possibile il canale**, però richiede maggiori risorse dato che è necessario gestire un timer e una parte di buffer per ogni slot aperto della finestra. Un **problema** di questo protocollo (delle sliding windows in generale) è che l'**apertura massima della finestra deve essere al più uguale alla metà delle etichette disponibili per evitare la perdita di sincronizzazione**.

Piggybacking

È una tecnica che consiste nello **sfruttare un messaggio del destinatario al mittente come *passaggio* per il messaggio di conferma** ACK. Il campo ACK è posto nell'header del frame. Chiaramente, il piggybacking dell'ACK si usa solo se vi è un messaggio del destinatario al mittente nell'immediata possibilità d'inviare la conferma, altrimenti si invia l'ACK separatamente. Di conseguenza questa tecnica funzione bene quando la comunicazione tra le due parti è bilanciata.

2.4 PPP

Il protocollo PPP, *Point to Point Protocol*, viene utilizzato per gestire la configurazione della rilevazione d'errore di una linea, supportare molteplici protocolli, permettere l'autenticazione e molto altro. Provvisto di numerose opzioni, il protocollo PPP ha le seguenti tre caratteristiche principali:

- un metodo di framing che permette di delimitare in modo non ambiguo la fine di un frame e l'inizio del successivo. Il formato del frame permette di gestire anche la rilevazione di errori;

- un protocollo per gestire la connessione, il test della linea, negoziare le opzioni di collegamento e gestire la disconnessione in modo pulito quando la linea non serve più. Questo protocollo è chiamato **LCP**, *Link Control Protocol*;

- una modalità per negoziare le opzioni relative al livello di rete, in modo indipendente dall'implementazione di tale livello che verrà usata per la comunicazione. Il metodo scelto avrà un diverso NCP (*Network Control Protocol*), un protocollo di controllo della rete, per ogni livello di rete supportato.

Questo protocollo viene largamente utilizzato nelle connessioni DSL (per il collegamento utente-provider) e GPRS.

Nome	Numero di bytes	Descrizione
Flag	1	indica l'inizio o la fine del frame
Address	1	indirizzo broadcast
Control	1	byte di controllo
Protocol	2	indica il protocollo del campo data
Data	variabile (da 0 a 1500)	campo di dati
FCS	2 (o 4)	somma di correzione

Figura 2.1: Frame PPP

Il campo più importante di un frame PPP è il campo **Protocol** che specifica il tipo di protocollo PPP in uso: se il primo bit è a 1 si sta usando LCP mentre se è a 0 NCP. Ci sono 11 tipi di frame LCP, 4 di configurazione 2 di terminazione, 2 di rifiuto, 2 di echo e 1 di test.

Configurazione: configure-request, configure-Ack, configure Nak, e configure-reject, vengono usati per stabilire e configurare la connessione, si può scegliere la lunghezza del campo dati, che livello di error-detection usare e se trasmettere o meno i campi Address e Control.

Terminazione: terminate-request e terminate-ack.

Rifiuto: code-reject (richiesta sconosciuta) e protocol-reject (protocollo richiesto non supportato).

Echo: echo-request e echo-reply, per il test della qualità della rete.

Test: discard-request

I frame PPP poi possono essere incapsulati in frame Ethernet (PPPoE) o ATM (PPPoA).

Capitolo 3

Data Link: MAC

3.1 Protocolli ad accesso multiplo

3.1.1 ALOHA

ALOHA puro

Il protocollo ALOHA è un sistema a contesa ideato per trasmettere via radio i segnali in broadcast. Ogni stazione trasmette ogniqualvolta ha bisogno di farlo, senza verificare se il canale è libero.

Dopo aver trasmesso, resta in ascolto per notare se si sono verificate collisioni e, in caso, riprova dopo un tempo casuale. Solitamente si attua utilizzando un meccanismo di **back-off**, secondo il quale la ritrasmissione avviene dopo un ritardo selezionato casualmente compreso tra 0 e $(K-1)T$, dove T è il tempo di trasmissione del frame e K può eventualmente dipendere dal numero di collisioni già avvenute.

Si possono trasmettere con successo Ge^{2G} frame per unità di tempo, dove G indica il numero di trasmissioni medie per unità di tempo, raggiungendo un **utilizzo della banda pari al 18,4 percento**.

Slotted ALOHA

Nel protocollo Slotted ALOHA, il tempo viene diviso in **intervalli discreti detti slot**, cadenzati da una stazione principale. Tutte le stazioni possono trasmettere solamente all'inizio dello slot, dimezzando così le possibilità di collisione e raddoppiando l'utilizzo della banda.

In altre parole, In ALOHA un frame poteva collidere anche solo parzialmente, con un frame che era già stato inviato, o viceversa, con un frame appena inviato da un'altra stazione. Di conseguenza il periodo in cui il frame poteva collidere era 2 volte il suo tempo di vita. Con la versione slotted il periodo critico viene dimezzato, visto che **una collisione può verificarsi solo quando due frame vengono inviati contemporaneamente nello stesso slot**.

3.1.2 Protocolli Carrier Sense

Sono protocolli multi-accesso in cui ogni stazione è consapevole di condividere il canale con altre stazioni.

CSMA 1-persistente

Quando una stazione deve trasmettere, prima di farlo, **ascolta il canale per sapere se è già occupato**. In questo caso la stazione resta in attesa finché non si libera e poi trasmette. Se è libero trasmette i propri dati *senza esitazione*. Se avviene una collisione la stazione aspetta un tempo casuale e poi ricontrolla il canale. Il protocollo è chiamato *1-persistente* perché la stazione trasmette con probabilità 1 quando trova il canale libero.

Con questo protocollo si ottiene un **data rate superiore al 50 percento** anche se ci sono due problemi critici:

1. non viene considerato il tempo di propagazione del segnale: se due stazioni si trovano lontane tra loro e una inizia a trasmettere, prima che l'altra veda il canale occupato passa del tempo durante il quale vede il canale ancora libero.

2. se due stazioni che vogliono trasmettere trovano il canale occupato, appena si libera trasmettono contemporaneamente, generando una collisione (corretto con le versioni p-persistente e non persistente).

CSMA non persistente

In questo protocollo **si tenta consapevolmente di essere meno ingordi**. Prima di trasmettere, ogni stazione controlla il canale: se nessun altro sta trasmettendo inizi a inviare i dati ma, se il canale è occupato, **la stazione non esegue un controllo continuo per impossessarsene subito alla fine della trasmissione, bensì attende per un intervallo casuale** prima di ripetere l'algoritmo. Di conseguenza, questo meccanismo permette di utilizzare meglio il canale, ma allunga i ritardi rispetto a CSMA 1-persistente.

CSMA p-persistente

Si applica ai canali divisi in intervalli temporali. Quando è pronta a trasmettere, ogni stazione controlla il canale. Se lo trova libero, trasmette subito con probabilità p e rimanda la trasmissione all'intervallo successivo con probabilità $q = 1 - p$. Il processo si ripete fino a quando il frame non è stato trasmesso o un'altra stazione ha iniziato a trasmettere. In quest'ultimo caso, la stazione sfortunata si comporterebbe come se ci fosse stata una collisione. Se la stazione inizialmente avesse trovato il canale occupato, avrebbe atteso fino all'intervallo successivo per poi applicare l'algoritmo descritto in precedenza.

CSMA con rilevamento delle collisioni

Questo protocollo, CSMA/CD (*CSMA with collision detection)*, è alla base delle classiche LAN Ethernet. Le stazioni **ascoltano il canale durante la trasmissione e, se rilevano una collisione, interrompono bruscamente l'operazione** dal momento che i frame inviati sono comunque rovinati. Questa strategia permette di **risparmiare tempo e banda**.

È importante capire che l'individuazione delle collisioni è un processo analogico. L'hardware della stazione deve ascoltare il canale durante la trasmissione; se il segnale letto è diverso da quello inviato, sa che sta avvenendo una collisione. Se una stazione rivela una collisione, interrompe la trasmissione, aspetta per una quantità di tempo casuale e quindi prova di nuovo.

3.1.3 Protocolli senza collisione

Basic Bitmap

Ogni periodo di contesa, *contention period*, è diviso in N intervalli (*slot*). Se la stazione 0 deve inviare un frame, trasmette un bit 1 durante l'intervallo 0. In generale, quindi, **la stazione i-esima, quando ha un frame da inviare, invia un 1 nell'intervallo i-esimo**. Una volta trascorsi gli N Intervalli, tutti sanno chi deve trasmettere e in ordine iniziano a farlo. Una volta finito il giro di trasmissioni, si ricomincia da capo.

Nonostante la mancanza intrinseca di collisioni, questo protocollo presenta comunque dei difetti. **Se ci sono troppe stazioni il contention period diventa troppo lungo** causando uno spreco di banda, specialmente se solo poche stazioni vogliono trasmettere. Inoltre, se una stazione ha bisogno di trasmettere, però il suo slot per prenotarsi è già passato, deve rimanere inattiva fino al prossimo giro di trasmissioni.

Binary countdown

Ogni stazione intenzionata a utilizzare il canale **comunica in broadcast il proprio indirizzo sotto forma di stringa binaria**, partendo dal bit più significativo. In questo modo nel canale è presente l'OR degli indirizzi di tutte le stazioni che vogliono trasmettere. Successivamente, per decidere chi può trasmettere, **si inizia a guardare la stringa presente nel canale a partire dal bit più significativo**: se è impostato a 1, vengono scartate tutte le stazioni con quel bit pari a 0 e si prosegue analizzando il secondo bit della stringa. **Vince la stazione che rimane, ossia quella che ha l'indirizzo più alto**. Esempio: se le stazioni 0010, 0100, 1001 e 1010 vogliono trasmettere, vincerà 1010. Al termine della trasmissione, rincomincia un nuovo turno.

È abbastanza evidente la proprietà per cui **le stazione con il numero più alto hanno una priorità maggiore** rispetto alle stazioni con il numero più basso. Per evitarlo, si possono usare due strategie:

- una strategia in stile ALOHA, nella quale una stazione **dopo aver trasmesso aspetta un tempo casuale** prima di ritrasmettere. Comporta però un difetto: il canale potrebbe rimanere inattivo se ci sono poche stazioni che trasmettono.
- quando una stazione ha ottenuto il diritto di trasmettere **si porta a priorità (indirizzo) 0**, mentre tutte le altre che non hanno trasmesso aumentano la propria priorità di 1.

Con questo metodo l'efficienza del canale è pari a:

$$\frac{d}{d + \log_2 N}$$

ma se il formato del frame viene scelto con cura in modo che l'indirizzo del mittente costituisca il primo campo del frame, si recuperano anche questi $\log_2 N$ bit e l'efficienza raggiunge il 100%.

3.1.4 Protocolli a contesa limitata

Adaptive Tree Walk Protocol

Protocollo di trasmissione che combina la strategia Collision Detection con quella Collision Avoidance. Le stazioni vengono viste come le foglie di un albero binario. Ad ogni slot di trasmissione tutti quelle che devono trasmettere trasmettono. Se si verifica una collisione vengono creati due slot, uno per il sottoalbero destro e uno per quello sinistro. Se in uno dei due slot si verifica una collisione si ripete lo stesso procedimento finché non tutti non riescono a trasmettere.

Con questa tecnica si riescono ad individuare le stazioni che vogliono trasmettere con al più $\log_2 N$ tentativi. Nel lato pratico questa tecnica può essere ancora migliorata: è possibile infatti tenere traccia delle stazioni che hanno trasmesso recentemente ed iniziare il ad esaminare l'albero a partire da una certa profondità. Se ci sono A stazioni attive si può iniziare ad esaminare l'albero dai nodi di profondità $P = \log_2 A$.

3.2 Protocolli per LAN Wireless

3.2.1 Stazione Nascosta

Si verifica quando **una stazione non riesce a rilevare, a causa della distanza, altri concorrenti per il mezzo trasmissivo**. Il problema della stazione nascosta si verifica in questa situazione:

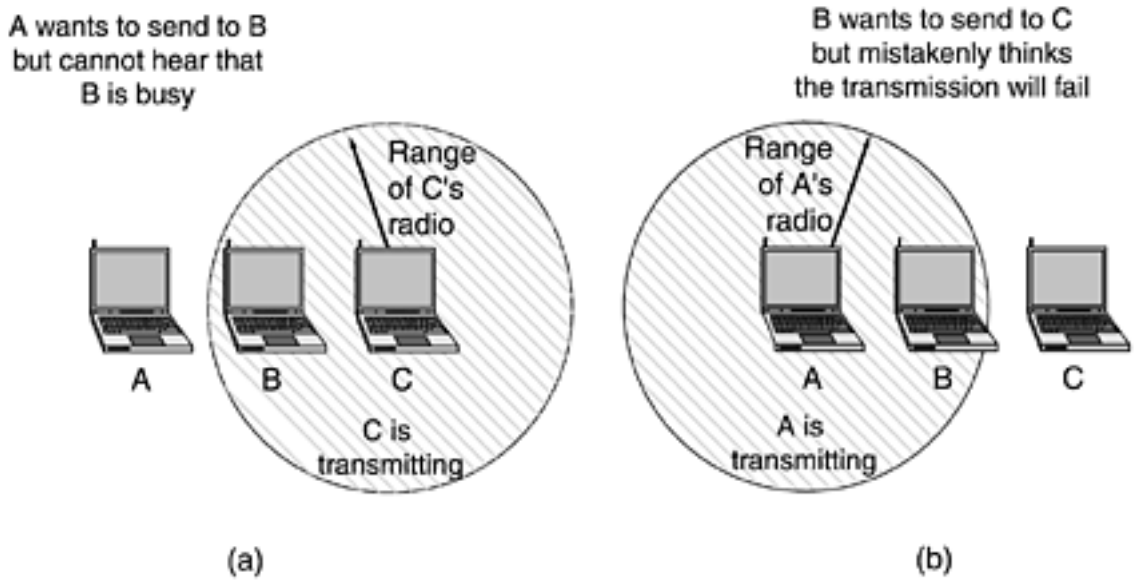

Figura 3.1: Esempio stazioni Wireless

1. A sta trasmettendo dei frame a B;
2. anche C vuole trasmettere a B;
3. C controlla il canale, non sente la trasmissione di A e quindi inizia a trasmettere a B.

Le due trasmissioni interferiscono tra loro portando alla perdita di entrambi i frame.

3.2.2 Stazione Esposta

Si verifica quando una stazione **non trasmette perché sente il canale occupato, anche se la sua trasmissione non interferirebbe** con la trasmissione già in corso.

1. B sta inviando dei frame ad A
2. C vuole trasmettere verso D, quindi ascolta il canale, lo trova occupato e non trasmette, anche se una trasmissione da C verso D non interferirebbe con la trasmissione di B.

3.2.3 MACA e MACAW

MACA è un protocollo di accesso al canale che si basa sull'uso di **due particolari frame**: RTS (*Request To Send*) e CTS (*Clear To Send*). Entrambi i frame contengono la dimensione del frame dati che sarà scambiato, in modo

che le altre stazioni possano stimare la durata della trasmissione. Quando una **stazione A vuole trasmettere ad una stazione B**, invia il frame RTS. Se B accetta la trasmissione, invia ad A il frame CTS, dopodiché la trasmissione può iniziare.

Le stazioni che sentono solo il frame RTS devono rimanere in silenzio per consentire ad A di sentire il CTS. Le stazioni che sentono solo il CTS o entrambi restano in silenzio per evitare interferenze.

La versione successiva (MACAW) introduce il meccanismo di ACK per il frame dati e un altro frame di controllo chiamato DS contenente la **dimensione del frame dati** che si sta per trasmettere. In questo modo se delle stazioni volevano trasmettere ad una stazione che sta già trasmettendo sanno tra quanto la stazione sarà libera.

3.3 Ethernet

3.3.1 Codifica di Manchester

Questo tipo di codifica nasce dalla **necessità di dover determinare senza ambiguità il punto iniziale, centrale e finale di ogni bit**, senza impulsi esterni. Si divide **il periodo di un bit in due intervalli uguali**: se si deve inviare 1 si tiene un **livello di tensione alto nel primo intervallo e basso nel secondo**. Viceversa se si deve inviare uno 0.

In questo modo **è più semplice la sincronizzazione** tra mittente e destinatario anche con lunghe sequenze di 0 o 1. Di contro, la **banda viene dimezzata**. Esiste una variante di questa codifica detta differenziale, che si basa sul cambio di transazione tra intervalli. Se è uno 0 avviene il cambio di transizione, altrimenti no. È più immune al rumore anche se più complessa.

3.3.2 Binary Exponential Backoff

È un algoritmo, usato in CSMA/CD e quindi in Ethernet, **per stabilire il tempo di attesa casuale una volta riscontrata una collisione**.

1. Si calcola T_p, il tempo di propagazione di andata e ritorno del frame nel caso peggiore (*slot time*);
2. si divide il tempo in intervalli P che durano T_p ciascuno;
3. stabilito un contatore $i = 1$, finché la trasmissione non avviene con successo:
   (a) si sceglie casualmente un numero N tale che $0 \leq N < 2^i$;
   (b) si attendono $N \times P$ intervalli prima di ritentare la trasmissione;
   (c) si incrementa i.

Si noti che N **cresce esponenzialmente** con il numero di collisioni avvenute: questo garantisce che il tempo di attesa sia minimo se ci sono poche a stazione a collidere e che invece possa essere molto alto nel caso ci siano molte più stazioni. Esiste anche una versione ***truncated***, in cui si fissa un **valore massimo per il contatore** i dopo il quale, in caso di ritrasmissione fallita, l'operazione viene annullata e ai livelli più alti viene restituito un errore. Ad esempio, se i può valere al massimo 10, il massimo tempo di attesa possibile equivale a 1023 slot time.

3.3.3 Frame Ethernet

Vengono di seguito illustrati i campi dati di un frame Ethernet secondo lo standard ***DIX***, leggermente diverso dallo standard IEEE 802.3.

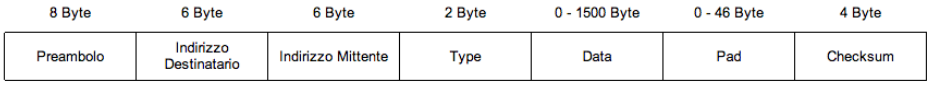

Figura 3.2: Frame Ethernet

Preambolo Composto da 8 byte, ognuno dei quali contiene la sequenza di bit 10101010. Permette al clock del ricevente di sincronizzarsi con quello del trasmittente. In IEEE 802.3 i due bit meno significativi dell'ultimo byte formano un campo a sé stante: SoF, *Start Of Frame delimiter*, che avvisa il ricevente dell'immediata fine del campo Preambolo.

Indirizzo Destinatario/Mittente Indica gli indirizzi MAC di mittente e destinatario. Nel campo destinatario, se il bit più significativo è impostato a 1 significa che la trasmissione è diretta a un gruppo di stazioni (multicast). Se tutti i bit sono impostati 1, la trasmissione è di tipo broadcast.

Type Banalmente, indica al ricevente cosa fare con il frame arrivato, cioè a quale livello di rete passarlo. Nello standard IEEE 802.3 questo campo è detto *Length* e contiene la lunghezza del frame e per gestire il passaggio a livello di rete è stato creato un livello intermedio chiamato LLC (*Logical Link Control*).

Data Contiene i dati che devono essere trasmessi.

Pad E' un campo di riempimento usato per garantire una dimensione minima di 64 byte al frame. In questo modo la trasmissione del frame richiede almeno 2 volte il tempo di roundtrip, garantendo che il mittente possa accorgersi di una eventuale collisione.

Checksum CRC a 32 bit per il controllo su un eventuale errore di trasmissione.

3.3.4 Storia e versioni di Ethernet

10 Base 5 È la prima versione, veniva usato un grosso cavo coassiale che permetteva il collegamento di un transceiver ogni 2,5 metri, quest'ultimo si occupa del carrier detection e del collision detection. Il numero 10 indica la velocità (10Mbps) e 5 la lunghezza massima dei segmenti (500 metri).

10 Base 2 Viene usato un tipo di cavo coassiale più sottile, le giunzioni vengono effettuate mediante connettori BNC e il transceiver viene spostato all'interno del pc. E' più economico e affidabile però la lunghezza massima di un segmento cala (200 metri) così come il numero di computer per segmento (30).

10 Base T Il cavo condiviso viene sostituito da hub collegati mediante cavi UTP dedicati. I costi diminuiscono, l'affidabilità aumenta così come il numero di computer per segmento (1024). Diminuisce però la lunghezza massima di un segmento (100 metri).

10 Base F Si basa sulla fibra ottica, permette segmenti fino a 2km.

Fast Ethernet (100 Base T4-TX-FX) Il clock viene aumentato di un fattore 10 però, dato che con Ethernet $Efficienza = \frac{1}{Banda \cdot Lunghezza}$, è stato necessario utilizzare degli switch al posto degli hub. La versione T4 usa 4 cavi UTP (100 metri), la TX usa un cavo UTP5 (sempre 100 metri) mentre FX usa la fibra ottica e arriva fino a 2km

Gigabit Ethernet Oltre ad incrementare la velocità, introduce anche una modalità point-to-point. Per ottenere questa velocità viene perso il carrier sense. I frame vengono trasmessi a blocchi (frame bursting), riuscendo a tenere una copertura di 200 metri. Per evitare problemi di sincronizzazione, i dati vengono codificati in modo da evitare sequenze di 0 o 1 troppo lunghe. Inoltre, viene usato l'encoding 8B/10B in modo che non ci siano mai 4 bit consecutivi identici in una word e che non ci siano mai più di sei 0 o sei 1 nella stessa word. Viene introdotto anche un frame speciale per mandare in pausa la trasmissione, dato che se uno switch si blocca anche per 1ms si possono accumulare un elevato numero di frame.

Capitolo 4

Network

Il livello di rete si occupa del trasporto dei pacchetti lungo tutto il percorso, dall'origine alla destinazione finale, che per essere raggiunta può richiedere molti salti attraverso router intermedi lungo il percorso. Questa funzione è chiaramente distinta da quella del livello data link, che ha il più modesto obiettivo di spostare i frame da un capo all'altro del cavo. Nel scegliere il percorso che devono seguire i vari pacchetti, deve anche evitare di sovraccaricare alcune linee di comunicazione, lasciando altre completamente libere. Deve poi occuparsi dei problemi che nascono quando la sorgente e la destinazione si trovano in reti diverse.

4.1 Tecniche di routing

4.1.1 Flooding

È un algoritmo statico che funziona in questo modo: **ogni pacchetto viene ritrasmesso su tutte le linee tranne quella da cui è arrivato**. Tuttavia, aumenta pericolosamente il carico di rete, rischiando di congestionarla. Per limitare ciò si può usare il metodo di **hop counting**, che consiste nel dare ad ogni pacchetto un tempo di vita, basato su un numero massimo di stazioni attraversabili (hop, salto), allo scadere del quale il pacchetto viene distrutto e la trasmissione interrotta. Un'alternativa consiste nel **tener traccia dei pacchetti già arrivati** per evitare di trasmetterli nuovamente ed inutilmente, richiedendo però molta memoria.

Questa trasmissione broadcast porta con sé anche diversi vantaggi, primo tra tutti la **robustezza ai cambiamenti della rete**. Inoltre, è il più semplice da implementare, viene ovviamente scelta sempre la via migliore e non sono necessarie modifiche per passare da trasmissioni point-to-point a trasmissioni broadcast.

Esiste una variante del flooding, detta **flooding selettivo**, in cui ogni router non trasmette in tutte le direzioni, ma solo in quelle che approssimativamente vanno nella direzione giusta.

4.1.2 Distance Vector Routing

Si tratta di un algoritmo dinamico, perché tiene conto dello stato della rete. Opera in modo che ogni router conservi una tabella che definisce la **migliore distanza conosciuta per ogni destinazione e il collegamento che conduce a tale destinazione**. Queste tabelle sono aggiornate scambiando informazioni con i router vicini. Alla fine del processo ogni router conosce il collegamento migliore per raggiungere **qualsiasi destinazione**.

Quando un router si connette alla rete, richiede a tutti i vicini la loro tabella; tenendo conto del tempo necessario per ricevere le tabelle e delle informazioni contenute in esse, il router è in grado di costruirsi la propria. Ogni T msec i router aggiornano le loro tabelle allo stesso modo.

Ogni voce in queste tabelle contiene due campi: la linea di trasmissione preferita e la stima del tempo o della distanza associata a quella destinazione. Questa tecnica non tiene conto della capacità di banda, problema risolvibile usando come metrica la distanza pesata. Inoltre, quando un nodo ha un problema di sovraccarico o viene spento si verifica il count-to-infinity, ossia la lenta convergenza al verificarsi di *cattive notizie*.

4.1.3 Link state routing

L'idea di base di questo algoritmo può essere riassunta in cinque passaggi. Ogni router deve:

1. **scoprire i propri vicini** e relativi indirizzi di rete: appena acceso, il router invia uno speciale pacchetto *HELLO* su ogni linea punto a punto. Il router dell'altro capo della linea deve rispondere fornendo il proprio nome. Tutti i nomi devono essere univoci;

2. **misurare la distanza** o la metrica di costo di ogni vicino: s'invia attraverso la linea uno speciale pacchetto *ECHO*, al quale l'altra parte deve rispondere immediatamente. Così il router ottiene una stima del ritardo di trasmissione con il vicino, che può essere usato nella funzione per calcolare il costo;

3. costruire un **pacchetto contenente tutte le informazioni** raccolte: ogni pacchetto inizia con l'identità del trasmittente, seguita da un numero di sequenza, dall'età e da una lista dei vicini. Per ogni vicino è riportato il ritardo misurato.

4. **inviare tale pacchetto a tutti gli altri router** e ricevere da loro i pacchetti: in ogni pacchetto è inserito un numero di sequenza, incrementato per ogni nuovo pacchetto inviato. I router tengno traccia di tutte le coppie (router sorgente, sequenza) rilevate. Quando arriva un nuovo pacchetto link state, il router confronta i dati con quelli che ha già analizzato. Se è nuovo, viene inoltrato tramite **flooding**, altrimenti

viene scartato perché obsoleto. Il campo riguardante l'età del pacchetto viene decrementato da ogni router durante il processo di flooding, per garantire che nessun pacchetto possa perdersi nella rete e durare un periodo indefinito di tempo. Se l'età raggiunge lo 0, il pacchetto viene scartato.

5. **elaborare il percorso più breve verso tutti gli altri** router: viene usato l'algoritmo di Dijkstra per costruire il percorso più breve verso tutte le possibili destinazioni. I risultati di questo algoritmo indicano al router che linea usare per raggiungere qualsiasi destinazione.

Link State può gestire reti composte da un gran numero di nodi, sa convergere rapidamente generando raramente cammini ciclici. Tuttavia, permettere ad ogni nodo di avere una mappa della rete rende il meccanismo complesso e costoso dal punto di vista della memoria e delle capacità di elaborazione dei nodi.

4.1.4 Routing gerarchico

A volte la rete è talmente estesa che avere informazioni riguardo a tutti i router diventa impossibile. In questi casi la rete viene divisa in regioni: ogni router conosce solo i router della propria regione. Così facendo, quando si connettono diverse sottoreti, ogni regione vede un'altra come un semplice nodo.

4.1.5 Reverse Path Fowarding

Quando riceve un pacchetto broadcast, il router verifica se il pacchetto è giunto attraverso la linea che normalmente è utilizzata per inviare i pacchetti alla sorgente di trasmissione broadcast. In caso affermativo, c'è una forte probabilità che il pacchetto broadcast stesso abbia seguito il **percorso migliore dal router** e che perciò sia la **prima copia arrivata** al router. In questo caso il router inoltra le copie del pacchetto su tutte le linee, esclusa quella di input. Se al contrario il pacchetto broadcast è giunto da una linea diversa da quella preferita per raggiungere la sorgente, il pacchetto è scartato, in quanto è probabile che si tratti di un duplicato.

4.2 Qualità della rete

4.2.1 Parametri del QoS

Il QoS rappresenta la qualità del servizio basandosi su una serie di parametri. Nelle reti i più importanti sono:

Reliability (Affidabilità) dato dal tasso d'errore della rete

Bandwidth velocità della rete

Delay il tempo medio impiegato da un pacchetto per arrivare a destinazione

Jitter grado di variazione del tempo di arrivo dei pacchetti

L'importanza di questi parametri varia in base all'utilizzo della rete, per esempio quando c'è del traffico voce è importante avere delay e jitter minimi piuttosto che un'elevata affidabilità mentre quando si vuole trasferire un file è più importante avere una rete affidabile e veloce. Un fattore critico per avere un buon QoS è la gestione delle congestioni, che in un ambito connection-oriented è più semplice (Virtual Circuit [1]) rispetto all'ambito connection-less (Datagram).

4.2.2 Congestioni in reti di datagrammi

Ci sono varie strategie per gestire la congestione che possono essere combinate tra loro, le principali sono

Choke *trattato a parte*

Load Shedding *trattato a parte*

Random Early Detection *trattato a parte*

Bucket *trattato a parte*

Buffering non è propriamente un sistema di gestione della congestione dato che il ricevente si limita a memorizzare più pacchetti prima di elaborarli riducendo così l'influenza del jitter. (Es: YouTube)

Traffic Shaping chi trasmette e chi riceve si accordano sul rate di trasmissione (*Service Level Agreement*) in modo da evitare di sovraccaricare la rete. La velocità accordata può anche variare dinamicamente in base al traffico.

4.2.3 Choke packet

Quando un router si accorge che c'è congestione invia un pacchetto speciale (choke packet) a chi sta trasmettendo i dati. L'**host sorgente, una volta ricevuto il chocke packet, deve ridurre il traffico inviato** alla destinazione di un certo tasso percentuale, solitamente il 50%. Per uscire dallo stato di choke si usa il fading: se per un certo periodo di tempo non si ricevono altri choke packet si riprende a trasmettere normalmente. Inoltre, dopo aver ricevuto un choke per un po' di tempo si ignorano successive richieste di

[1] E' sufficiente fare admission control, si cerca di evitare i nodi congestionati e se non è possibile si proibisce la creazione di nuovi canali

choke, perché più router potrebbero aver mandato il choke allo stesso host. Esiste anche un'altra versione detta **hop-by-hop choke** nella quale anche i router attraversati dal choke packet dimezzano il loro rate di trasmissione verso la linea dalla quale è arrivato.

4.2.4 Load Shedding

Quando un **router è troppo carico, scarta alcuni pacchetti**. La scelta può essere fatta in due modi:

Wine vengono cancellati i pacchetti più nuovi. Adottato per esempio nel trasferimento di file e nel remote login.

Milk vengono cancellati i pacchetti più vecchi. Adottato per esempio nelle trasmissioni multimediali.

Per migliorare l'algoritmo di decisione, i pacchetti possono essere contrassegnati da un livello di priorità.

4.2.5 Random Early Detection

L'idea di base è simile a quella del load shedding, con la sola differenza che al posto di aspettare che avvenga la congestione, lo **scarto dei pacchetti viene fatto 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 avvisata della cancellazione del pacchetto per evitare traffico inutile. Semplicemente, la sorgente non vedendo l'ACK prende provvedimenti.

4.2.6 Leaky e Token Bucket

Sono dei filtri che possono essere implementati sia a livello SW sia HW e consentono di limitare il data rate massimo del canale e di contenere i burst. Funzionano mediante una coda FIFO di capienza limitata: quando la coda è piena i nuovi pacchetti che non riescono ad entrare vengono distrutti. Le due versioni differiscono per il modo in cui i pacchetti escono dalla coda.

Leaky i pacchetti escono con un data rate fisso. Questo può anche essere uno svantaggio perché a volte basterebbe un data rate un po' più sostenuto per gestire meglio il traffico.

Token ad ogni ciclo di clock vengono generati dei token: affinché un pacchetto possa passare deve essere disponibile un token. iIn caso di scarso traffico i token in eccesso vengono accumulati in modo da gestire più efficientemente i burst.

4.3 IP

Internet Protocol (IP) è un protocollo di rete appartenente allo stack di protocolli TCP/IP su cui è basato il funzionamento della rete Internet. È classificato al livello di rete nel modello ISO/OSI ed è **nato appositamente per interconnettere reti eterogenee** per tecnologia, prestazioni e gestione. Pertanto, può essere implementato sopra altri protocolli di livello data link, come Ethernet e ATM.

È un **protocollo a pacchetto senza connessione di tipo best effort**, cerca cioè fare *del suo meglio* senza però garantire alcuna forma di affidabilità della comunicazione in termini di controllo di errori, controllo di flusso e controllo di congestione a cui quindi dovranno supplire i protocolli di trasporto di livello superiore, come ad esempio TCP. La versione correntemente usata di IP è detta IPv4, per distinguerla dalla più recente IPv6, nata dall'esigenza di gestire meglio il crescente numero di computer (host) connessi ad Internet.

Il principale compito di IP è l'**indirizzamento e l'instradamento** (commutazione) **tra sottoreti eterogenee**, che a livello locale utilizzano invece un indirizzamento proprio, tipicamente basato sull'indirizzo fisico o indirizzo MAC, e protocolli di livello datalink del modello ISO-OSI (Ethernet, Token Ring, Token bus, ecc). Per far ciò è necessario assegnare un nuovo modello comune a cui tutte le sottoreti devono sottostare per poter comunicare ed interoperare tra loro: tale modello è costituito proprio dal protocollo IP e dal suo indirizzamento. Questo infatti comporta:

- l'**assegnazione a ciascun terminale** che ne fa richiesta (al momento della connessione) **di un nuovo indirizzo**, univocamente associato all'indirizzo MAC locale, **detto indirizzo IP**. Può essere assegnato tramite il protocollo DHCP;

- la definizione delle **modalità o procedure tese ad individuare il percorso di rete** per interconnettere due qualunque sottoreti durante una comunicazione tra host sorgente di una certa sottorete e host destinatario di un'altra sottorete, cui l'indirizzo IP appartiene. La conoscenza di questo percorso di rete comporta a sua volta l'assegnazione e la conoscenza dell'indirizzo IP a ciascun commutatore (router) che collega la rete dell'host emittente con quella dell'host destinatario, cioè quindi la conoscenza della sequenza di tutti i router di tutte le sottoreti da attraversare.

Come già detto, all'interno di una rete IP, ad ogni interfaccia viene assegnato un nuovo indirizzo univoco. Operare un indirizzamento al livello di rete tramite i soli indirizzi MAC, pur essendo questi univoci per ciascun terminale host, non sarebbe stato possibile perché essi **non danno vita ad un indirizzamento gerarchico**, cioè non sono raggruppabili in sottoreti

con lo stesso prefisso identificativo come invece lo sono gli indirizzi IP. Esisterebbero quindi notevoli problemi di scalabilità nell'implementare tabelle di instradamento indicizzate. La corrispondenza tra indirizzo IP e MAC è gestita tramite il protocollo ARP.

4.3.1 Header IPv4

+	Bits 0–3	4–7	8–15	16–18	19–31
0	Version	Internet Header length	Type of Service (now DiffServ and ECN)	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				
160	Options (optional)				
160 o 192+	Data				

Figura 4.1: Header IPv4

Version indica la versione del protocollo IP, fisso a 4.

IHL indica la lunghezza dell'header, espressa in word da 4 byte.

Type of service serve per settare il QoS, amche se il più delle volte è ignorato.

Total Length lunghezza totale del pacchetto (header+dati) espressa in byte.

Identification identifica i frammenti in cui è stato diviso un pacchetto.

Flags il primo bit è fisso a 0, il secondo,**DF**, se settato a 1 indica che il pacchetto non deve essere frammentato, il terzo,**MF**, se settato a 1 indica che il pacchetto è un frammento di un pacchetto più grande mentre se settato a 0 indica che era l'ultimo frammento.

Fragment Offeset indica l'offset di un frammento rispetto al pacchetto dati originale, espresso in word da 8 byte;

TTL rappresenta l'età massima del pacchetto.

Protocol indica il protocollo usato nel campo dati.

Header checksum campo per l'error detection relativo all'header, realizzato con somme in complemento a 1.

Indirizzi destinatario/mittente.

Options opzioni per l'uso specifico del protocollo, non molto usate.

4.3.2 Indirizzi IP

Gli indirizzi IPv4, come già detto, sono lunghi 32 bit e permettono di definire una gerarchia. Ogni indirizzo, infatti, è composto da una parte di rete di lunghezza variabile nei primi bit e di una parte per l'host negli ultimi. La parte di rete ha lo stesso valore per tutti gli host di una singola rete, come in una LAN Ethernet. Questo significa che una rete corrisponde a un blocco contiguo di indirizzi IP; tale blocco è chiamato **prefisso**.

Gli indirizzi di rete sono scritti in **notazione decimale puntata**. In questo formato ognuno dei 4 byte è rappresentato in forma decimale con un numero che varia tra 0 e 255. Ad esempio, l'indirizzo 10000000.11010000.0000 0010.10010111, in esadecimale 80.D0.02.97, diventa, nel formato decimale 128.208.2.151. **I prefissi sono scritti dando l'indirizzo IP più basso del blocco e la grandezza del blocco**. La **dimensione è determinata dal numero di bit nella porzione di rete**, il che significa che deve essere una potenza di due. Per convenzione viene scritta dopo il prefisso come una barra obliqua seguita dalla lunghezza in bit della parte dedicata alla rete. Nel nostro esempio, se il prefisso contiene 2^8 indirizzi IP e quindi 24 bit sono dedicati alla rete, è scritto come 128.208.0.0/24.

La **lunghezza del prefisso corrisponde a una maschera binaria di 1** nella parte di rete. quando viene scritta in questo modo è chiamata **subnet mask**. Si può effettuare un'operazione di AND logico con l'indirizzo IP per estrarre la porzione di rete. Continuando con l'esempio, la subnet mask è 255.255.255.0.

4.3.3 Subnetting

Subnetting è una procedura che consiste nel **suddividere un blocco di indirizzi in più parti per usarle internamente come reti multiple**, presentandole tuttavia all'esterno come una singola rete. La suddivisione non deve per forza essere uniforme, ma ogni parte deve essere allineata in modo che tutti i bit nella parte più bassa, dedicata all'host, possano essere usati.

Quando arriva un pacchetto, il router guarda l'indirizzo di destinazione e, per capire a quale subnet appartiene, effettua un AND logico dell'indirizzo di destinazione con la maschera di tutte le subnet. Poi controlla se il risultato corrisponde al prefisso.

4.3.4 CIDR

Il CIDR, *Classless Inter-Domain Routing*, è un sistema di **assegnazione degli IP a blocchi di taglia variabile**, introdotto nel 1993 per sostituire lo schema classful secondo il quale tutti gli indirizzi IP appartengono ad una specifica classe (A, B, C) a taglia fissa. Questo nuovo schema d'indirizzamento consente una **migliore gestione degli indirizzi**, che diventano

sempre più scarsi con il crescere di Internet, e migliora le prestazioni dell'instradamento grazie ad una più efficiente organizzazione delle tabelle di routing.

Per gestire l'indirizzamento i router usano delle *aggregate entries*, altrimenti le tabelle di indirizzamento sarebbero troppo grandi: se ci sono due o più indirizzi IP simili, il router li memorizza in una stessa entry della tabella dato che è ragionevole pensare che le relative macchine siano vicine. Quando il router deve inviare un pacchetto cerca nella sua tabella l'entry con il **prefisso comune più lungo**. Può capitare che un indirizzo IP venga aggregato in un entry in cui non c'entra niente, in questo caso il router deve essere in grado di gestire l'eccezione e creare un entry a parte per quell'indirizzo.

4.3.5 Descrizione dell'header IPv6

Figura 4.2: Header IPv6

Version: indica la versione del protocollo, fisso a 6.

Traffic Class: non ancora utilizzato, pensato per specificare il tipo di traffico (dati, trasmissioni real-time ecc.).

Flow Label: serve a specificare il flusso di appartenenza del pacchetto in modo che i pacchetti di uno stesso flusso vengano trattati allo stesso modo.

Payload Length: dimensione del campo dati, non viene considerato l'header.

Next Header: serve per specificare dove si trova, se presente, un ulteriore header contenente altre opzioni.

Hop Limit: numero massimo di Hop che il pacchetto può attraversare

Indirizzi: destinatario/mittente.

4.3.6 Principali differenze tra IPv4 e IPv6

- indirizzi più lunghi: 16 byte di IPv6 contro i 4 di IPv4. Questo si traduce nel **supporto a miliardi di host anche con un'allocazione di indirizzi altamente inefficiente**;
- **semplificazione** dell'intestazione. Ad esempio, si è deciso di togliere l'error detection per rendere la trasmissione più veloce. Dopotutto, era ridondante e rallentava l'instradamento dato che il decremento del campo TTL per ogni nodo attraversato rendeva necessario il ricalcolo del checksum;
- opzioni del protocollo: in IPv4 erano limitate a 40 byte mentre con IPv6, grazie al campo *Next Header*, sono potenzialmente illimitate. Lascia quindi **spazio ad evoluzioni future**;
- IPv6 fornisce un grado di sicurezza maggiore;
- IPv6 permette a un host di spostarsi senza cambiare il suo indirizzo;
- Il vecchio protocollo può coesistere con quello nuovo.

4.4 Altri protocolli

4.4.1 DHCP

DHCP, *Dynamic Host Configuration Protocol*, è un protocollo che permette l'assegnazione manuale o automatica dell'indirizzo IP. L'idea di base è semplice: esiste un server *speciale* che assegna gli indirizzi alle macchine che lo richiedono. Questo server può trovarsi sulla stessa LAN del richiedente.

Quando una macchina vuole ottenere un indirizzo IP, manda in broadcast un particolare pacchetto chiamato *DHCP DISCOVER*. Il gestore di rete centralizzato risponde alla macchina inviando un pacchetto contenente un indirizzo. Dato che nella stessa rete ci possono essere più gestori DHCP, l'host richiedente deve confermare al gestore che ha accettato l'IP proposto, il quale dovrà chiudere l'accordo con un ACK.

Essendo la rete dinamica, le informazioni dopo un po' di tempo devo essere rinnovate mediante un meccanismo di fading chiamato **leasing** secondo il quale sia il gestore sia la macchina interessata sanno quando scade l'informazione, in modo da poterla rinnovare prima che scada.

4.4.2 ARP

ARP, *Address Resolution Protocol*, è un protocollo di servizio appartenente alla suite del protocollo Internet IPv4, il cui compito è fornire la *mappatura* tra indirizzo IP e MAC di un terminale in una rete locale. Il suo analogo in IPv6 è NDP, *Neighbor Discovery Protocol*.

Per inviare un pacchetto IP, è necessario incapsularlo in un frame di livello data link, che dovrà avere come indirizzo di destinazione il MAC address dell'host a cui lo si vuole inviare. ARP viene utilizzato per ottenere questo indirizzo. Se il pacchetto deve essere inviato ad un'altra sottorete, l'indirizzo MAC da richiedere sarà quello del gateway o del router.

Ogni macchina connessa alla rete conserva in memoria una tabella, detta *ARP Cache*, con le corrispondenze IP-MAC già precedentemente richieste, in modo da evitare d'interrogare continuamente la rete per inviare ogni pacchetto. Le informazioni contenute in questa cache vengono cancellate dopo un certo periodo di tempo, tipicamente 5 minuti.

Quando un host deve inviare un pacchetto ad un IP non presente in tabella, manda in broadcast un messaggio detto *ARP REQUEST*, contenente il proprio indirizzo MAC e l'indirizzo IP della macchina di cui si vuole conoscere l'indirizzo IP. Tutti gli host della rete ricevono la richiesta e, in ciascuno di essi, il protocollo ARP verifica se viene richiesto il proprio indirizzo MAC confrontando il proprio IP con quello ricevuto. L'host che verifica positivamente questa corrispondenza provvede ad inviare in **unicast** una risposta, *ARP Reply*, contenente il proprio MAC.

Per ottimizzare il protocollo, una macchina, quando si collega alla rete, manda in broadcast la richiesta della propria associazione IP-MAC, così tutte le altre macchine la possono memorizzare. Inoltre, se qualcuno risponde significa che si è verificato un errore nell'assegnazione degli indirizzi IP.

Si noti che l'ARP Request ad un nodo aggiorna completamente la tabella ARP presente nella cache, senza rispetto per le voci già esistenti. Particolare molto importante perché causa di diversi attacchi chiamati *ARP Spoofing*, che verranno approfonditi nella sezione dedicata alla sicurezza delle reti.

4.4.3 NAT

Il NAT, *Network Address Translation*, costituisce una soluzione adottata per risolvere il problema dell'esaurimento degli indirizzi IP, **simulando una rete in un unico indirizzo**. Ad esempio, un'azienda può avere un solo indirizzo IP per il traffico Internet. Internamente, ogni host riceve un indirizzo univoco per instradare i dati nella rete aziendale. Quando il traffico esce da questa rete avviene una traduzione dell'indirizzo a quello di Internet, effettuato dal dispositivo NAT.

Il problema sta nel decidere a chi inviare internamente la risposta inviata dal Web, che come indirizzo ha quello generico aziendale. Siccome, la quasi totalità dei pacchetti IP trasporta carichi utili TCP e UDP, si è pensato di sfruttare queste informazioni per implementare l'inoltro. TCP e UDP, come si vedrà in seguito, utilizzano e includono nei propri header una porta sorgente e una di destinazione del pacchetto.

Ogni volta che un pacchetto diretto verso l'esterno raggiunge l'apparto NAT, oltre a sostituire l'indirizzo, viene sostituito anche il campo *Source*

port, con un indice che punta alla tabella di traduzione NAT. Al contrario, quando viene ricevuta la risposta, si usa il campo *Source port* ricevuto viene usato per ottenere l'indirizzo e la porta del mittente originale all'interno della rete NAT.

Sebbene questo schema in un certo senso risolva il problema dell'esaurimento degli indirizzi IP, molti della comunità lo considerano un vero e proprio abominio.

Viola il modello gerarchico di IP Non è più vero che ogni indirizzo IP identifica a livello mondiale una singola macchina.

Rompe il modello di connettività end-to-end Poiché l'associazione viene effettuata dai pacchetti in uscita, i pacchetti di entrata non possono essere accettati se non dopo quelli in uscita. Esistono comunque alcune tecniche per arginare questo problema.

Trasforma Internet in una rete orientata alla connessione Questo perché l'apparato NAT deve conservare le informazioni relative a ogni connessione che lo attraversa. **Se l'apparato NAT si blocca, tutte le sue connessioni TCP vanno perse**. Questo non succede in assenza di NAT.

Viola la regola principale della stratificazione dei protocolli Di norma, il livello k non deve essere costretto a formulare alcuna ipotesi su ciò che il livello $k + 1$ ha inserito nel payload. NAT distrugge questa indipendenza perché si basa sui livelli superiori TCP e UDP. Se in futuro TCP venisse aggiornato a TCPv2, con uno schema d'intestazione diverso, NAT non funzionerebbe più.

I processi su Internet non sono obbligati a usare TCP e UDP Se due utenti si accordassero per usare un nuovo protocollo di trasporto, il meccanismo di NAT non funzionerebbe più.

Alcune applicazioni usano più connessioni TCP o porte UDP NAT non sa gestire queste situazioni, a meno che non vengano prese speciali precauzioni.

Nonostante ciò, NAT è molto usato nella pratica, specialmente per reti domestiche o in piccole aziende, in quanto è l'unica tecnica che riesca a gestire il problema della mancanza di indirizzi IP. Per questa ragione è inverosimile che sparisca, anche qualora IPv6 fosse ampiamente adottato.

Capitolo 5

Transport

Il livello di trasporto si basa sul livello di rete per fornire il **trasporto dei dati da un processo su una macchina sorgente a un processo su una macchina destinazione** con un livello di affidabilità desiderato e indipendente dalle reti fisiche attualmente in uso. Fornisce le astrazioni necessarie alle applicazioni per utilizzare la rete.

5.1 UDP

UDP, *User Datagram Protocol*, è un protocollo di trasporto **senza connessione** che predilige la velocità al controllo. È usato dal protocollo DNS. Trasmette segmenti costituiti da un'intestazione di 8 byte seguita dal carico utile.

5.1.1 Header UDP

+	Bit 0-15	16-31
0	Source Port (optional)	Destination Port
32	Length	Checksum (optional)
64+	Data	

Figura 5.1: Header UDP

L'intestazione UDP è così composta:

- Source port e Destination port, per un totale di 4 byte, che indicano le porte a cui sono associati i processi di destinazione;
- 2 byte per la lunghezza totale del pacchetto, header e campo dati;
- 2 byte per il checksum opzionale globale: header, header IP *virtuale* e dati.

5.2 TCP

TCP Header

Bit offset	Bits 0–3	4–7	8–15	16–31
0	Source port			Destination port
32	Sequence number			
64	Acknowledgment number			
96	Data offset	Reserved	CWR ECE URG ACK PSH RST SYN FIN	Window Size
128	Checksum			Urgent pointer
160	Options (optional)			
160/192+	Data			

Figura 5.2: Header TCP

È stato progettato per riuscire a garantire **solide prestazioni anche 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 a 64 KB** che invia in un datagramma IP autonomo. È un **protocollo orientato alla connessione: sia il mittente che il ricevente devono creare dei socket** che possiedono un indirizzo (porta). Un singolo socket supporta più connessioni contemporanee. Le connessioni si identificano dalla coppia di porte (sorgente, destinazione). Le entità TCP si scambiano dati sotto forma di segmenti. Un segmento TCP consiste in un **header di 20 byte seguito dal payload**.

5.2.1 Header TCP

L'header TCP è composto da:

Source port - Destination port Identificano gli estremi locali della connessione.

Sequence Number Numero del pacchetto all'interno del flusso dati.

ACK Number Specifica il successivo byte previsto all'interno del flusso dati.

TCP Header Length Lunghezza dell'header espressa in word (gruppi di 32 bit). Necessaria perché il campo Options ha lunghezza variabile.

Gruppo di 4 byte inutilizzato Riservato per usi futuri. Il fatto che sia ancora inutilizzato indica la bontà del protocollo TCP.

Flags CWR-ECE Usati per segnalare una congestione.

URG Impostato a 1 quando ci sono dei dati urgenti referenziati da *Urgent Pointer*.

ACK Impostato a 1 se c'è un ACK in piggyback.

PSH Impostato a 1 se il pacchetto deve essere consegnato direttamente all'applicazione ricevente, senza essere preventivamente archiviato in un buffer.

RST Impostato 1 se la connessione non è più valida.

SYN Impostato a 1 quando viene richiesta l'apertura di una connessione.

FIN Impostato a 1 quando il mittente del pacchetto vuole terminare la connessione.

Window size Dimensione della finestra di ricezione del mittente, espressa in byte.

Checksum Calcolato mediante la somma in complemento a 1 dell'header (con campo checksum nullo), dei dati e di uno pseudo header composto da indirizzi IP del destinatario/mittente, un byte pari a 0, un byte di 1 e due byte che rappresentano la lunghezza totale del pacchetto.

Urgent pointer Puntatore alla fine della parte del pacchetto contenente i dati urgenti. L'inizio deve essere determinato dall'applicazione ricevente, senza aiuto da parte di TCP.

Options Opzioni facoltative per usi avanzati del protocollo.

5.2.2 Connessione TCP

1. Il server resta in attesa con le primitive *LISTEN* e utilizza *ACCEPT* per accettare connessioni in ingresso;

2. per connettersi un client esegue una primitiva *CONNECT* con $SYN = 1$ e $ACK = 0$;

3. quando questo segmento arriva al server, l'entità TCP ricevente controlla se esiste un processo server che ha eseguito una *LISTEN* sulla porta indicata nel campo *Destination port*. In caso negativo invia una risposta con $RTS = 1$ per rifiutare la connessione;

4. connessione accettata, si risponde con un segmento di ACK.

In caso di richiesta di contemporanea di connessione, ne viene presa in considerazione solo una.

Quando una connessione viene rilasciata, s'invia un segmento con $FIN = 1$. Quando si riceve l'ACK del FIN, il flusso in quella direzione viene chiuso. Tuttavia, può continuare nel verso opposto: **per interrompere totalmente una connessione**, entrambe le parti inviano un segmento con $FIN = 1$, in modo da segnalare che le due non hanno più nulla da dirsi. Quando un FIN riceve il corrispettivo ACK, la connessione in quella direzione viene chiusa. Quando entrambe saranno chiuse la connessione verrà rilasciata. Per evitare il problema dei due eserciti, cioè il caso in cui il segmento di risposta vada perso, si usano dei timer in modo da rilasciare comunque la connessione anche in mancanza di un ACK per un FIN.

Capitolo 6

Application

I livelli al di sotto del livello applicazione non svolgono alcun lavoro per gli utenti. Come negli altri livelli, anche in quello applicativo sussiste l'esigenza di protocolli di supporto che permettono alle applicazioni di funzionare. Uno di questi è DNS, che gestisce i nomi di dominio all'interno di Internet.

6.1 DNS

Anche se i programmi possono teoricamente fare riferimento a host, caselle di posta e altre risorse tramite i loro indirizzi di rete (IP), le persone ricordano quest'ultimi con difficoltà.Perciò, sono stati introdotti **nomi ad alto livello** per fornire alle macchine un nome più leggibile e mnemonico, separato dai rispettivi indirizzi. Poiché la rete intrinsecamente accetta solo indirizzi IP, è necessario tuttavia un **meccanismo per convertire i nomi in indirizzi di rete**. Per risolvere questo problema fu inventato DNS, *Domain Name System*.

DNS introduce uno **schema di denominazione gerarchico basato su domini, implementato attraverso un database distribuito**. È usato principalmente per associare nomi di host a IP, ma può essere usato anche per altri scopi (es alias).

6.1.1 Risoluzione di un nome di dominio

Per **risolvere** un nome di dominio e ottenere quindi il corrispettivo indirizzo IP, un programma applicativo invoca una procedura di libreria chiamata *resolver*, passando il nome come un parametro. Il resolver invia un **pacchetto UDP** contenente la richiesta al server DNS locale: se il dominio cercato è all'interno della giurisdizione di tale *name server*, un gruppo composto dall'indirizzo IP e altre informazioni, detto *record*, viene restituito. Altrimenti, il name server inizia una ricerca in remoto chiamata **interrogazione ricorsiva**:

1. si parte dalla cima della gerarchia dei nomi chiedendo a ognuno dei ***root name server***, che hanno informazioni su ognuno dei domini di primo livello. Questi name server sono molto importanti e tutti gli altri possiedono informazioni su come trovarli;

2. il root server restituisce informazioni sul name server che si occupa di quel particolare dominio di primo livello cercato;

3. il processo si ripete, appunto in maniera ricorsiva, finché non si riesce a risolvere il nome o finché uno dei server interpellati durante la ricorsione non restituisce un messaggio d'errore perché non contiene record riguardanti il dominio cercato.

6.1.2 Composizione di un record DNS

Ogni record è una quintupla formata dai seguenti elementi.

Nome Dominio indica il nome del dominio a cui si riferisce il record.

Tempo di vita indica per quanto considerare valida l'informazione. È un indicatore della stabilità del record.

Classe fissa a IN per le informazioni riguardanti Internet.

Tipo specifica il tipo di record, ossia indica che cosa è contenuto nel campo valore (es: A per IPv4 o AAAA per IPv6).

Valore contiene il valore del record, può essere una stringa ASCII o un dominio.

6.1.3 La sicurezza e i DNS

Un server DNS può subire due tipi di attacchi: ai dati e al server stesso.

1. I dati di un server DNS **possono essere corrotti** quando il server vittima richiede informazioni agli altri server. Infatti, un attaccante può ascoltare la trasmissione e **rispondere alla richiesta con un record DNS falso** (*DNS Spoofing*).

2. Un server DNS inoltre può essere **vittima di attacchi DoS** per mandarlo in sovraccarico e far cadere un pezzo di rete.

Il sistema DNS può anche essere usato per fare attacchi di tipo DDoS, in cui l'attaccante manda, a nome della vittima, delle richieste di risoluzione di più domini a più server DNS i quali, con le loro risposte, sovraccaricheranno la macchina vittima (*DNS Amplification*).

DNSsec

Esiste una variante di DNS, **DNSsec** che aggiunge due campi al record:

Key La chiave pubblica del dominio.

SIG La firma del server DNS.

Questo protocollo ha bisogno di un'**informazione iniziale condivisa**, fornita dai root server, che a loro volta si basano su una **chiave conosciuta da solo 7 persone al mondo**.

Capitolo 7

Sicurezza

7.1 Introduzione

7.1.1 Glossario

Cifrario Trasformazione del testo originale **carattere per carattere**.

Codice Rimpiazzo di una **parola con un'altra**

I messaggi da cifrare sono detti **testo in chiaro**. L'output è il **testo cifrato**. L'arte di forzare i cifrari, chiamata **criptoanalisi** e l'arte di inventarli, **crittografia**, sono note sotto il nome collettivo di **crittologia**. **Decriptare** è l'attività di decifrazione da parte di un criptoanalista (intruso), mentre **decifrare** è l'operazione legittima di lettura di un messaggio cifrato.

7.1.2 Principio di Kerckhoff

Il principio di Kerckhoff afferma che **tutti gli algoritmi devono essere pubblici, solo le chiavi devono essere tenute segrete**. Tenere invece segreto l'algoritmo è una forma di sicurezza detta **per occultamento**. Il segreto sta quindi in un buon algoritmo con chiavi lunghe per aumentare il fattore lavoro.

7.1.3 Principi crittografici fondamentali

Ridondanza Tutti i messaggi cifrati devono contenere informazioni ridondanti, ossia non necessarie alla comprensione del messaggio.

Attualità È necessario avere la possibilità di verificare che ogni messaggi ricevuto sia attuale.

7.2 Chiave Condivisa

7.2.1 Cifrari a sostituzione

In un cifrario a sostituzione, **ogni lettera o gruppo di lettere viene rimpiazzato da un'altra lettera o gruppo di lettere** per mascherare il messaggio. Una semplice generalizzazione consiste nello **spostare l'alfabeto del testo cifrato di k lettere**. In questo caso k diventa la chiave del metodo generale, che sta nell'avere un alfabeto spostato circolarmente. Il miglioramento successivo consiste nel far sì che ognuno dei simboli del testo in chiaro corrisponda ad altri simboli. Il sistema generale per la sostituzione simbolo a simbolo viene chiamato **sostituzione monoalfabetica**, dove la chiave è la stringa di lettere che corrisponde all'interno alfabeto.

Per decriptare i testi cifrati con questo metodo, si utilizza un **approccio statistico** chiamato *frequency analysis*, 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 ben nota. Un altro approccio è quello di provare con una parola che dato il contesto ha una buona probabilità di essere nel testo e da lì ricavare le varie lettere.

7.2.2 Cifrari a trasposizione

I cifrari a sostituzione conservano l'ordine dei simboli del testo in chiaro, limitandosi a mascherare la loro apparenza. I **cifrari a trasposizione, invece, riordinano le lettere** senza mascherarle. Si utilizza una matrice in cui il testo in chiaro viene scritto orizzontalmente, per righe, fino a riempire la matrice, eventualmente usando alcuni caratteri per occupare celle rimaste vuote. Il **testo cifrato viene trasmesso per colonna**, secondo un ordine stabilito in base ad una chiave usata: può essere anche una stringa con l'ordine determinato tramite lessicografia.

Per poter attaccare un cifrario a trasposizione si vede dapprima se le **frequenze delle lettere nel testo cifrato corrispondono alle frequenze statistiche** nella lingua presa in esame. Da ciò si può capire se il cifrario è a trasposizione. Il passo successivo è **scoprire di quante colonne e formata la matrice e il loro ordine**.

7.3 Algoritmi a chiave simmetrica

7.3.1 DES

Sono dei sistemi di crittografia che si basano su product cipher (combinazioni di P-Box e S-Box [1]). DES sfrutta 19 passaggi che lavorano su **blocchi da**

[1] I P-Box sono delle funzioni che prendono in input un blocco di bit e lo permutano. Gli S-Box prendono in input un blocco di bit e li alterano. Tutte le modifiche dipendono da una chiave

64 bit e con una chiave da 56 bit:

- 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;
- i passaggi intermedi sono funzionalmente uguali ma parametrizzati con diverse funzioni della chiave.

7.3.2 Triplo DES

Triplo DES si basa sull'**utilizzo a cascata di DES**. Per criptare il testo:

1. lo si codifica con una chiave K_1;
2. il risultato viene decodificato con una chiave K_2;
3. il tutto viene poi ricodificato ancora con K_1,

Questo procedimento è stato adottato per **retrocompatibilità con DES**: infatti, se si usano due chiavi uguali, la prima e la seconda operazione si annullano ed il terzo passaggio codifica il testo secondo l'algoritmo classico.

7.3.3 AES

AES, *Advanced Encryption Standard* è l'algoritmo, nato nel 1997 a seguito di un concorso pubblico, che ha sostituito DES come standard ufficiale del governo statunitense. Si tratta di un **cipher block** con blocchi e chiavi da 128 e 256 bit.

7.3.4 Cipher Modes

Electronic Code Book

Il testo viene diviso in blocchi che vengono poi **cifrati uno dopo l'altro** usando la stessa chiave. L'ultimo pezzo di testo in chiaro, se necessario, viene riempito per fargli raggiungere la lunghezza dei blocchi precedenti.Ogni blocco può dunque essere visto come pagina di un libro.

Crudelia può tuttavia memorizzarsi tutte le pagine e inviarle a Bob con un **ordine casuale rischiando di modificare lo stato** di Bob. Questo problema viene aggirato dai Cipher Modes che aggiungo ad un blocco criptato della dipendenze al blocco precedente.

Chaining Mode

Per evitare gli attacchi che possono capitare con ECB, si **collegano tutti i blocchi cifrati in diversi modi**, in modo che un eventuale operazione di scambio o rimpiazzo renda i dati senza significato a partire dal punto in cui è stata operata la sostituzione.

Ogni blocco di testo in chiaro è messo in XOR con il precedente blocco cifrato, prima di eseguire la cifratura vera e propria. Per il primo blocco, lo XOR viene calcolato con un blocco di dati casuali detto vettore di inizializzazione (IV). Un chiaro vantaggio di questo metodo sta nel fatto che **non produce lo stesso testo cifrato a partire da blocchi di testo in chiaro uguali**. Lo **svantaggio** principale consiste nel dover aspettare che un **intero blocco di testo cifrato arrivi a destinazione prima che la decifrazione possa cominciare**.

Feedback Mode

Usato in certi casi in cui è richiesta una decifrazione *al volo*, prima che tutto il blocco sia ricevuto, come nel caso dei terminali interattivi. Si usa quindi una cifratura byte a byte mediante un registro di shift ausiliario. Inizialmente, se non sono presenti dei byte già cifrati, il registro viene riempito con il vettore di inizializzazione. Per ottenere un byte criptato, si deve:

1. criptare il contenuto del registro;
2. prendere il byte più a sinistra e metterlo in XOR con il byte da criptare;
3. inserire il byte criptato nel registro.

Allo stesso modo per decriptare un byte, prima si cripta il registro e poi si fa lo XOR per ottenere il byte in chiaro. Infine si inserisce nel registro il byte criptato, cioè quello ricevuto.

Stream Mode

Anziché cifrare il testo, **si cifra il vettore di inizializzazione con una chiave crittografica e poi si usa il risultato per cifrare il testo mediante XOR**. Il risultato della cifratura di IV viene ulteriormente cifrato (keystream) per produrre il secondo blocco in uscita, quindi si prosegue analogamente per il terzo e via dicendo.

Counter Mode

Simile allo stream mode, anziché cifrare continuamente il vettore di inizializzazione **si cifra il vettore di inizializzazione più un numero sequenziale**. Se il vettore di inizializzazione non cambia tra una trasmissione e l'altra c'è il rischio di un attacco del tipo keystream reuse (vale anche per lo stream mode).

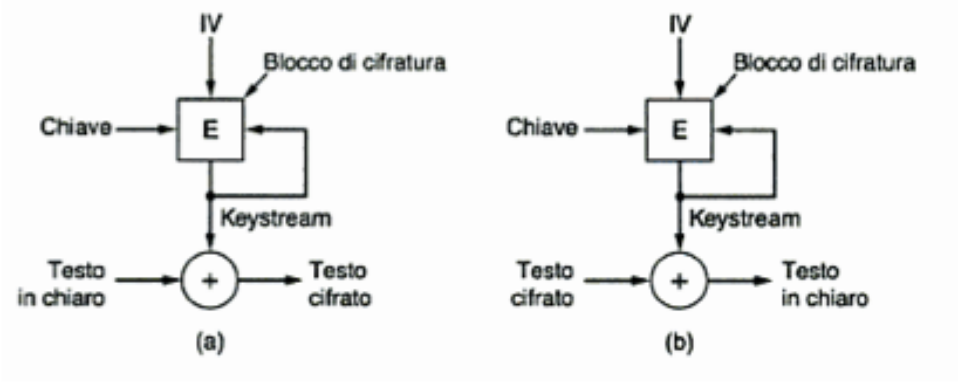

Figura 7.1: Stream Mode

7.4 Algoritmi a chiave pubblica

7.4.1 Diffie-Hellman

1. Alice e Bob si scambiano due numeri molto grandi, P e G in chiaro.
2. Alice e Bob scelgono un numero random ciascuno, A e B.
3. Alice invia a Bob $G^A \mod P$ (chiave pubblica di Alice), lo stesso vale per Bob ($G^B \mod P$)
4. Entrambi si calcolano $G^{A^B} \mod P$ che è la loro chiave segreta.
5. Alice e Bob possono ora comunicare in modo criptato usando la chiave segreta

Con i dati che può ottenere Crudelia, il calcolo della chiave segreta è un **problema intrattabile risolvibile solo a forza bruta**.

7.4.2 RSA

Alice vuole comunicare con Bob in modo segreto.

1. Bob sceglie due numeri P e Q molto grandi e primi.
2. Bob calcola:

   prod $= P \cdot Q$.
   ino $= (P-1) \cdot (Q-1)$.
   dec un numero coprimo di *ino*
   enc tale che $enc \cdot dec$ è congruo a $1 \mod ino$

3. Bob invia ad Alice la sua chiave pubblica, cioè la coppia *(end,prod)*

4. Alice divide il messaggio in blocchi di dimensione minore di *prod* e lo cripta con $C = P^{enc} \mod prod$

5. Bob decripta il messaggio in $P = C^{dec} \mod prod$

Crudelia per decifrare il testo ha bisogno della chiave privata di Bob e per calcolarla deve trovare P e Q. Il calcolo di P e Q è un **problema intrattabile e risolvibile solo mediante forza bruta**.

7.5 Firme digitali

7.5.1 Hash crittografico

Una funzione, per essere considerata un hash crittografico, deve avere le seguenti proprietà:

1. Dato P è facile calcolare $H(P)$;

2. Dato $H(P)$ è praticamente impossibile trovare P;

3. Dato P è praticamente impossibile trovare Q tale che $H(P) = H(Q)$;

4. Una piccola variazione di P fa variare completamente $H(P)$

Alcuni esempi di hash crittografici sono MD5, SHA-1 e SHA-2.

7.5.2 Hash Message Auth Code

E' un sistema di autenticazione a chiave condivisa. Assieme al messaggio viene inviato l'hash del messaggio e della chiave. Tramite HMAC è possibile garantire sia l'integrità che l'autenticità del messaggio.

7.5.3 Preimage Attack

È un attacco in cui si cerca di **trovare un messaggio che ha uno specifico valore hash**. Se l'hash è di n bit servono circa 2^n tentativi a forza bruta.

7.5.4 Birthday Attack

L'attacco del compleanno consiste, data una funzione f, nel trovare due numeri x_1 e x_2 tali che $f(x_1) = f(x_2)$. Tale coppia di valori (x_1, x_2) è chiamata **collisione**. A causa del paradosso del compleanno, quest'attacco può risultare efficiente: applicato agli hash, per trovare due messaggi che hanno lo stesso hash con una probabilità del 50% bastano $2^{\frac{n}{2}}$ tentativi.

7.6 Gestione delle chiavi pubbliche

7.6.1 Certificati

Vengono usati per garantire l'identità del proprietario e possono essere rilasciati solo da un Autorità di Certificazione. Uno standard per i certificati è X.509, il quale prevede vari campi tra cui:

- il numero di serie del certificato
- l'autorità che l'ha emesso
- il proprietario
- la sua chiave pubblica
- la firma della CA.

Per sapere che una CA è veramente una CA e non Crudelia che si finge tale, è necessario che ogni computer abbia al suo interno i certificati che certificano l'autenticità delle CA.

7.7 Sicurezza delle comunicazioni

7.7.1 IPsec

E' una variante del protocollo IP che aggiunge un sistema di autenticazione a chiave condivisa mediante Diffie-Hellman. IPsec può essere implementato in due modalità:

Transport I dati relativi all'autenticazione vengono inseriti all'inizio del campo dati del pacchetto IP.

Tunnel Viene creato un nuovo pacchetto IP contenente al suo interno i dati per l'autenticazione e il vecchio pacchetto IP.

Ci sono due protocolli supportati da IPsec ed entrambi possono funzionare in modalità transport o tunnel: **AH** e **ESP**.

Entrambi gli header sono simili e contengono i seguenti campi.

NextHeader Indica il tipo di protocollo usato per trasmettere i dati.

SPI Identifica, in combinazione con l'indirizzo IP, la Security Association.

Sequence Number Relativo alla connessione, serve per evitare attacchi di tipo replay.

La differenza tra i due protocolli sta nella parte autenticata (con HMAC):

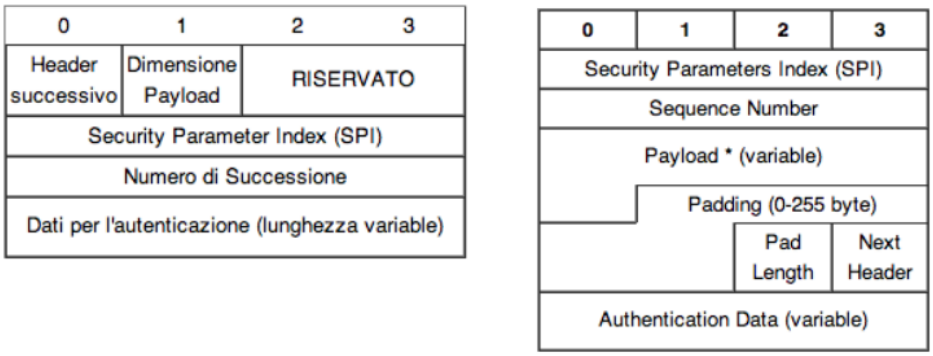

Figura 7.2: A sinistra l'header del protocollo AH mentre a destra quello ESP

- con AH viene autenticato l'intero pacchetto (i campi variabili vengono calcolati come se fossero 0);
- con ESP invece vengono autenticati solamente l'header ESP e il contenuto del pacchetto. Con ESP i dati trasmessi nel campo payload vengono criptati.

7.8 Protocolli di autenticazione

7.8.1 3-Way Handshake

E' un protocollo di autenticazione a chiave condivisa, usato anche nel sistema GSM. Alice e Bob vogliono comunicare tra loro, per essere sicuri di parlare l'uno con l'altro:

1. Alice sfida Bob mandandogli un numero casuale da criptare con la chiave condivisa;
2. Bob risponde alla sfida e ne lancia una analoga ad Alice;
3. Alice risponde alla sfida.

Dato che la chiave condivisa è conosciuta solo da Alice e Bob solo loro possono rispondere alla sfida.

Reflection Attack e soluzione

Crudelia sfrutta Bob per farsi passare la risposta:

1. Crudelia sfida Bob;
2. Bob risponde alla sfida e a sua volta sfida Crudelia con un numero X;

3. Crudelia, che non può rispondere alla sfida, sfida nuovamente Bob usando però il numero X;

4. Bob risponde alla sfida fornendo a Crudelia la risposta alla prima sfida;

5. Crudelia risponde alla prima sfida di Bob.

In questo modo Bob crede di comunicare con Alice quando in realtà sta comunicando con Crudelia. Nella realtà viene quindi usata una versione autenticata:

1. Alice sfida Bob mandandogli R_a

2. Bob risponde con R_b,$HMAC(R_a, R_b, A, B, K_{ab})$

3. Alice risponde con $HMAC(R_a, R_b, K_{ab})$

Quando Alice riceve la risposta di Bob può calcolarsi l'HMAC e controllare se coincidono. Crudelia non potrà mai generare lo stesso HMAC dato che non conosce la chiave condivisa.

7.9 Minacce alla sicurezza

7.9.1 Attacco Man in the Middle

È 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 che il collegamento è stato compromesso**. L'attaccante deve essere in grado di osservare e intercettare il transito dei messaggi tra le due vittime.

Supponiamo che Alice voglia comunicare con Bob, e che Giacomo voglia spiare la conversazione, e se possibile consegnare a Bob dei falsi messaggi.

1. 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.

2. Giacomo può semplicemente inviare ad Alice una chiave pubblica della quale possiede la corrispondente chiave privata.

3. 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.

4. Giacomo quindi li intercetta, li decifra, ne tiene una copia per sé, e li cifra nuovamente, dopo averli alterati se lo desidera, usando la chiave pubblica che Bob aveva originariamente inviato ad Alice.

5. Quando Bob riceverà il messaggio cifrato, crederà che questo provenga direttamente da Alice.

Questo tipo di attacco può essere fatto anche all'interno di una rete locale mediante ARP Poisoning.

7.9.2 DoS

È una tipologia di attacco volta a far collassare la macchina vittima. Un attacco DoS, *Denial of Service*, può essere fatto a livello network (*smurf*) con le richieste ECHO di ICMP, oppure a livello Transport (*SYN attack*) inviando in continuazione richieste di connessione TCP senza confermarle.

Un attacco DoS può essere fatto anche in versione *distribuita* (DDoS) mediante l'utilizzo di più macchine zombie che attaccano contemporaneamente la vittima. Esiste inoltre una versione *reversed* effettuata mediante *IP spoofing*, nella quale l'attaccante invia un elevato numero di pacchetti ECHO modificati che hanno come mittente l'IP della vittima.

7.10 Sicurezza nelle reti WireLess

7.10.1 FHSS

FHSS, *Frequency Hopping Spread Spectrum*, consiste nell'avere la **frequenza di trasmissione che varia ad intervalli regolari**, rendendo più difficile intercettare la trasmissione.

7.10.2 Bluetooth

La sicurezza con Bluetooth si basa su FHSS e un sistema di crittografia a chiave simmetrica con blocchi da 128 bit.

7.10.3 802.11

WEP

In WEP, *Wired Equivalent Privacy*, le trasmissioni vengono cifrate mediante una chiave simmetrica da 40-104-232 bit (WEP-40/104/232) e un cifrario a blocchi RC4 secondo la modalità stream chyper.

Il **punto debole del sistema WEP è il vettore di inizializzazione che, essendo composto da soli 24 bit, rende probabile il riutilizzo di una sua configurazione** (attacco di tipo *keystream reuse*). Infatti, già nel 2001 si riusciva a bucare una rete WEP in circa 15 minuti. Da segnalare, inoltre, l'attacco ai magazzini Marshall che ha permesso ai malintenzionati di rubare 45.7 milioni di carte di credito (evento registrato come record del mondo).

WPA

WPA sfrutta TKIP, un protocollo di cifratura basato su RC4 con una chiave (*passphrase*) di 128 bit (da 8 a 32 caratteri ASCII). Introduce tuttavia un nuovo problema di origine sociale: **gli utenti tendono ad usare chiavi corte e simili a parole**, creando una vulnerabilità agli attacchi di tipo *dictionary*.

Nel 2008 viene scoperta una falla che ha portato allo standard **WPA2**, in cui viene usato CCMP (*Counter Mode with Cipher Block Chaining Message Authentication*); chiavi e blocchi restano però di 128 bit.

Capitolo 8

Appendice: 802.11 + Denteblu