

Reti di comunicazione

Raffaele Daniele Facendola

November 10, 2011

Contents

1 Sorgenti di traffico	4
1.1 Modello ON-OFF deterministico - VBR	4
1.2 Modello ON-OFF statistico - VBR	4
2 Tecniche di moltiplicazione	6
2.1 TDM	6
2.1.1 Moltiplicazione deterministica	6
2.1.2 Moltiplicazione statistica	6
3 Tecniche di commutazione	7
3.1 Commutazione di circuito	7
3.2 Commutazione di pacchetto	7
3.2.1 Connection oriented (CO)	7
3.2.2 Connectionless (CL)	7
4 Segnali	8
4.1 Mezzi trasmissivi	8
4.1.1 Thick ethernet	8
4.1.2 Thin ethernet	8
4.1.3 Fibra ottica	8
4.2 Configurazioni di linea	9
4.3 Trasmissione dei segnali	9
5 Tecniche di codifica	10
5.1 Codifica unipolare	10
5.2 Codifica polare	10
5.2.1 Non return to zero level (NRZ-L)	10
5.2.2 Non return to zero inverted (NRZ-I)	10
5.2.3 Return to zero	10
5.2.4 Coded mark inversion (CMI)	10
5.2.5 Manchester	10
5.2.6 Manchester differenziale	11
5.3 Codifica bipolare	11
5.3.1 Alternate mark inversion (AMI)	11
5.3.2 Pseudoternario	11
5.4 Codici a blocchi	11
6 Protocolli ARQ (Automatic Repeat reQuest)	12
6.1 Stop & Wait	12
6.2 Go-back-n	13
6.3 Selective repeat	14

7	Protocollo HDLC (High-level datalink control)	15
7.1	Bit stuffing	15
7.2	Tipologie di trame	15
7.3	Piggybacking	15
7.4	Frame check sequency	16
8	Protocollo ALHOA	17
8.1	Protocollo Slotted ALOHA	17
9	Progetto IEEE 802	18
9.1	Livello LLC	18
9.2	Livello MAC	19
9.2.1	Ethernet	20
9.2.2	Fast ethernet	22
9.2.3	Gigabit ethernet	23
9.2.4	10 Gigabit ethernet	24
10	Prestazioni	24
10.1	Calcolo del throughput	24
10.2	Efficienza di ALHOA	24
10.3	Efficienza di Slotted ALHOA	25
11	NScript e Nam	26
11.1	Nodi	26
11.2	Link	26
11.3	Agent	26
11.3.1	UDP agent	27
11.3.2	NULL agent	27
11.3.3	Loss monitor agent	27
11.4	Sorgenti di traffico	27
11.4.1	CBR	27
11.4.2	ExpOnOff	28
11.4.3	Poisson	28
11.4.4	VBR	28
11.5	Eventi	28

1 Sorgenti di traffico

In generale il modello di una sorgente di traffico è basata su 3 parametri:

- Frequenza di picco **P**: indica quanti bit vengono emessi ogni secondo dalla sorgente attiva (durante l'intervallo di tempo T_{on})
- Frequenza media **A**: indica quanti bit vengono emessi in media ogni secondo dalla sorgente
- Fattore di burstiness **B**: Definito come A/P . Il fattore è anche uguale a $\frac{T_{on}}{T_{on}+T_{off}}$

Riconosciamo due tipi di sorgenti: **CBR** (Constant bit rate) in cui il fattore di burstiness è uguale ad 1 ($A = P$) e **VBR** (Variable bit rate) il cui fattore di burstiness è inferiore ad 1 (la sorgente emette informazioni in maniera intermittente).

1.1 Modello ON-OFF deterministico - VBR

Una sorgente di questo tipo alterna periodi di trasmissione alla velocità di picco (ON) a periodi di silenzio (OFF). I periodi di ON T_{ON} e di OFF T_{OFF} sono fissati ed uguali tra loro.

1.2 Modello ON-OFF statistico - VBR

Si comporta allo stesso modo delle sorgenti deterministiche, ma in questo caso i tempi di T_{ON} e T_{OFF} sono due variabili aleatorie (distribuzione esponenziale o poissoniana).

Nel caso di modelli statistici **esponenziali** i valori di T_{ON} e di T_{OFF} sono generati da un processo **senza memoria** (o markoviano). Vengono introdotti i seguenti valori:

- P_{ON} probabilità che la sorgente sia attiva
- P_{OFF} probabilità che la sorgente sia spenta
- λ frequenza media di transizione da OFF ad ON
- μ frequenza media di transizione da ON ad OFF
- $1/\lambda$ tempo medio di permanenza nello stato di OFF
- $1/\mu$ tempo medio di permanenza nello stato di ON

Con i valori di cui sopra è possibile calcolare la probabilità di transizione dallo stato OFF allo stato ON mediante il prodotto $P_{OFF} \cdot \lambda$ e la probabilità che avvenga il contrario mediante $P_{ON} \cdot \mu$

Il sistema di cui sopra gode della proprietà di **congruenza** per la quale $P_{ON} + P_{OFF} = 1$ e di **equilibrio statistico** per il quale $\lambda \cdot P_{OFF} = \mu \cdot P_{ON}$ da cui segue che:

- $P_{ON} = \frac{\lambda}{\lambda + \mu}$
- $P_{OFF} = \frac{\mu}{\mu + \lambda}$
- $B = \frac{1/\mu}{1/\lambda + 1/\mu} = \frac{\overline{T_{ON}}}{T_{ON} + T_{OFF}}$

Trasformazione di una sorgente VBR in una VBR equivalente (o una CBR)

E' sempre possibile trasformare il modello di una VBR in un altro purchè si conservino i valori:

- La somma dei tempi di ON e di OFF $T_{ON} + T_{OFF}$
- La frequenza media di trasmissione A.
- La lunghezza dei pacchetti L

E' possibile azzerare totalmente il periodo di OFF in modo da trasformare una sorgente VBR in una CBR.

2 Tecniche di moltiplicazione

Per moltiplicazione intendiamo la modalità di divisione di un canale unico affinché più sorgenti possano trasferire dati in maniera indipendente.

Esistono due tipi di moltiplicazione: la **TDM** (Time division multiplexing) e la **FDM** (Frequency division multiplexing)

2.1 TDM

La moltiplicazione a divisione di tempo consiste nel garantire a ciascuna sorgente un intervallo di tempo (**slot**) in cui poter fruire di tutta la capacità del canale per trasferire le informazioni desiderate.

2.1.1 Moltiplicazione deterministica

È la moltiplicazione più semplice: ad ogni istante di tempo t_i la sorgente $[i]_N$ (i modulo N, dove N è il numero totale di sorgenti) può inviare dati: qualora non dovesse essere pronta lo slot sarà "sprecato". Ogni sorgente invia dati ogni Nt secondi, mentre il **frame** (ovvero l'intervallo di tempo campione suddiviso tra le varie sorgenti) è costituito esattamente da N slot.

2.1.2 Moltiplicazione statistica

Questa moltiplicazione viene usata quando il numero di sorgenti è troppo alta o comunque non tutte le sorgenti trasferiscono dati a frequenza costante causando l'inattività del canale per troppo tempo. In questo caso il frame è diviso in un numero di slot più basso del numero delle sorgenti e, pertanto, il corretto funzionamento è garantito da un buffer in cui vengono memorizzate le trame in eccesso. Il moltiplicatore deve garantire un flusso in uscita superiore alla somma delle frequenze medie di tutte le sorgenti $f_m \geq \sum_{i=1}^N A_i$

3 Tecniche di commutazione

Per commutazione s'intende l'insieme delle azioni necessarie affinché una **unità informativa** (UI) venga instradata attraverso il ramo di uscita più opportuno.

3.1 Commutazione di circuito

In questo caso tutte le UI da trasferire seguiranno un percorso stabilito ad inizio comunicazione: vi è una fase iniziale di **instaurazione** in cui viene inviata una UI al fine di configurare i nodi intermedi e quindi vengono inviati i dati veri e propri. Una volta instaurata la connessione i router si limiteranno a diffondere la UI lungo il percorso memorizzato senza ulteriori ritardi. A comunicazione terminata vi è la fase di **rilascio** in cui il circuito viene distrutto. Un esempio è il servizio di telefonia: il circuito si instaura ad inizio chiamata, durante la comunicazione il percorso seguito dai dati rimane invariato e quindi il circuito viene distrutto al termine della stessa.

3.2 Commutazione di pacchetto

In questo caso i dati non vengono inviati in un unico flusso continuo ma vengono suddivisi in trame più piccole dette pacchetti.

3.2.1 Connection oriented (CO)

Si virtualizza la commutazione di circuito: vi è una fase in cui viene stabilito il percorso dei pacchetti (**instaurazione percorso end-to-end**) e quindi questi ultimi vengono inviati in sequenza attraverso lo stesso. Alla fine della trasmissione esso viene rilasciato. I servizi connection oriented garantiscono che le trame arrivino sempre in ordine corretto. Per i servizi connection oriented il tempo di ritardo è sempre costante, mentre la multiplazione adottata è di tipo deterministico.

3.2.2 Connectionless (CL)

Non viene instaurato nessun percorso end-to-end e tutte le trame vengono trattate come unità indipendenti: le trame inviate possono seguire percorsi differenti e non vi è alcuna garanzia che la sequenza sia rispettata. A differenza dei servizi CO il tempo di ritardo può variare nel tempo e la multiplazione è di tipo statistico.

4 Segnali

Per poter ricostruire dei segnali a partire dai suoi campioni è necessario che la frequenza di campionamento sia superiore al doppio della frequenza massima del segnale da campionare (**teorema di Nyquist-Shannon**)

4.1 Mezzi trasmissivi

In un determinato mezzo trasmissivo la potenza del segnale si attenua in modo esponenziale con la distanza di propagazione:

$I(l) = I(0)e^{-\alpha l}$ dove $I(0)$ è la potenza in ingresso, $I(l)$ è la potenza in uscita e l è la lunghezza del tratto. L'attenuazione in decibel è pari a $\alpha_{[dB]} = 10 \log_{10}(\frac{I(0)}{I(l)})$. L'unità di misura di α è il dB/Km .

4.1.1 Thick ethernet

Cavo coassiale in rame schermato usato nella specifica 10Base5. Ha un'attenuazione pari a $50 - 80 dB/Km$.

4.1.2 Thin ethernet

Cavo coassiale in rame schermato usato nella specifica 10Base2. Ha un'attenuazione pari a $100 - 150 dB/Km$.

4.1.3 Fibra ottica

A seconda di come la luce si propaga nel mezzo riconosciamo la fibra ottica **monomodale** e la fibra ottica **multimodale**. La prima è più veloce della seconda in quanto la sorgente utilizzata è di tipo Laser (contro il LED di quella multimodo) e ciò consente di coprire distanze maggiori.

Giacchè la fibra ha dei valori di attenuazione che dipendono dalla lunghezza d'onda del segnale che si vuole trasmettere riconosciamo tre intervalli (detti **finestre**) in cui l'attenuazione risulta particolarmente bassa:

- Prima finestra: Posta a cavallo di 820nm di lunghezza d'onda, ha un'attenuazione pari a $2 dB/Km$. E' la più economica.
- Seconda finestra: a cavallo di 1310nm ha un'attenuazione pari a $0.4 dB/Km$.
- Terza finestra: a cavallo di 1550nm ha un'attenuazione pari a $0.2 dB/Km$. Richiede le sorgenti più costose.

4.2 Configurazioni di linea

Il mezzo trasmissivo può essere di tipo **simplex** (se l'informazione viaggia lungo una sola direzione), **half-duplex** (se l'informazione può viaggiare in entrambi i versi ma non contemporaneamente) o **full duplex** (se l'informazione viaggia in entrambi i sensi, anche contemporaneamente).

4.3 Trasmissione dei segnali

Se un canale è soggetto ad un rumore allora la capacità massima del suddetto sarà pari a $C = W \log_2(1 + SNR)$ dove W è la frequenza di trasmissione e SNR (**signal-to-noise ratio**) è il rapporto tra la potenza del segnale trasmesso e la potenza del rumore (**teorema di Shannon-Hartley**). La differenza (in dB) tra la potenza media del trasmettitore e la sensibilità del ricevitore è detta **power budget**. Essa è pari a $P_{Tx} - P_{Rx} = \alpha l + A_{extra} + M$ dove l è la lunghezza del tratto, A_{extra} è l'attenuazione extra ed M è il margine di "sicurezza". L'indice di cui sopra è, generalmente, il primo strumento di dimensionamento per un sistema trasmissivo.

5 Tecniche di codifica

Per tecnica di codifica s'intende l'insieme delle strategie che ci permettono di mappare una sequenza di simboli (0 ed 1) in valori di grandezze fisiche da imprimere al mezzo trasmissivo per trasmettere l'informazione (corrente, tensione, ecc.).

5.1 Codifica unipolare

- 0 mappato come 0 (assenza di segnale)
- 1 mappato come livello alto HL (+)

5.2 Codifica polare

La notazione X/Y indica che a metà del periodo vi è una transizione da X ad Y.

5.2.1 Non return to zero level (NRZ-L)

- 0 mappato come livello basso (LL, -)
- 1 mappato come HL

5.2.2 Non return to zero inverted (NRZ-I)

- 0 mappato come livello precedente (se l'ultima codifica ha restituito LL allora viene mappato come LL)
- 1 mappato come livello precedente invertito (se l'ultima codifica è stata HL allora viene mappato come LL)

5.2.3 Return to zero

- 0 mappato come transizione LL/0
- 1 mappato come transizione HL/0

5.2.4 Coded mark inversion (CMI)

- 0 mappato come LL/HL
- 1 mappato una volta come LL/LL e l'altra volta come HL/HL

5.2.5 Manchester

- 0 mappato come HL/LL
- 1 mappato come LL/HL

5.2.6 Manchester differenziale

- 0 mappato come HL/LL se l'ultimo codice è stato HL/LL, mappato come LL/HL se l'ultimo codice è stato LL/HL (vi è una transizione rispetto all'ultimo codice generato)
- 1 mappato come LL/HL se l'ultimo codice è stato HL/LL, mappato come HL/LL se l'ultimo codice è stato LL/HL (non vi è nessuna transizione rispetto all'ultimo codice generato).

5.3 Codifica bipolare

5.3.1 Alternate mark inversion (AMI)

- 0 mappato come assenza di segnale
- 1 mappato una volta come HL e l'altra volta come LL

Per evitare lunghe sequenze di zeri si effettuano delle sostituzioni (**scrambling**):

- Specifica **bipolar 8-zero substitution** (B8ZS): se l'ultimo segnale era positivo una sequenza di 8 zeri è sostituita con la stringa **000+0+**, se l'ultimo segnale era negativo invece la sequenza viene sostituita con **000-0+**.
- Specifica **high-density bipolar 3** (HDB3): vi è una violazione di codice ogni 4 zeri consecutivi.

5.3.2 Pseudoternario

- 0 mappato una volta come HL e l'altra volta come LL
- 1 mappato come assenza di segnale

5.4 Codici a blocchi

E' possibile mappare blocchi di m bit mediante codice di n bit (come m minore di n, ovviamente) usando opportune tabelle.

- Codice binario: mBnB (esempi: 4B5B, 8B10B, 64B66B)
- Codice ternario: mBnT (esempi: 4B3T, 8B6T)
- Codice quaternario: mBnQ (esempi 2B1Q)

6 Protocolli ARQ (Automatic Repeat reQuest)

Questo tipo di protocolli si preoccupano di richiedere il rinvio di pacchetti in caso di errori di trasmissione di varia natura.

6.1 Stop & Wait

La strategia su cui si basa S&W è molto semplice: si invia un pacchetto e quindi si attende un riscontro da parte del destinatario mediante un pacchetto di ACK (Acknowledge) entro un intervallo di tempo prestabilito. Se l'ACK arriva allora il pacchetto successivo può essere inviato, in alternativa, se l'ACK non arriva e l'intervallo di tempo scade (**timeout**) si presuppone che il pacchetto sia stato perduto e quindi si procede al reinvio.

Il tempo necessario per trasferire N trame è dato dalla seguente:

$T_{tot} = N(T_f + T_a + 2T_p + 2\tau)$ dove T_f è il tempo necessario per trasferire la trama (dipende fondamentalmente dalla lunghezza della trama e dalla capacità del canale), T_a è il tempo di trasmissione relativo alla trama di riscontro, T_p è il tempo di elaborazione del sorgente e del destinatario e τ è il tempo di propagazione necessario affinché la trama arrivi da un capo all'altro della linea (viene considerata due volte perché si considera sia la trama che il riscontro).

I tempi di trasferimento T_f e T_a sono rispettivamente pari a Lf/C e L_a/C dove C è la capacità del canale ed L è la lunghezza delle due trame.

In **assenza di errori** l'efficienza del protocollo è

$\eta = \frac{1}{1+2a}$ dove a è il tempo di propagazione normalizzato $a = \frac{\tau}{T_f} = \frac{d \cdot C}{v \cdot L}$ con d distanza tra le stazioni, v velocità di propagazione ed L lunghezza del pacchetto.

L'efficienza del collegamento, anche detta throughput è data da:

$$THR = \eta C.$$

Il valore minimo del time-out deve essere pari a $T_a + 2\tau$ questo per garantire che la trama di ACK abbia il tempo di arrivare a destinazione senza far scadere il timeout.

Ogni trama presenta anche un bit di numerazione (che vale alternativamente 0 o 1), ciò serve per scartare i pacchetti o i riscontri duplicati.

In **presenza di errori** l'efficienza massima del protocollo sarà $\eta_{max} = \frac{1-P}{1+2a}$ dove P è la probabilità che ci sia una trama errata.

6.2 Go-back-n

Nel protocollo GBN le trame seguono una numerazione ciclica modulo $N = 2^b$ dove b è il numero di bit utilizzato per la codifica dell'indice di trama. Rispetto allo S&W il protocollo GBN trasmette tutte le trame in sequenza senza aspettare per ciascuna di essere un riscontro: qualora il destinatario dovesse incorrere in errori richiederà il rinvio delle trame a partire da quella che ha subito l'errore. Il mittente appena riceve la trama di riscontro negativo NACK (n) provvederà immediatamente alla trasmissione delle nuove trame (a partire da n).

Il tempo di trasferimento di N trame è $T_{tot} = NT_f + T_a + 2\tau = N\frac{L_f}{C} + \frac{L_a}{C} + 2\tau$.

Per controllare il flusso sul collegamento dati si adotta la strategia della **sliding window**. In fase di trasmissione l'ampiezza della finestra indica l'intervallo di indici delle trame che possono essere inviate senza ricevere un riscontro. Ogni riscontro in **sequenza** fa scorrere la finestra (ovvero incrementa di 1 sia il limite inferiore che quello superiore dell'intervallo in modulo N).

In fase di ricezione l'ampiezza della finestra indica il numero di trame che possono essere accettate, ogni accettazione in sequenza fa scorrere la finestra in ricezione.

Per evitare ambiguità la somma delle ampiezze delle finestre deve essere minore o uguale ad N .

Per evitare perdite di efficienza la seguente condizione deve essere soddisfatta: $W_s T_f \geq T_f + T_a + 2\tau$. Se la condizione di cui sopra non venisse rispettata il riscontro relativo al primo pacchetto inviato arriverebbe oltre la fine della trasmissione dell'ultima trama, costringendo il trasmettitore ad attendere prima di poter inviare le nuove trame. In questo caso si dice che la finestra sta **strozzando** la comunicazione.

In **assenza** di errori l'efficienza garantita dal protocollo è

- $\eta = 1$ se la finestra non strozza, $W_s > 1 + 2a$
- $\eta = \frac{W_s T_f}{T_f + 2\tau}$ se la finestra strozza

In **presenza** di errori l'efficienza, invece, è:

- $\eta = \frac{1-P}{1+P+2aP}$ dove P è la probabilità che una trama sia errata (se la finestra non strozza)
- $\eta = \frac{W_s(1-P)}{(1+2a)(1+(W_s-1)P)}$ se la finestra strozza

6.3 Selective repeat

Il protocollo SR riprende la stessa idea del GBN, tuttavia in caso di mancato riscontro non costringerà la sorgente ad inviare tutta la sequenza a partire dalla trama mancante ma soltanto quella che è stata persa. Una differenza sostanziale è che il selective repeat richiede che l'ampiezza della finestra in trasmissione e di quella in ricezione debba essere esattamente la metà di N .

In **assenza** di errori l'efficienza garantita è uguale al protocollo GBN.

In **presenza** di errori, invece, essa è pari a:

- $\eta = 1 - P$ se la finestra non strozza, $W_s > 1 + 2a$
- $\eta = \frac{W_s(1-P)}{1+2a}$ se la finestra strozza

7 Protocollo HDLC (High-level datalink control)

La trama del protocollo HDLC è costituita nel modo seguente:

- **Flag:** 8 bit, indica l'inizio di trama o la fine (01111110)
- **Address:** Multiplo di 8 bit, indica l'indirizzo della stazione a cui è indirizzato la trama
- **Control:** 8 o 16 bit, Indica il tipo di trama e contiene la numerazione
- **Information:** variabile, contiene le informazioni che si vogliono trasferire
- **FCS** (frame check sequency): 16 o 32 bit, usato per la rilevazione degli errori

7.1 Bit stuffing

Per evitare che all'interno della trama possa esistere la combinazione 01111110 usata dal procollo per delimitare il singolo pacchetto, prima dell'invio della trama viene inserito uno 0 ogni sequenza di cinque 1. In fase di ricezione, invece, esso viene rimosso seguendo la stessa logica. Il meccanismo è totalmente trasparente al protocollo e prende il nome di **bit stuffing**.

7.2 Tipologie di trame

- **Information:** La trama contiene delle informazioni, il campo "control" inizia con uno 0 a cui seguono 3 bit di numerazione del pacchetto. I restanti 4 bit vengono usati per fare **piggybacking**.
- **Supervisory:** La trama inizia con la sequenza 10, i restanti due bit rappresentano le 4 trame supervisory (**Received-Ready (ACK)**, **Received-NotReady (ACK** ma con richiesta di disconnessione), **Reject (NACK** per protocollo GBN), **Selective Reject (NACK** per protocollo SR). I restanti 4 bit servono per fare piggybacking (una trama di questo tipo non ha campo **information**).
- **Unnumbered:** Trame di controllo varie che iniziano con 11: indicano connessione, disconnessione e via dicendo (una trama di questo tipo non ha campo **information**).

Il 5° bit del campo "control" **P/F** è settato ad 1 se la stazione primaria richiede una risposta dalla secondaria (**poll**) o se la stazione secondaria ha finito di trasmettere (**final**). In ogni altro caso è posto uguale a 0.

7.3 Piggybacking

Nelle trasmissioni full-duplex è possibile sfruttare le trame informative che da A vengono inviate a B per inviare anche riscontri (ACK o NACK) relativi ai pacchetti che B ha inviato ad A in precedenza senza dover necessariamente inviare una trama di ACK o NACK apposita per ottimizzare i tempi. Ci si riferisce a questa tecnica con il nome di **piggybacking**.

7.4 Frame check sequency

La sequenza di FCS a k bit protegge la trama da eventuali errori che possono insorgere durante la trasmissione dati.

Qualora il mittente volesse trasmettere una trama $P(X)$, il contenuto del campo "informazioni" dovrebbe essere riempito con la sequenza: $P'(X) = P(X)X^k + R(X)$ dove $R(X)$ è definito come $P(X)/D(X)$ dove $D(X)$ è un polinomio divisore dato.

In fase di ricezione il destinatario sarà sicuro di aver ricevuto una trama corretta se e solo se il resto della divisione $\frac{P'(X)}{D(X)}$ sarà pari a 0. In caso contrario la trama contiene degli errori. Per passare da una stringa binaria ad un polinomio della forma $X^a + X^b + X^c$ è sufficiente considerare che il polinomio avrà un numero di termini X pari al numero di 1 contenuti nella stringa e ciascuna X sarà elevata ad un esponente pari agli indici di ciascun 1. (esempio $1010 = X^3 + X^1$)

8 Protocollo ALHOA

Il protocollo ALHOA è molto semplice ed è basato sul seguente algoritmo:

- La stazione trasmette il pacchetto
- Attende il riscontro ACK (tempo impiegato 2τ)
- Se il riscontro è arrivato la trasmissione è finita, altrimenti c'è stata una collisione, quindi la stazione attende un tempo **casuale** compreso tra 0 e KT , dove T è il tempo di trasmissione del pacchetto e K dipende dal numero di collisioni avvenute in precedenza secondo la legge $2^k - 1$, con k il minimo tra il numero di collisioni n ed un limite detto **backoff limit**.

Ogni stazione è indipendente e quindi può trasmettere in qualsiasi istante di tempo: la trasmissione ha successo solo se nessun'altra stazione trasmette nei T istanti precedenti alla stessa. Il periodo di vulnerabilità è pertanto $2T$ (T precedente alla trasmissione e T necessario affinché il pacchetto si diffonda in tutta la rete).

8.1 Protocollo Slotted ALOHA

Slotted ALOHA rappresenta un'evoluzione del protocollo citato in precedenza per far fronte alla sua bassa efficienza. In questo caso ogni stazione non trasmette più in un istante di tempo casuale ma può farlo solo in un istante di tempo multiplo del tempo di trasmissione di ciascun pacchetto T . L'algoritmo di trasmissione, eccetto ciò, è identico.

In questo caso il periodo di vulnerabilità è la metà in quanto le collisioni avvengono solo se due o più stazioni trasmettono entrambe all'inizio di ciascuno slot (non possono trasmettere prima di T istanti rispetto all'inizio di un'altra trasmissione).

9 Progetto IEEE 802

Il progetto IEEE 802 comprende una famiglia di protocolli standard per la gestione dei diversi livelli di comunicazione tra nodi.

Al livello **fisico** troviamo i vari tipi di collegamento:

- Unshielded twisted pair (UTP)
- Shielded twisted pair (STP)
- Onde radio
- Fibra ottica
- ecc.

Al livello di **accesso al mezzo** (**MAC**, media access control) troviamo

- **IEEE 802.3**: Ethernet
- **IEEE 802.5**: Token ring (ormai in disuso)
- **IEEE 802.11**: Wireless LAN
- ecc.

Al livello di **collegamento** (LLC) troviamo

- **IEEE 802.2**: Logical link control (LLC)

Lo standard **IEEE 802.1** contiene le specifiche generali del progetto 802 suddivise nelle parti di overview, architecture, bridging e management, incentrate sull'unificazione delle interfacce offerte dalle reti LAN o MAN verso il livello di rete nonostante le diverse tecnologie trasmissive.

9.1 Livello LLC

Il livello si occupa di fornire un'interfaccia unificata verso il livello di rete e si basa su un derivato del protocollo HDLC pur essendo realizzato totalmente via software. Esso offre i servizi **connectionless** con o senza riscontro e quelli **connection-oriented**.

Il PDU (Protocol data unit o pacchetto) è così formato:

- **Destination address:** 1 byte, indica l'indirizzo del destinatario
- **Source address:** 1 byte, indica l'indirizzo della sorgente
- **Control:** 1 o 2 byte, serve per le informazioni di controllo
- **Information:** dalla lunghezza variabile, contiene le informazioni trasportate dall'unità informativa.

9.2 Livello MAC

Il livello si occupa di risolvere la condivisione del mezzo trasmissivo

Il PDU è così formato:

- **Destination address:** 6 byte, indica l'indirizzo del destinatario o l'indirizzo di **multicast** (gruppi a cui appartengono più macchine contemporaneamente) o l'indirizzo di **broadcast** (tutta la rete).
- **Source address:** 6 byte, indica l'indirizzo della sorgente
- **Information:** di lunghezza variabile, incapsula la PDU del livello LLC
- **FCS:** 4 byte, posta a protezione del pacchetto (valgono le stesse considerazioni fatte per HDLC)

Gli indirizzi del livello di MAC sono unici e sono memorizzati nella ROM di ciascuna scheda di rete: i primi 3 byte rappresentano il codice del costruttore (**OUI** organization unique identifier), mentre gli altri 3 rappresentano la numerazione progressiva delle schede di rete. Le grosse aziende possiedono più di un OUI. L'indirizzo è di tipo **big-endian** (ovvero il bit più a destra è anche il più significativo).

Il metodo di accesso al mezzo utilizzato dal livello è generalmente il CSMA/CD (**carrier-sense multiple-access / collision detection**) che consiste nell'ascoltare il canale per verificare che sia libero prima di una eventuale trasmissione delle informazioni e quindi si preoccupa di gestire le collisioni, le quali si verificano quando il segnale emesso da una stazione troppo lontana non si è ancora propagato a tutto il mezzo e quindi ascoltando il canale si ha l'erronea convinzione che esso sia libero. Se durante una trasmissione avviene una collisione la stazione emette un pacchetto di jam per avvisare tutte le stazioni collegate al mezzo e le due stazioni che hanno colliso vanno in **backoff**: attendono cioè un tempo casuale prima di ritrasmettere compreso nell'intervallo $[0; KT]$ dove T è lo **slot time** e dipende dalla frequenza di cifra C del canale ed indica ogni quanto viene inviata una cifra e K è pari a $2^k - 1$, dove k è uguale al minimo tra il numero di collisioni avvenute ed un limite detto **backoff limit**. Se il canale è congestionato il sistema di backoff esponenziale consente (entro certi limiti) di abbassare la probabilità di collisione. Il CSMA/CD

possiede due diversi modi d'accesso al canale: quella "**non-persistent**" e quella "**1-persistent**".

Nella prima variante, qualora una stazione ascoltando il canale lo dovesse trovare occupato, la stazione attende un tempo casuale prima di provare a trasmettere. Nella seconda, invece, quando il canale è occupato la stazione attende che il canale si liberi e trasmette immediatamente dopo.

9.2.1 Ethernet

Il metodo di accesso utilizzato dallo standard ethernet è il cosiddetto CSMA/CD (carrier-sense multiple-access / collision detection) che consiste nell'"ascoltare" il canale per verificare che sia libero e quindi nel gestire le eventuali collisioni (è possibile che una stazione particolarmente lontana abbia iniziato a trasmettere ed il segnale non è ancora arrivato alla nuova stazione che ascolta il canale). La topologia logica usata è a bus: i dati inviati da una stazione raggiungono tutte le altre, le quali decidono se accogliere la trama o scartarla qualora non si fosse i legittimi destinatari. Ovviamente fisicamente la tipologia potrebbe essere qualunque.

Il formato della trama è leggermente diverso:

- **Preamble**: 7 byte, consente la sincronizzazione in ricezione
- **SFD**: 1 byte, è lo start frame delimiter, inviato in violazione del codice Manchester.
- **Destination address**: 6 byte, come sopra
- **Source address**: 6 byte, come sopra
- **Length**: 2 byte, contiene la lunghezza delle informazioni
- **Informazioni**: da 0 a 1500 bytes, incapsula la PDU dell'LLC
- **Pad**: da 0 a 46 bytes, garantisce la lunghezza minima della trama (ovvero 64 byte)
- **FCS**: 4 byte, frame check sequence

Il livello di MAC si occupa di garantire una spaziatura minima tra due trame successive trasmesse (**interframe gap**) e scarta automaticamente tutte le trame più piccole della lunghezza minima oltre che a gestire l'FCS e la generazione del preambolo.

I parametri Ethernet sono i seguenti:

- **Frequenza di cifra**: 10Mbps
- **Slot time**: $51.2\mu s$ (512 bps)
- **Interframe gap**: $9.6\mu s$

- **Attempt limit:** 16 (massimo numero di tentativi di ritrasmissioni dopo aver colliso)
- **Backoff limit:** 10 (numero di tentativi oltre il quale il backoff time non aumenta)
- **Jam size:** 32 bit (lunghezza del pacchetto di jam)
- **Minimum frame size:** 64 byte
- **Maximum frame size:** 1518 byte

Mediante la famiglia ethernet sono stati sviluppati diversi tipi di standard, raggruppati sotto due grandi famiglie: **mixing segment** (quando sono connesse più di due stazioni e la topologia è a bus) e **link segment** (quando le stazioni connesse sono solo due e la topologia è detta punto-a-punto).

- **10base5:** realizzato mediante cavo coassiale spesso (**thick RG213**) - Mixing segment
 - Lunghezza massima di ogni segmento: 500m (bus)
 - Lunghezza massima di ogni **drop** (o AUI - attachment unit interface, cavi utilizzati per connettere la stazione al bus): 50m
 - Distanza minima tra due stazioni (MAU, medium access unit) lungo il bus: 2.5m
 - Numero di stazioni massime per segmento: 100
- **10base2:** realizzato mediante cavo coassiale sottile (**thin RG58**) - Mixing segment
 - Lunghezza massima di ogni segmento: 185m (bus)
 - Lunghezza massima di ogni **drop** (cavi utilizzati per connettere la stazione al bus): 50m
 - Distanza minima tra due stazioni lungo il bus: 0.5m
 - Numero di stazioni massime per segmento: 30
- **10base-FP:** per tipologia a stella realizzate in fibra ottica - Mixing segment
 - **Mai realizzata!**
 - Lunghezza massima del segmento: 1Km
 - Numero di stazioni massime per segmento: 33
- **10base-T:** realizzato mediante doppino di rame (UTP).
 - Lunghezza massima di ogni segmento: 100m (bus)
- **10base-FL:** fibra ottica

- Usata per la connessione punto-punto tra due stazioni
- Lunghezza massima del segmento: 2Km (senza repeater), 1Km (con repeater)
- **10base-FB**: fibra ottica con caratteristiche di fault tolerance (tolleranza agli errori)
 - Usata per la realizzazione delle **backbone**, connette due apparati ethernet (non utenti!)
 - Lunghezza massima del segmento: 2Km (senza repeater), 1Km (con repeater)
 - La trasmissione è sincronizzata mediante un clock interno

Per esentendere la lunghezza del mezzo trasmissivo ci si avvale di apparecchi detti repeater (o hub quando i collegamenti sono più di 2) i quali si comportano in maniera passiva ripetendo il segnale in ingresso su tutti i nodi di uscita.

In generale tra due stazioni ci possono essere al massimo 5 segmenti, 4 repeater e 2 transceiver (i dispositivi che si occupano di ricevere e trasmettere i dati per ogni stazione). La su detta è nota come regola del 5-4-2. Nel caso in cui venissero utilizzati esattamente 5 segmenti e 4 ripetitori allora almeno due segmenti devono essere di tipo **link** ed all massimo 3 devono essere di tipo **mixing**.

Nelle condizioni di cui sopra, tutti i collegamenti **ottici** di tipo link non possono superare i 500 m, mentre quelli di tipo mixing i 300. Qualora i segmenti fossero solo 4, invece, i collegamenti di tipo link non possono superare i 1000m, mentre quelli di tipo mixing i 700. In quest'ultimo caso il link ottico tra stazione e ripetitore non può superare i 400m (10Base-FL) o i 300m (10Base-FP).

9.2.2 Fast ethernet

Rappresenta la naturale evoluzione di ethernet in cui però la frequenza di cifra è aumentata fino ai 100Mbps. E' possibile utilizzare degli apparati in grado di separare le stazioni che possono collidere (dividono cioè il **dominio di collisione**) chiamati **switch** (si tratta di hub "intelligenti" che non ripetono il segnale a chiunque ma soltanto alla stazione\segmento corretto).

I parametri Fast ethernet sono i seguenti:

- **Frequenza di cifra**: 100Mbps
- **Slot time**: $5.12\mu s$ (512 bps)
- **Interframe gap**: $0.96\mu s$

Lo standard mantiene la dimensione massima e minima di ethernet e ciò significa che le reti sono 10 volte più piccole.

Esistono 4 sotto standard sviluppati per i vari mezzi fisici utilizzati:

- **100base-TX**: realizzato mediante 2 doppini UTP o STP con connettori RJ45, usa la codifica a blocco 4B5B a 125Mbaud (125M simboli al secondo in cui ogni simbolo è in grado di rappresentare anche più di un bit). La distanza massima di ogni segmento deve essere di 100m.
- **100base-FX**: realizzato con 2 fibre ottiche multimodali, usa la codifica 4B5B per il blocco e NRZI per la codifica di linea. Ogni segmento può raggiungere i 2Km in full duplex (ma solo 100m in half duplex).
- **100base-T4**: realizzato con 4 doppini UTP con connettori RJ45, usa la codifica 8B6T a 25Mbaud e ogni segmento può raggiungere i 100m (half duplex soltanto).
- **100base-T2**: realizzato con 2 doppini UTP e connettori RJ45, ogni segmento half duplex può raggiungere i 100m. La velocità è 25Mbaud.

9.2.3 Gigabit ethernet

Un'ulteriore evoluzione di ethernet moltiplica fino a 100 volte la velocità originale dello standard (1Gbps).

Giacchè aumentare di 100 volta la velocità di una rete mantenendo la dimensione della PDU uguale significa ridurre di 100 volte le dimensioni massime di una rete, lo standard ridefinisce una nuova dimensione minima della PDU: 512 byte. La nuova PDU è identica a quella di ethernet ma alla fine aggiunge un campo **extension** di dimensione variabile tra i 0 e i 448 byte.

Lo standard permette anche il **frame bursting**, ovvero consente la trasmissione di più trame consecutive separandole da un IFG di durata minima di 96ns. E' opportuno considerare che la prima trama possiede il campo **extension** ma quest'ultimo non risulta più necessario per le altre trame. In ogni caso il burst non può superare gli 8192 byte.

Gli standard sviluppati mediante gigabit ethernet sono:

- **1000base-CX**: realizzato mediante 2 doppini STBP dalla dimensione massima di 25 m con codifica a blocco 8B10B a 1.25 GBaud e codifica di linea NRZ.
- **1000base-T**: realizzato con 4 doppini UTP dalla dimensione massima di 100m.

9.2.4 10 Gigabit ethernet

Rappresenta l'evoluzione di gigabit ethernet realizzata unicamente in fibra ottica alla frequenza di cifra di 10Gbps:

- **10gbase-SR**: 300m di fibra multimodo
- **10gbase-LR**: 10Km di fibra monomodo
- **10gbase-ER**: 40Km di fibra monomodo
- **10gbase-LX4**: 300m di fibra multimodo e 10Km di fibra monomodo
- **10gbase-SW**: 300m di fibra multimodo
- **10gbase-LW**: 10Km di fibra monomodo
- **10gbase-EW**: 40Km di fibra monomodo

10 Prestazioni

Le prestazioni di una rete si basano su tre metriche principali:

- **Throughput** (traffico smaltito in un'unità di tempo)
- Π_p , **Loss probability** (probabilità che un pacchetto venga perso)
- **Average transfer delay** (tempo che impiega un pacchetto ad attraversare la rete)

I parametri che verranno utilizzati sono:

- **N**: numero di stazioni sulla rete
- λ : frequenza di arrivo dei pacchetti in ogni stazione[Hz]
- **L**: lunghezza dei pacchetti
- **C**: capacità del canale

10.1 Calcolo del throughput

- **Traffico totale sulla rete**: $A_0 = N \cdot \lambda \cdot L$ (supponendo che sia Π_p nulla)
- **Traffico normalizzato (efficienza)**: $S = N \cdot \lambda \cdot \frac{L}{C}$
- **Throughput**: $THR = S \cdot C$ [bps]

10.2 Efficienza di ALHOA

- $S = Ge^{-2G}$ dove G è il numero medio di tentativi di trasmissione da parte delle stazioni in un tempo uguale al tempo di trasmissione T di un pacchetto.
- $S_{max} = \frac{1}{2e} = 0.184$ (per G=0.5)

10.3 Efficienza di Slotted ALHOA

- $S = Ge^{-G}$
- $S_{max} = \frac{1}{e} = 0.368$ (per $G=1$)

11 NScript e Nam

Il software **nscript** è un editor grafico per la creazione di reti, le quali sono generate mediante un linguaggio di scripting detto **tcl**. I file contenenti tale codice hanno estensione **.tcl**. Il software **ns** prende in input tali file e ne esporta la relativa simulazione mediante file **.nam**. Il software **nam**, a sua volta, accetta in input tali file e consente di effettuare un'animazione grafica della simulazione di cui sopra.

Una simulazione tcl inizia sempre con la riga:

```
set nome_simulazione [new Simulator]
```

Per comodità nome_simulazione sarà abbreviato come ns.

Una simulazione tcl termina sempre con la riga:

```
$nome_simulazione run
```

11.1 Nodi

I nodi rappresentano i punti di intersezione tra più collegamenti all'interno di una rete. Graficamente sono rappresentati mediante dei cerchi.

Sintassi tcl: **set nome_nodo [\$ns node]**

11.2 Link

Un link rappresenta un collegamento fisico tra due **nodi** distinti. A ciascun link è assegnata una **capacità** (il software accetta capacità in Mb anzichè in Mbps!), un **intervallo di propagazione** (il tempo che ci impiega un pacchetto per transitare sulla rete) ed una **modalità di buffering** che sostanzialmente indica come si comportano i pacchetti sul link (la modalità di base è **DropTail** ed indica che superato il limite del buffer del **nodo** i restanti pacchetti vengono distrutti).

Sintassi collegamento duplex: **set duplex_link \$nome_nodo_1 \$nome_nodo_2 [capacità] [intervallo_di_propagazione] DropTail**

Es: **set duplex_link \$node0 \$node1 10Mb 200ms DropTail**

Graficamente sono rappresentati da dei tratti spessi che collegano due nodi.

E' possibile impostare la dimensione massima della coda su un link mediante la seguente sintassi:

```
$ns queue-limit $nome_nodo_0 $nome_nodo_1 [limite]
```

11.3 Agent

Gli agenti rappresentano il comportamento del livello di rete. Una volta creato l'agent è necessario collegarlo ad un nodo mediante la sintassi:

```
$ns attach-agent $nome_nodo $nome_agent
```

E' necessario connettere tra di loro due agent differenti affinché la trasmissione abbia successo:

```
Sintassi: $ns connect $nome_agent_0 $nome_agent_1
```

Graficamente sono rappresentati da un quadrato diviso a metà mediante una linea orizzontale. Il collegamento agent-node è rappresentato mediante una linea, la connessione agent-agent mediante una linea puntinata.

11.3.1 UDP agent

Sono gli agenti responsabili della commutazione di pacchetto all'interno della rete. Vengono usati generalmente per tutti quei nodi attraverso i quali viene inviato del traffico all'esterno.

Sintassi: **set nome_agent [new Agent/UDP]**

11.3.2 NULL agent

Sono degli agenti che vengono usati in ricezione il cui utilizzo è solo quello di accogliere i pacchetti e di distruggerli.

Sintassi: **set nome_agent [new Agent/NULL]**

11.3.3 Loss monitor agent

Sono equivalenti ai NULL agent ma in questo caso permettono la raccolta di informazioni circa i pacchetti per fini statistici.

Sintassi: **set nome_agent [new Agent/LossMonitor]**

11.4 Sorgenti di traffico

Le sorgenti di traffico rappresentano le stazioni connesse alla rete e sono rappresentate graficamente mediante dei rombi.

Le sorgenti di traffico vanno associate agli agent mediante la sintassi:

\$nome_sorgente attach-agent \$nome_agent

11.4.1 CBR

Una CBR è definita come una sorgente che trasmette pacchetti ad un ritmo costante ν [pkts/s] (ovvero un pacchetto ogni $1/\nu$ secondi). La velocità di trasmissione è in genere costante e pari alla capacità del canale, anche se non è detto che la lunghezza dei pacchetti sia la stessa (con l'unico vincolo che il tempo di trasmissione del pacchetto deve essere minore di $1/\nu$). Se la lunghezza dei pacchetti rimane la stessa allora una CBR di questo tipo è uguale ad una VBR periodica.

Sintassi: **set nome_sorgente [new Application/Traffic/CBR]**

E' possibile impostare 2 dei 3 diversi parametri della sorgente (il terzo è calcolato di conseguenza):

- **Dimensione pacchetti:** *\$nome_sorgente set packetSize_ [dimensione]*
- **Velocità di picco:** *\$nome_sorgente set rate_ [rate]*
- **Intervallo di trasmissione:** *\$nome_sorgente set interval_ [intervallo]*.
(l'intervallo è dato da $interval = \frac{dimensione \cdot 8bit}{rate}$)

11.4.2 ExpOnOff

Rappresenta la sorgente statistica esponenziale (vedi primo capitolo).

Sintassi: **set** nome_sorgente [**new Application/Traffic/ExpOnOff**]

I parametri personalizzabili sono:

- **Dimensione pacchetti:** \$nome_sorgente **set packetSize** _ [dimensione]
- **Velocità di picco:** \$nome_sorgente **set rate** _ [rate]
- **T_on:** \$nome_sorgente **set burst_time** [secondi]
- **T_off:** \$nome_sorgente **set idle_time** [secondi]

11.4.3 Poisson

Rappresenta la sorgente statistica poissoniana (in realtà si tratta di un workaround a causa delle limitazioni del simulatore).

Sintassi: **set** nome_sorgente [**new Application/Traffic/Exponential**]

\$nome_sorgente **set rate** _ 100000Mb

\$nome_sorgente **set burst_time** 0s

I parametri personalizzabili sono:

- **Dimensione pacchetti:** \$nome_sorgente **set packetSize** _ [dimensione]

Fondamentalmente si alza il valore di picco ad un valore altissimo e si riduce al minimo il valore di t_on

11.4.4 VBR

Una VBR è una sorgente che nei periodi di ON si comporta come una CBR, mentre nei periodi di OFF rimane inattiva.

11.5 Eventi

Gli eventi determinano gli istanti di avvio e di arresto di tutte le sorgenti presenti sulla rete. Graficamente sono rappresentati da dei cronometri collegati alle relative sorgenti.

- **Inizio trasmissione:** \$ns at [tempo] "\$nome_sorgente start"
- **Fine trasmissione:** \$ns at [tempo] "\$nome_sorgente stop"
- **Fine simulazione:** \$ns at [tempo] "finish"