

Reti di telecomunicazioni

sav

2025

Contents

0.1	Strategie di controllo errore	1
0.1.1	Protocolli ARQ (Automatic Repeat-reQuest)	1
0.1.2	Efficienza di stop and wait in assenza di errori	3
0.1.3	Efficienza di stop and wait in presenza di errori	4
0.1.4	Protocolli Continuous ARQ	7
0.1.5	Dimensione delle finestre di trasmissione/ricezione	12
0.1.6	Efficienza GBN & SR	12
0.1.7	Efficienza con riscontri fuori dalla finestra di trasmissione W_s	15
0.1.8	Confronto tra protocolli di linea	16
0.2	Livello 2 Frame DHLC	17
0.2.1	Struttura del frame	17
0.2.2	Bit stuffing	17
0.2.3	Controllo degli errori casuali con CRC	18
0.2.4	Protocollo point to point (PPP)	20
0.3	Protocolli di accesso	22
0.3.1	Troughput di rete	23
0.3.2	Protocollo ALOHA puro	24
0.3.3	Protocollo ALOHA slotted - prestazioni	27
0.3.4	Protocolli Carrier Sense Multiple Access - CSMA (rilevazione canale)	28

0.1 Strategie di controllo errore

I pacchetti di livello 2 sono chiamati frame/trame, il PCI della PDU(della trama) è diviso in header e trailer; nel trailer sono presenti i campi utili alle strategie di controllo degli errori.

È possibile distinguere due tipi principali di strategie:

1. FEC (Forward Error Correction): il mittente invia dati ridondanti, in modo che il destinatario possa correggere gli errori senza richiedere una ritrasmissione.
2. ARQ (Automatic Repeat-reQuest): il mittente invia i dati e il destinatario richiede la ritrasmissione dei dati corrotti.

0.1.1 Protocolli ARQ (Automatic Repeat-reQuest)

Nei sistemi di telecomunicazioni, Automatic Repeat-reQuest è una strategia di controllo di errore, che svolge il compito di rivelare un errore (ma non di correggerlo). I pacchetti corrotti vengono scartati e viene richiesta la loro ritrasmissione. I protocolli di tipo ARQ più comuni sono:

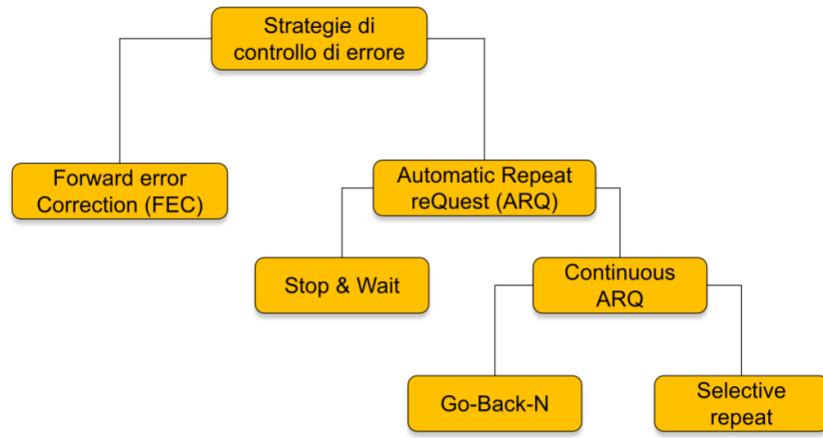
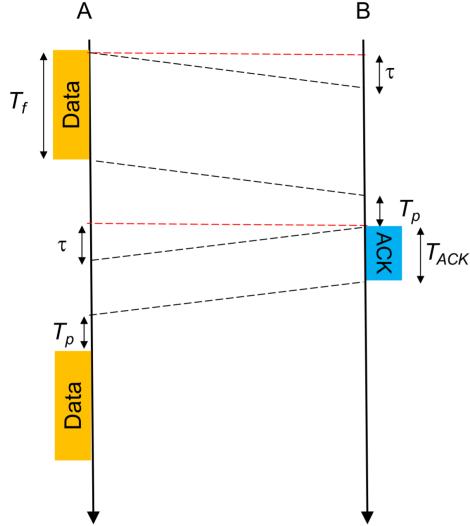


Figure 1: Strategie di controllo degli errori: confronto tra FEC e ARQ

- Stop-and-Wait
- Go-Back-N
- Selective Repeat

Stop-and-Wait ($W_S = W_R = 1$)

Lo stop-and wait non è una forma di Continuous ARQ, per cui è NON è ammessa la trasmissione continua di frame numerati. È una tecnica semplice e "discontinua": il mittente trasmette un solo frame alla volta e attende l'ACK prima di inviarne un altro; i riscontri sono cumulativi.



Il protocollo Stop-and-Wait prevede che il mittente attenda il riscontro del destinatario per ogni frame inviato, prima di procedere con il frame successivo.

Il mittente invia una trama, quindi si ferma (stop) e attende (wait) il riscontro (Ack) da parte del ricevente prima di trasmettere una nuova trama.

- T_f : tempo di trasmissione di una trama [s]
- T_{ACK} : tempo di trasmissione di un riscontro [s]
- T_p : tempo di elaborazione di una trama [s]
- τ : tempo di propagazione [s]

Figure 2: Schema Stop-and-Wait

0.1.2 Efficienza di stop and wait in assenza di errori

Prima di definire l'efficienza del protocollo, è necessario definire il tempo necessario a trasmettere una trama:

$$T_{tot} = T_f + T_{ACK} + 2T_p + 2\tau \quad (1)$$

T_p come già detto è il tempo di elaborazione della trama, dipende dalla velocità di elaborazione e quindi non dovrebbe essere precisamente uguale in trasmissione e ricezione, ma per semplicità lo consideriamo uguale. Inoltre T_f , il tempo di trasmissione della trama, è anche definito come il tempo utile, poiché tutti gli altri tempi che consideriamo sono relativi a operazioni di controllo, elaborazione e invio del pacchetto. L'efficienza (η) è perciò definita come:

$$\eta = \frac{T_f}{T_{tot}} = \frac{T_f}{T_f + T_{ACK} + 2T_p + 2\tau} \quad (2)$$

Si possono effettuare alcune semplificazioni, considerando che T_{ACK} e T_p sono molto più piccoli di T_f e τ , quindi possiamo considerare:

$$T_{tot} = T_f + 2\tau \quad (3)$$

da cui si ottiene l'efficienza:

$$\eta = \frac{T_f}{T_f + 2\tau} = \frac{1}{1 + 2\frac{\tau}{T_f}} = \frac{1}{1 + 2\alpha} \quad (4)$$

dove $\alpha = \frac{\tau}{T_f}$ è il rapporto di propagazione normalizzato

- il throughput è pari al prodotto tra la frequenza di cifra C (capacità del canale) e l'efficienza: ηC
- anche in assenza di errori, l'efficienza non è mai pari a 1.

0.1.3 Efficienza di stop and wait in presenza di errori

I problemi che possono verificarsi sono:

- il frame o l'ACK arrivano errati
- il frame non arriva

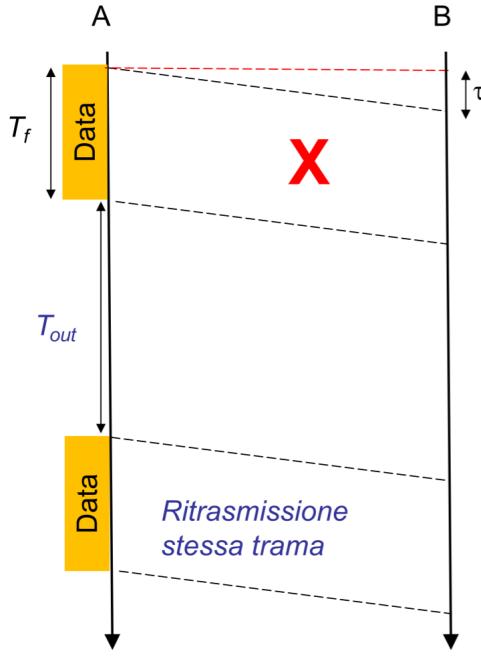


Figure 3: Errore nel frame: quando si verifica un errore nel frame, il mittente si accorge del problema grazie alla scadenza del timer(no riscontro) avviato durante la trasmissione. In tal caso, il frame viene ritrasmesso. Questo processo garantisce che i dati vengano ricevuti correttamente, ma può ridurre l'efficienza del protocollo a causa delle ritrasmissioni necessarie.

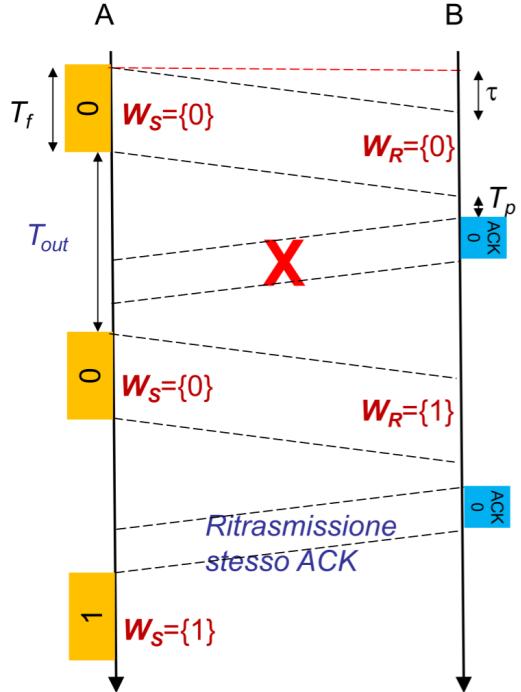


Figure 4: Errore nell'ACK: quando si verifica un errore nell'ACK, B pensa di aver inviato un ACK correttamente, perciò fa scorrere la finestra; ma il mittente non riceve il riscontro atteso entro il tempo stabilito dal timer. Perciò A invierà nuovamente il frame, che però risulterà una copia agli occhi di B, che riscontrerà nuovamente la stessa trama. Anche in questo caso, il frame viene ritrasmesso.

Come calcolo il T_{out} (time che avvio nel momento nella trasmissione del frame)?

Deve valere almeno il tempo di invio del ACK(T_{ACK}), più il tempo di propagazione(andata e ritorno) più il tempo di elaborazione del frame(T_p , in ricezione e trasmissione):

$$T_{out} \geq T_{ACK} + 2\tau + 2T_p \quad (5)$$

Il numero di frame trasmessi per la corretta ricezione del frame è detto N_S ed è in media considerabile come una variabile aleatoria;

inoltre il numero totale di frame ritrasmessi(N_R) è dato dal numero di frame trasmessi meno il frame inviato con successo:

$$N_R = N_S - 1 \quad (6)$$

Per calcolare il tempo totale necessario per la trasmissione si deve considerare che per ogni ritrasmessione si perde un tempo pari a $T_f + T_{out}$ cui si deve aggiungere il tempo necessario per la trasmissione che avviene con successo:

$$T_{tot} = (N_S - 1)(T_f + T_{out}) + T_f + T_{ACK} + 2T_p + 2\tau \quad (7)$$

Calcolo di $\overline{N_S}$ la variabile N_S è aleatoria, in quanto dipende dagli errori sul canale di trasmissione; si calcola l'efficienza media del protocollo SW in presenza di errori considerando il valor medio di N_S

$$\begin{aligned} N_S = 1 & \quad pr\{N_S = 1\} = 1 - P \\ N_S = 2 & \quad pr\{N_S = 2\} = P(1 - P) \\ N_S = 3 & \quad pr\{N_S = 3\} = P^2(1 - P) \\ \vdots & \quad \vdots \\ N_S = i & \quad pr\{N_S = i\} = P^{i-1}(1 - P) \end{aligned}$$

Figure 5: Probabilità di successo e insuccesso nella trasmissione di un frame in Stop-and-Wait

Definiamo il frame error rate (FER) come la probabilità che un frame venga ricevuto con errore, che sarà il valore di P nel calcolo successivo. Se P è la probabilità che un frame sia errato, la probabilità che un frame sia trasmesso correttamente è $1 - P$.

La variabile aleatoria N_S rappresenta il numero di tentativi necessari affinché un frame sia ricevuto correttamente.

La sua distribuzione è di tipo geometrico:

$$P(N_S = k) = P^{k-1}(1 - P)$$

dove k è il numero di tentativi (ritrasmissioni più il primo invio). Il valor medio di N_S è:

$$\overline{N_S} = \frac{1}{1 - P}$$

$$\overline{N_S} = E[N_S] = \sum_{i=1}^{+\infty} iP^{i-1}(1 - P) = \frac{1}{(1 - P)^2}(1 - P) = \frac{1}{1 - P}$$

Spiegazione La formula sopra mostra il calcolo del valor medio $\overline{N_S}$ per una variabile aleatoria geometrica, che rappresenta il numero medio di tentativi necessari affinché un frame venga trasmesso correttamente.

La distribuzione geometrica è appropriata perché ogni trasmissione è indipendente e la probabilità di successo (ricezione corretta) è costante e pari a $1 - P$, dove P è la probabilità di errore del frame. La somma $\sum_{i=1}^{+\infty} iP^{i-1}(1 - P)$ rappresenta la media pesata di tutti i possibili numeri di tentativi, ciascuno moltiplicato per la probabilità che siano necessari esattamente i tentativi.

Sviluppando la somma, si ottiene che il valor medio è $\frac{1}{1-P}$: questo significa che, in media, il numero di tentativi cresce all'aumentare della probabilità di errore P . Se P è vicino a zero, bastano pochi tentativi; se P aumenta, il numero medio di ritrasmissioni cresce rapidamente.

Perciò, ripetendo, se si vuole ottenere il valore medio di una variabile aleatoria bisogna effettuare uno studio probabilistico; questa probabilità sarà in media pari alla moltiplicazione tra due probabilità:

- probabilità di successo ($1-P$): frame trasmesso correttamente
- P^{i-1} rappresenta la sequenza di fallimenti che avvengono prima della trasmissione avvenuta con successo, che calcolo per un valore di i infinito (serie geometrica), caso peggiore (benchè irreale)

$$\Pr(N_S = i) = \underbrace{P^{i-1}}_{\# \text{ fallimenti}} \cdot \underbrace{(1 - P)}_{\text{successo finale}}$$

La distribuzione di probabilità è:

$$\Pr(N_S = i) = P^{i-1}(1 - P)$$

Il valore atteso si ottiene come:

$$E[N_S] = \sum_{i=1}^{\infty} i \cdot \Pr(N_S = i)$$

Errore sul bit ed errore sul frame

Frame error rate (FER) è possibile definire le probabilità per le quali ci sia un errore sui bit(BER, bit error rate) all'interno di un range oppure, di maggiore interesse a questo livello di rete, la probabilità di errore su un range di frame, quindi il FER(frame error rate).

Per calcolarlo ci interessa definire:

- L_f : lunghezza in bit del frame
- L_{ACK} : lunghezza in bit del ACK

e sarà pari al valore P utilizzato nel calcolo probabilistico precedente, perciò:

$$P = FER = 1 - (1 - p)^{(L_f + L_{ACK})} \quad (8)$$

Calcolo dell'efficienza

L'efficienza η del protocollo Stop and wait è data da:

L'efficienza η del protocollo Stop-and-Wait in presenza di errori si calcola come rapporto tra il tempo utile di trasmissione e il tempo totale medio necessario per trasmettere correttamente un frame, considerando anche le ritrasmissioni dovute agli errori:

$$\eta = \frac{T_f}{T_{tot}} \quad (9)$$

dove:

- T_f è il tempo di trasmissione del frame (dato utile)
- T_{tot} è il tempo totale per ogni tentativo di trasmissione (inclusi tempi di ACK, propagazione, elaborazione)

ricordando la formula di T_{tot}

$$T_{tot} = (N_S - 1)(T_f + T_{out}) + T_f + T_{ACK} + 2T_P + 2\tau \quad (10)$$

è possibile effettuare qualche semplificazione, trascurando termini piccoli (come T_{ACK}, T_P) rispetto al tempo di trasmissione; considerando inoltre il tempo minimo per T_{out} (pari a 2τ) (timer pari al tempo di propagazione di andata/ritorno del frame) per cui:

$$\eta \simeq \frac{T_f}{(N_S - 1)(T_f + 2\tau) + T_f + 2\tau} = \frac{T_f}{N_S(T_f + 2\tau)} = \frac{(1 - P)T_f}{T_f + 2\tau} \quad (11)$$

ricordando il rapporto di propagazione normalizzato α ($\alpha = \frac{\tau}{T_f}$), ottengo che l'efficienza:

$$\eta = \frac{1 - P}{1 + 2\alpha} \quad (12)$$

Dipendenza dell'efficienza dal ritardo di propagazione normalizzato Fissato il FER (dipendente dalla rumorosità del canale), l'efficienza dipende dal valore del ritardo di propagazione normalizzato

Considerando la distanza d e la velocità di propagazione v delle onde elettromagnetiche nel mezzo:

$$\alpha = \frac{\tau}{T_f} = \frac{d/v}{L_f/C} = \frac{dC}{vL_f}$$

Il protocollo Stop-and-Wait è efficiente (quasi $1 - P$) solo per valori piccoli di α , cioè:

- canali lenti (C basso)
- collegamenti a distanze ridotte (d basso)
- trame sufficientemente grandi (L_f alto, ma non troppo per non aumentare P)

0.1.4 Protocolli Continuous ARQ

I protocolli continuous ARQ, a differenza dello stop-and-wait, prevede l'invio di un'insieme di pacchetti(frame/trame) numerati, tramite sliding window. Vengono definite:

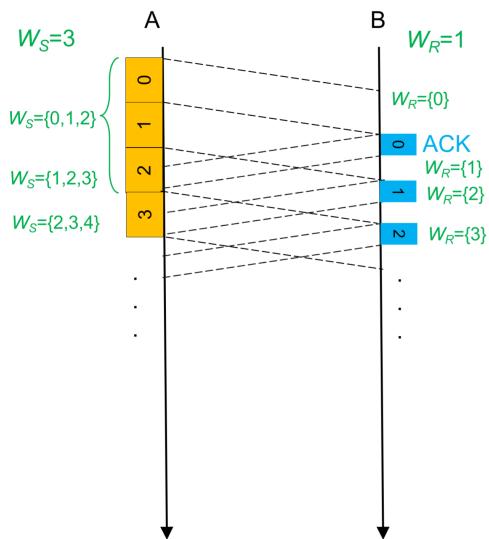
- finestra di trasmissione W_s : grandezza della finestra di trasmissione, perciò quanti pacchetti posso inviare "consecutivamente"(on the fly, in attesa di riscontro) e da cui posso aspettarmi un riscontro (ACK) cumulativo
- finestra di ricezione W_R : definisce la grandezza della finestra di ricezione,

Se il campo dedicato alla numerazione è costituito da b bit, si potranno numerare in modo differente fino a N_b trame.

Protocollo Go-Back-N ($W_s > 1$ e $W_R = 1$)

Senza errori - efficienza massima Il protocollo Go-Back-N è un protocollo di tipo Continuous ARQ, che consente la trasmissione continua di più frame numerati. In questo caso specifico, i segnali di riscontro sono cumulativi.

Nel caso in cui ci siano errori allora questo protocollo prevede di ritornare indietro nel momento della trasmissione in cui c'è stato l'errore(alla trama N), perciò go back N.



Nel seguente esempio la finestra di trasmissione è 3 e quella di ricezione 1; vengono riscontrati gli ack in modo corretto nei tempi previsti dalla finestra, che nel frattempo ad ogni ack corretto ricevuto scorre di posizione.

Questo vuol dire che nel caso in cui non ci siano errori con l'invio delle trame e con il riscontro degli ack, il seguente protocollo ha efficienza massima.

Figure 6: Go-Back-N senza errori

Go back N - con errori

Si possono riscontrare errori durante la ricezione del pacchetto e durante il riscontro degli ack; a seconda della tipologia dell'errore avremo soluzioni differenti.

Errore trasmissione del frame utilizzo del negative ACK (NACK)

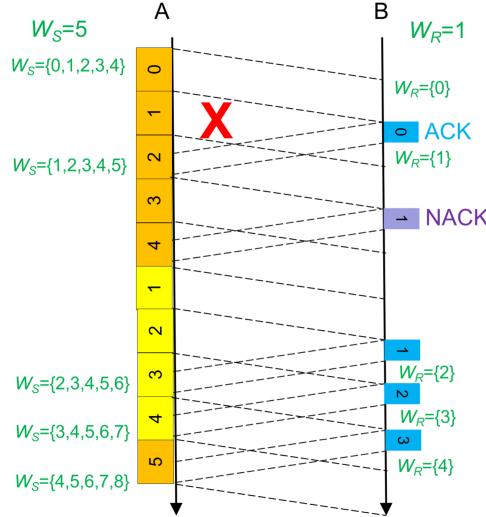


Figure 7: Go-Back-N con errori in trasmissione

Quando si verifica un errore in un frame, il destinatario scarta il frame errato e tutti i frame successivi.

Il mittente, una volta rilevato l'errore (tramite la ricezione di un NACK (negative ACK) inviato dal destinatario), ritrasmette il frame errato e tutti i frame successivi.

Errore ACK non ricevuto (T_{out}) caso in cui l'ACK non venga inviato o si sia perso

L'unico modo con il quale il mittente può evitare che il sistema si blocchi in caso di mancato riscontro (neanche NACK), è quello di avviare un timer ad ogni frame inviato, entro il quale va ricevuto un riscontro.

Se questo timer (T_{out}) scade allora la finestra torna indietro e ritrasmette da dove è stato riscontrato il problema.

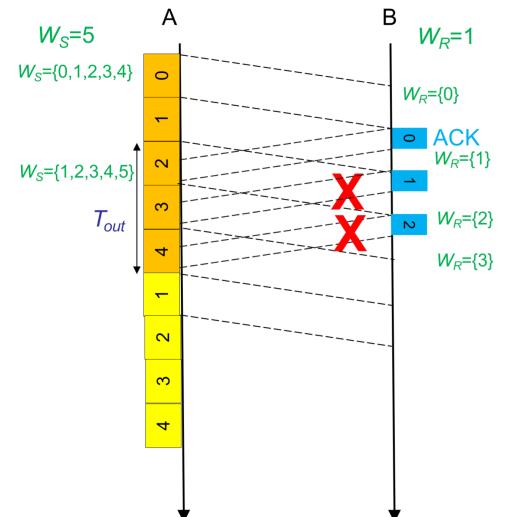


Figure 8: Go-Back-N con errore nell'ACK

Go back N - ACK equivocato(T_{out})

con conseguente ricezione di duplicati

Analisi del problema Quando si verifica un equivoco nell'ACK, ossia viene riscontrato erroneamente un frame, il mittente interpreta erroneamente il riscontro ricevuto e potrebbe ritrasmettere frame già ricevuti correttamente dal destinatario. Questo equivoco può portare a un aumento delle ritrasmissioni e a una riduzione dell'efficienza del protocollo. L'unico ACK corretto è l'ACK-0, quelli successivi sono errati; nel frattempo la finestra di ricezione ha continuato a scorrere, richiedendo i nuovi 1, 2, etc... Il problema è che i pacchetti 1, 2, 3 sono già stati ricevuti, l'errore c'è stato solo negli ACK, per i quali il trasmittente sta reinviando 1, 2, 3 precedenti, che però risulteranno come una copia (il ricevente non lo sa) di quelli trasmessi correttamente prima ma riscontrati in modo errato.

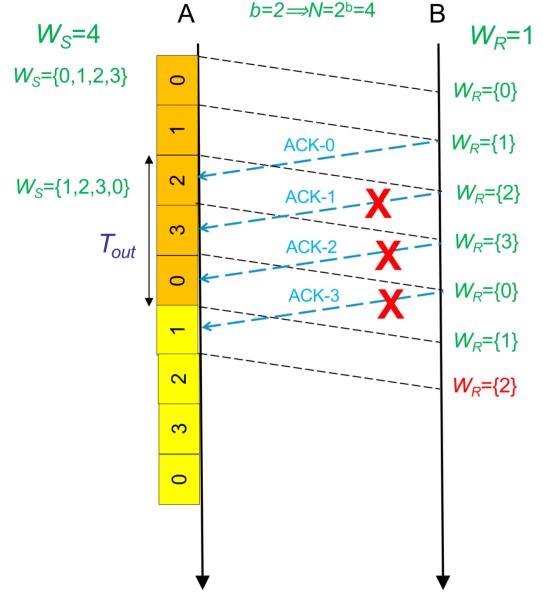


Figure 9: Go-Back-N con equivoco nell'ACK

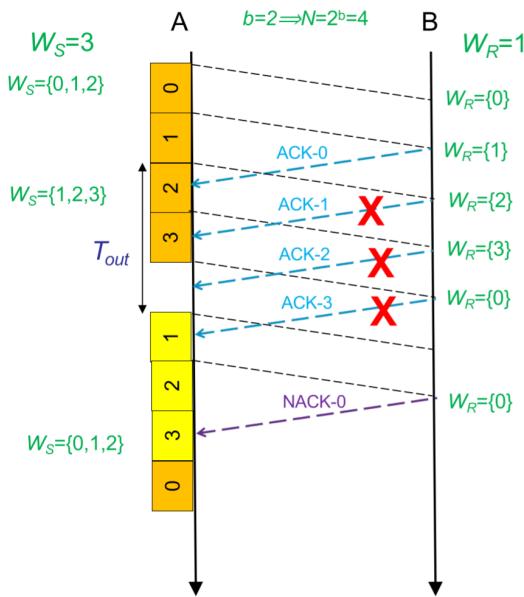


Figure 10: Go-Back-N con equivoco nell'ACK

Soluzione In questo esempio l'unico frame riscontrato correttamente è 0, come nel caso precedente; il problema dei duplicati si risolve poiché il primo frame che non ha avuto un riscontro corretto, ACK-1, ha avviato un timer entro il quale desidera ricevere un riscontro. Al termine di quest'ultimo il trasmittitore ritrasmette dal primo frame che non ha ricevuto riscontro, quindi da 1.

Il ricevente riceve 1, ma la sua finestra intatto ha slideato come se non ci fossero problemi fino a 0; quindi il ricevente si accorge del problema poiché riceve 1 al posto di 0; a questo punto invia un NACK-0 per dire al trasmittitore che quelli ricevuti precedentemente erano corretti, ma riscontrati erroneamente, quindi desidera continuare la comunicazione dal frame 0, senza ricevere duplicati. ($W_S + W_R \leq N$) Questo è possibile perché la finestra di trasmissione è grande $N-1$, dove N è il numero di frame.

Protocollo selective repeat

Il protocollo Selective Repeat è un protocollo di tipo Continuous ARQ che consente la ritrasmissione selettiva dei soli frame ricevuti con errore, evitando di ritrasmettere quelli ricevuti correttamente.

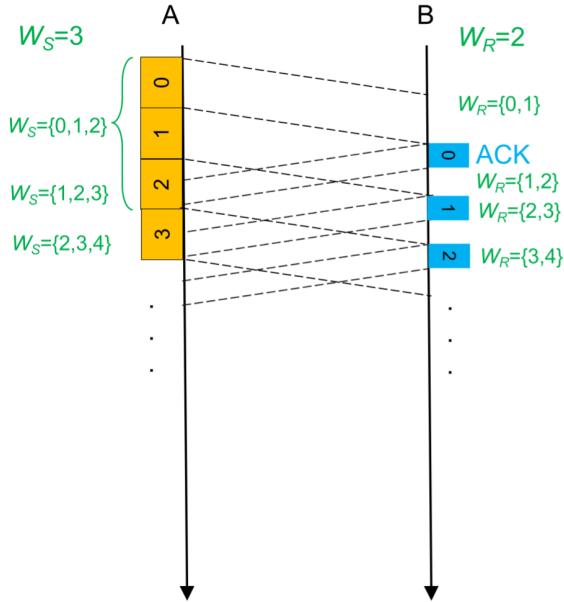


Figure 11: Selective Repeat

In assenza di errori Questo approccio migliora l'efficienza rispetto al Go-Back-N, poiché la finestra di ricezione può essere maggiore di 1 ($W_R > 1$).

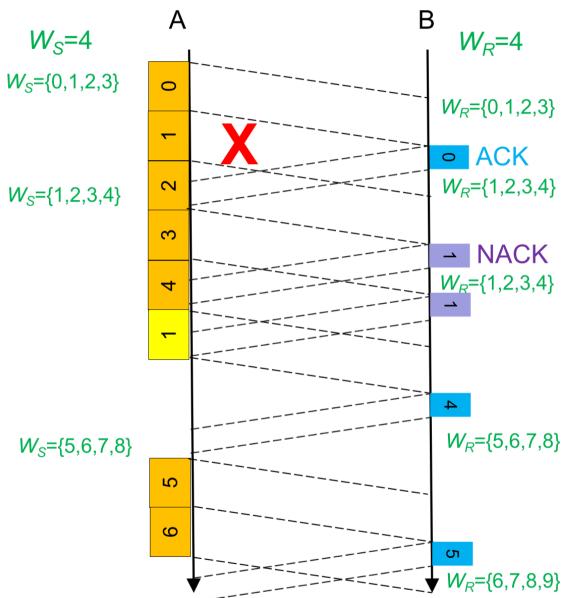


Figure 12: Selective Repeat con errori sulla trama

In presenza di errori sul frame Quando si verifica un errore in una trama, il protocollo Selective Repeat consente di ritrasmettere solo la trama errata (il ricevente invia un NACK in cui indica quale frame è andato perso), senza dover ritrasmettere tutte le trame successive come nel Go-Back-N. Questo approccio riduce il numero di ritrasmissioni necessarie e migliora l'efficienza complessiva del protocollo, specialmente in presenza di errori frequenti.

Selective repeat - equivocazione dei riscontri e soluzione (T_{out})

Problema equivocazione ACK Il problema sussiste quando i frame vengono trasmessi correttamente, la finestra di ricezione scorre correttamente, ma gli ACK inviati sono errati.

Perciò come avveniva erroneamente negli altri protocolli, vengono ricevuti dei frame duplicati perché il trasmettitore riceve degli ACK errati e quindi invia nuovamente i frame rispettivi (alla fine del T_{out}) a questi ACK; ma questi frame risulteranno a tutti gli effetti dei duplicati di quelli ricevuti correttamente dal ricevitore.

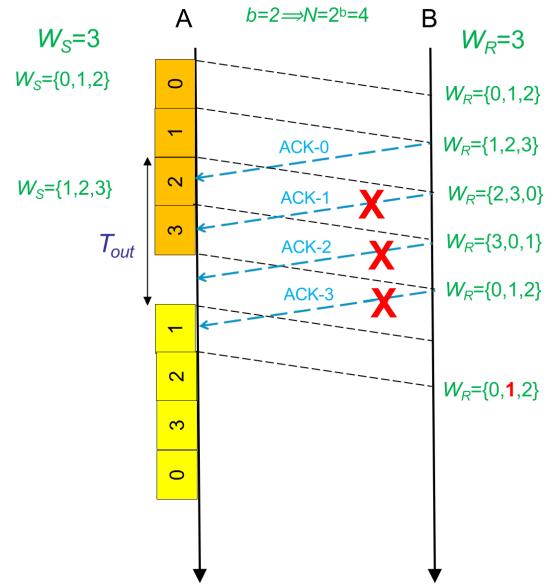


Figure 13: Selective Repeat con errori sull'ACK

Soluzione Questo problema viene risolto diminuendo la grandezza delle finestre ($W_S + W_R \leq N$), secondo i criteri discussi nel capoverso successivo. Ritrasmettendo solo i frame degli ACK riscontrati in modo errato, quindi non inviando nuovamente la tutta la finestra.

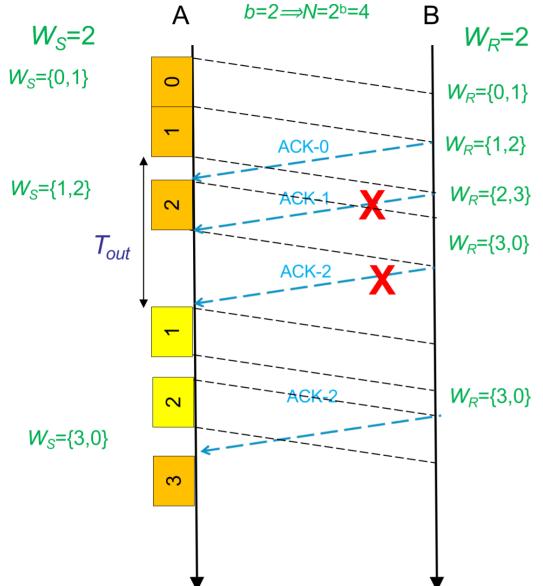


Figure 14: Selective Repeat con equivocazione risolta

0.1.5 Dimensione delle finestre di trasmissione/ricezione

Con la visione degli esempi riguardanti l'equivocazione, si può affermare che in entrambi i protocolli continuous ARQ, Go-Back-N e Selective Repeat, le finestre di trasmissione e di ricezione non possono essere pari a o maggiori di N in quanto si avrebbe equivocazione:

$$\begin{cases} W_S \leq N \\ W_R \leq N \end{cases} \quad W_S + W_R \leq N$$

Inoltre le finestre non devono sovrapporsi, poiché in caso di equivoco e di scorrimento completo della finestra di ricezione, si potrebbe riscontrare il problema di ricezione di duplicati; anche questo si risolve imponendo una condizione sulla grandezza di tali finestre:

Dimensione finestre GBN ed efficienza massima

Nel caso di GBN, la finestra di ricezione è necessariamente unitaria. Per cui la finestra di trasmissione deve essere al più N-1; poiché non conviene fissarla ad un valore inferiore al massimo (riduzione del throughput non giustificata), la scelta ottimale è fissare:

$$\begin{cases} W_S = N - 1 \\ W_R = 1 \end{cases}$$

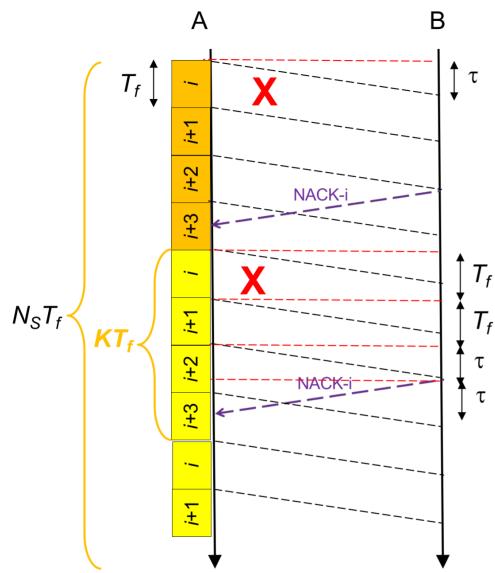
Dimensione finestre SR ed efficienza massima

Nel caso del selective repeat vanno considerati più fattori:

- non ha senso scegliere la finestra di trasmissione maggiore di quella di ricezione, perché si trasmetterebbero più trame di quante ne può ricevere e ordinare il ricevente
- non ha senso scegliere la finestra di ricezione maggiore di quella di trasmissione perché si ridurrebbe il throughput trasmettendo meno trame di quelle che si possono ricevere

Pertanto la soluzione ottimale è: $W_S = W_R = \frac{N}{2}$

0.1.6 Efficienza GBN & SR



Efficienza Go-Back-N Il trasmittitore si accorge dell'errore sulla trama i solo dopo $i+x$, dove x dipende dalla finestra di trasmissione e dal tempo in cui impiega ad arrivare il NACK- i relativo al frame errato.

Il costo della trasmissione del frame corretto è $N_S T_f$, dove N_S è il numero di frame che sono stati trasmessi e ritrasmessi per la corretta trasmissione di quel frame, pari a:

$$N_S = ("numero_errori" * K) + 1 \quad (13)$$

K è il numero di trame che vengono ritrasmesse ad ogni errore, il + 1 il frame corretto.

Figure 15: Efficienza del protocollo Go-Back-N

Considerando il valore medio delle trame trasmesse in totale, N_S , il tempo totale equivale a:

$$T_{tot} = \overline{N_S} * T_f \quad (14)$$

da cui l'efficienza η è pari al reciproco di $\overline{N_S}$:

$$\eta = \frac{T_f}{\overline{N_S} \cdot T_f} = \frac{1}{\overline{N_S}} \quad (15)$$

$$\begin{array}{ll} N_S = 1 & pr\{N_S = 1\} = 1 - P \\ N_S = K + 1 & pr\{N_S = K + 1\} = P(1 - P) \\ N_S = 2K + 1 & pr\{N_S = 2K + 1\} = P^2(1 - P) \\ \vdots & \vdots \\ N_S = iK + 1 & pr\{N_S = iK + 1\} = P^i(1 - P) \end{array}$$

Figure 16: Calcolo probabilità di N_S

Nella prima riga si illustra il caso in cui non ci siano errori, per cui $numero_errori * K + 1$ vale 1; nel secondo rigo l'errore è uno, nel terzo rigo gli errori sono due($numero_errori = 2$) e così via, fino al calcolo generale di $N_S = iK + 1$

Inoltre si calcola la probabilità pr

Calcolo del valor medio di N_S Il valor medio si otterrà sommando i valori assunti dalla variabile "i" per la probabilità di assumerli:

$$\begin{aligned} \overline{N_S} &= E[N_S] = \sum_{i=0}^{+\infty} (iK + 1)P^i(1 - P) = (1 - P) \left[\sum_{i=0}^{+\infty} iKP^i + \sum_{i=0}^{+\infty} P^i \right] = \\ &= (1 - P) \left[KP \sum_{i=1}^{+\infty} iP^{i-1} + \frac{1}{1 - P} \right] = (1 - P) \left[\frac{KP}{(1 - P)^2} + \frac{1}{1 - P} \right] = \frac{1 + P(K - 1)}{1 - P} \end{aligned}$$

Quanto vale K

Dal grafico sull'efficienza si può individuare il valore di K approssimato:

$$KT_f \simeq 2T_f + 2\tau \implies K = 2 + 2\alpha \quad (16)$$

sostituendo K nella formula dell'efficienza ottengo:

$$\eta = \frac{1 - P}{1 + P(K - 1)} = \frac{1 - P}{1 + P(1 + 2\alpha)} \quad (17)$$

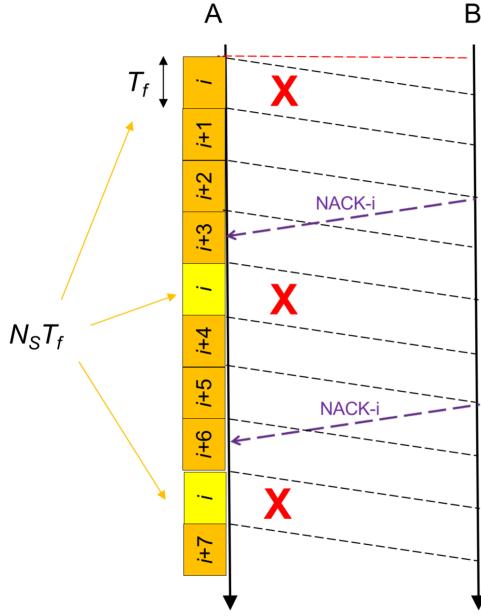


Figure 17: Efficienza del protocollo Selective Repeat

Per calcolare T_{tot} ho bisogno di individuare il valore medio di N_S :

$$\begin{aligned}
 N_S = 1 & \quad pr\{N_S = 1\} = 1 - P \\
 N_S = 2 & \quad pr\{N_S = 2\} = P(1 - P) \\
 N_S = 3 & \quad pr\{N_S = 3\} = P^2(1 - P) \\
 & \vdots \\
 N_S = i & \quad pr\{N_S = i\} = P^{i-1}(1 - P)
 \end{aligned}$$

Figure 18: Calcolo del valor medio di N_S per Selective Repeat

Efficienza Selective Repeat Nel protocollo Selective Repeat, grazie alla ritrasmissione selettiva dei soli frame errati, l'efficienza risulta superiore rispetto al Go-Back-N, soprattutto in presenza di errori frequenti.

Per valutarne l'efficienza si suppone che:

- i riscontri arrivano all'interno della finestra di trasmissione
- l'eventuale trama sbagliata sia sempre la stessa
- si trascurano i tempi di elaborazione e di trasmissione degli ack(τ, T_P)

l'efficienza la calcoliamo allo stesso modo:

$$\eta = \frac{T_f}{T_{tot}} \quad (18)$$

$$\eta = \frac{T_f}{\overline{N_S} \cdot T_f} = \frac{1}{\overline{N_S}} \quad (19)$$

va capito quanto vale T_{tot}

Nel protocollo Selective Repeat, il valore medio di N_S (numero di trasmissioni per frame corretto) si calcola considerando che ad ogni errore viene ritrasmesso solo il frame errato. La variabile aleatoria N_S segue una distribuzione geometrica come nel caso Stop-and-Wait, quindi:

$$\overline{N_S} = E[N_S] = \sum_{i=1}^{+\infty} i P^{i-1} (1 - P) = \frac{1}{1 - P}$$

dove P è la probabilità di errore sul frame. L'efficienza risulta quindi:

$$\eta = 1 - P$$

Si vede chiaramente che il SR ha l'efficienza più elevata tra tutti i protocolli di linea, ma è quello con la maggiore complessità (e quindi costi più alti) per la presenza dei due buffer (in trasmissione e ricezione) e per la necessità di gestire il riordinamento.

Go-Back-N

$$\eta_{GBN} = \frac{1 - P}{1 + P(1 + 2\alpha)}$$

Quando nel caso ideale non si considerano gli errori ($P=0$), GBN e SR hanno stessa efficienza pari a 1 (la massima teorica).

Si noti però che per valori piccoli del ritardo di propagazione normalizzato a tutti i protocolli hanno efficienza comparabile e pari a $1-P$ per cui, in tal caso, sono equivalenti.

Selective Repeat

$$\eta_{SR} = 1 - P$$

0.1.7 Efficienza con riscontri fuori dalla finestra di trasmissione W_s

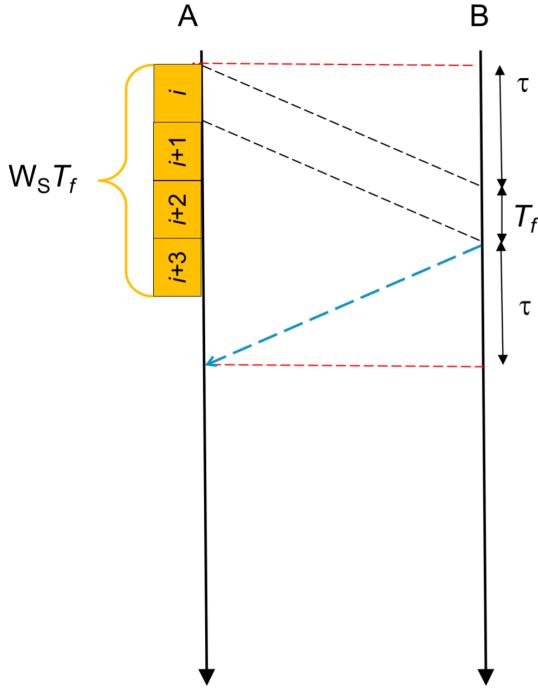


Figure 19: Riscontri fuori dalla finestra di trasm. W_s

Pertanto l'efficienza degli schemi visti prima, va moltiplicata per un fattore(w) di efficienza che pesa l'efficienza finale e che rappresenta il valore limite per SR e GBN nel caso di assenza di errori e riscontri fuori dalla finestra di trasmissione. Tale valore è dato dal rapporto tra tempo utile di occupazione del canale e tempo di attesa del primo riscontro:

$$\eta_W = \frac{W_s T_f}{T_f + 2\tau} = \frac{W_s}{1 + 2\alpha} \quad (21)$$

Per cui le efficienze si SR e GBN sono:

$$\eta_{GBN} = \frac{1 - P}{1 + P(W_s - 1)} \cdot \frac{W_s}{1 + 2\alpha} \quad \eta_{SR} = (1 - P) \cdot \frac{W_s}{1 + 2\alpha}$$

Nel GBN invece, il valore K è proprio W_s in quanto in caso di errore si ritrasmette l'intera finestra. Se si ipotizza stesso valore di N per SR e GBN, considerando il valore ottimale delle finestre, si ha:

$$\begin{aligned} \eta_{GBN} &= \frac{1 - P}{1 + P(N - 2)} \cdot \frac{N - 1}{1 + 2\alpha} \\ \eta_{SR} &= (1 - P) \cdot \frac{N/2}{1 + 2\alpha} \end{aligned}$$

0.1.8 Confronto tra protocolli di linea

	Stop&Wait	Go-back-N	Selective Repeat
$W_S \leq 1 + 2a$		$\frac{1 - P}{1 + P(N - 2)} \frac{N - 1}{1 + 2a}$	$(1 - P) \frac{N/2}{1 + 2a}$
$W_S > 1 + 2a$	$\frac{1 - P}{1 + 2a}$	$\frac{1 - P}{1 + P(1 + 2a)}$	$1 - P$

Figure 20: Confronto dell'efficienza tra i protocolli Stop-and-Wait, Go-Back-N e Selective Repeat

Confronto tra protocolli La scelta del protocollo di linea dipende dalla situazione:

- caso in cui i riscontri siano all'interno della finestra di ricezione: SR è sempre più efficiente, ma per bassi valori di α tutti hanno efficienza simile e pari a $1-P$
- caso in cui i riscontri siano fuori della finestra di trasmissione: il diverso valore della finestra di trasmissione può rendere più efficiente GBN rispetto al SR,
- canali poco rumorosi(P basso) e α elevato: si può avere un'efficienza maggiore del GBN: infatti, è vero che con il GBN si ritrasmette l'intera finestra, ma questo evento accade raramente, mentre durante la trasmissione senza errori si trasmettono più frame di continuo (es. link satellitari)
- canali molto rumorosi(P alto): sarà nuovamente più efficiente il SR poiché si ritrasmette spesso per cui conviene ritrasmettere la singola trama piuttosto che l'intera finestra

La scelta del protocollo dipende dal compromesso tra efficienza, complessità(\$) e condizioni del canale.

0.2 Livello 2 Frame DHL

0.2.1 Struttura del frame

Flag 01111110	Address	Control	Information	FCS	Flag 01111110
8 bit	$8 \times n$ bit	8/16 bit	Variable ($8 \times m$ bit)	16/32 bit	8 bit

Figure 21: Struttura del frame HDLC

- flag: sono posti ad inizio e fine frame così da delimitare il frame e far capire a chi lo riceve quando inizia e quando finisce il frame, questi flag sono SEMPRE 8 bit in questa sequenza: 01111110, ossia 7E in esadecimale
 - address: indirizzo del trasmettitore, ricevitore o broadcast(se tutti 1)
 - control: indica il tipo di frame
 - information: dati del livello superiore, payload
 - FCS(frame check sequence):
-

0.2.2 Bit stuffing

A proposito dei flag usati nel frame(01111110) che succede se questa sequenza invece fa parte della informazione utile del frame e non come flag delimitatore?

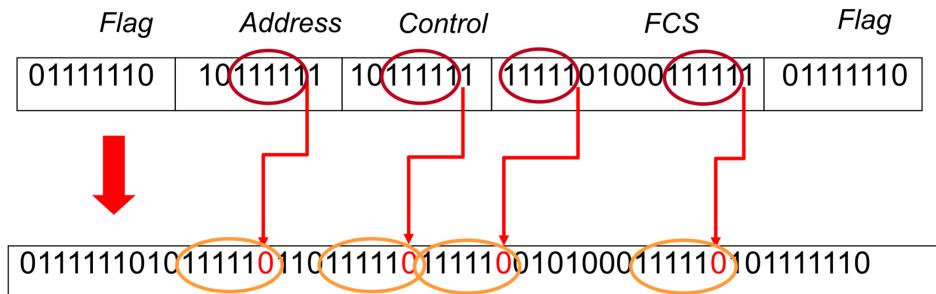


Figure 22: Esempio di bit stuffing

Risolvo inserendo uno 0 ogni cinque 1 ripetuti; questo fa perdere la molteplicità di 8 del frame.

0.2.3 Controllo degli errori casuali con CRC

HDLC numera i frame e usa il protocollo di linea, utilizzando 3 o 7 bit, usando una tecnica di piggybacking per gestire il flusso che viene dalla direzione opposta(finestre di trasmissione e ricezione).

Il CRC è un metodo per il calcolo di somme di controllo (checksum). Il nome deriva dal fatto che i dati d'uscita sono ottenuti elaborando i dati di ingresso i quali vengono fatti scorrere ciclicamente in una rete logica.

Sfrutta l'algebra dei campi finiti ed è utile per l'individuazione di errori casuali nella trasmissione dati (a causa di interferenze, rumore di linea, distorsione), il CRC non è invece affidabile per verificare la completa correttezza dei dati contro tentativi intenzionali di manomissione.

Polinomio generatore Un codice CRC è definito dal suo polinomio generatore di ordine r:

esempio $r = 3$

$$G(x) = x^3 + x^2 + 1 \Rightarrow 1101 \quad (22)$$

Cosa avviene in trasmissione

Con una scelta opportuna del polinomio generatore è possibile rilevare errori sul singolo bit etc... ;

setup del CRC per la trasmissione:

Traduco i bit che compongono header e payload in un polinomio $P(x)$ di coefficienti d; perciò di grado $d-1$.

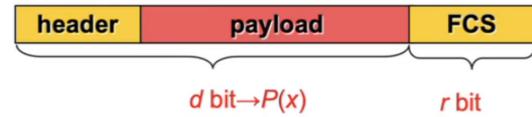


Figure 23: Schema del calcolo di CRC, $P(x)$

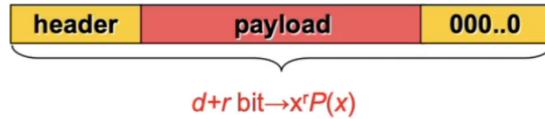
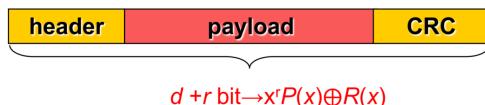


Figure 24: Schema del calcolo di CRC, $x^r P(x)$

Calcolo e verifica del CRC in trasmissione In trasmissione si divide il polinomio $x^r P(x)$ per $G(x)$, il resto di questa operazione[il polinomio $R(x)$] sono i bit del CRC, da inserire nel FCS del frame. Vedi operatore XOR appendice basi.

$$\frac{x^r P(x)}{G(x)} = Q(x) \oplus \frac{R(x)}{G(x)} \quad (23)$$



Devo quindi inserire i bit del polonomio $R(x)$ all'interno del FCS, effettuo l'operazione:

$$P^{Tx} = x^r P(x) \oplus R(x) \quad (24)$$

Figure 25: Inserimento bit in FCS

Esempio calcolo CRC

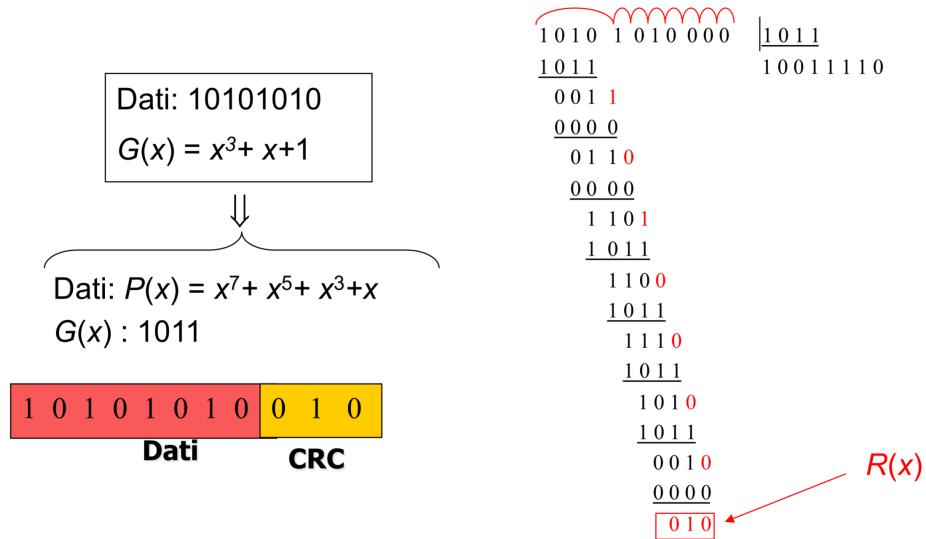


Figure 26: Esempio di calcolo CRC

Verifica CRC

C'è la necessità di verificare se il CRC calcolato ed inserito nel FCS sia corretto.

Polinomio relativo è necessario calcolare il polinomio relativo $P^{Rx}(x)$ per verificare CRC; questo polinomio non è altro che la trama di d+r bit ricevuti in ricezione.

Condizione necessaria Infine per verificare CRC c'è da tenere a mente la condizione necessaria affinché il CSC sia corretto, ossia che il resto della divisione tra polinomio relativo [$P^{Rx}(x)$] e polinomio generatore [$G(x)$] sia nullo:

$$\frac{P^{Rx}(x)}{G(x)} = Q'(x) \oplus R'(x) \quad (25)$$

Può accadere che il resto sia nullo con trama ricevuta errata (la condizione è necessaria, non sufficiente).

Non si può quindi essere certi della correttezza della trama (c'è sempre una probabilità non nulla di falso positivo).

Se il resto NON è nullo, invece, c'è la certezza che la trama sia errata perché la condizione necessaria è stata violata

Dimostrazione condizione necessaria Nel caso in cui la trama ricevuta sia corretta, il polinomio che costruiamo da questa trama deve essere uguale al polinomio originale (trama trasmessa), cioè:

$$P^{Rx}(x) = P^{Tx}(x) \quad (26)$$

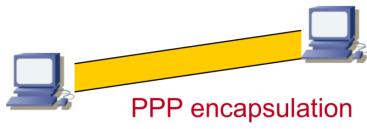
Se effettuiamo la divisione per il polinomio generatore:

$$\frac{P^{Rx}(x)}{G(x)} = \frac{P^{Tx}(x)}{G(x)} = \frac{x^r P(x) \oplus R(x)}{G(x)} = \frac{x^r P(x)}{G(x)} \oplus \frac{R(x)}{G(x)} \quad (27)$$

$$\frac{P^{Rx}(x)}{G(x)} = \frac{x^r P(x)}{G(x)} \oplus \frac{R(x)}{G(x)} = Q(x) \oplus \frac{R(x)}{G(x)} \oplus \frac{R(x)}{G(x)} = Q(x) \quad (28)$$

0.2.4 Protocollo point to point (PPP)

IETF RFC 1661, 1662
 protocollo di livello 2
 punto-punto
 diversi protocolli di livello superiore



Flag	Address	Control	Protocol	Information	FCS	Flag
8 bit	$8 \times n$ bit	8/16 bit	16 bit	Variable ($8 \times m$ bit)	16/32 bit	8 bit

Figure 27: Protocollo Point to Point

Il protocollo Point to Point è comunemente usato nello stabilire connessioni dirette tra due nodi.

Il frame è quasi identico a quello del HDLC, ha un campo aggiuntivo, quello protocol: specifica il protocollo di livello 3 che stiamo trasportando.

Network bit order - Big Endian e LittleEndian A differenza dello standard HDLC, big-endian, il protocollo PPP non è endianness specifico.

L'ordine dei byte (conosciuto anche come big-endian, little-endian o middle-endian a seconda dei metodi differenti), in informatica, indica modalità differenti usate dai calcolatori per immagazzinare all'interno della memoria dati di dimensione superiore al byte; dipende essenzialmente dall'architettura hardware usata.

"I termini big-endian e little-endian derivano dai nomi di due popolazioni che abitavano nelle favolose isole di Lilliput e Blefuscu nel romanzo I viaggi di Gulliver di Jonathan Swift. Queste erano entrate in rivalità per il modo in cui aprivano le uova - rompendo la punta o il fondo: a Lilliput, per editto dell'imperatore il cui figlio una volta si tagliò aprendo un uovo dall'estremità più grande, fu ordinato di aprire le uova dall'estremità più piccola (little endians); a Blefuscu si rifugiarono gli oppositori che volevano conservare la tradizione di rompere le uova dall'estremità più grande (big endians). A causa di questa differenza e della sua legittimazione imperiale era scoppiata tra le due isole una guerra sanguinosa."

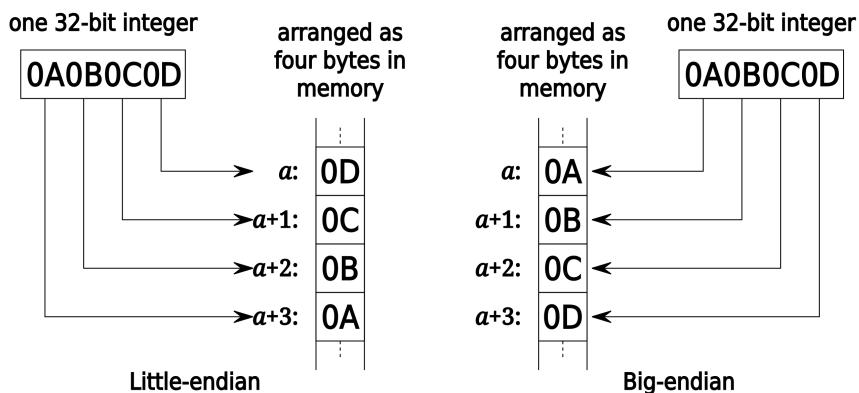


Figure 28: Big Endian e Little Endian

Byte stuffing

Anche qui, vista la struttura del pacchetto molto simile a quella del frame HDLC, c'è il rischio di confondere il dato informativo con un flag; perciò si attua il byte stuffing.

Sequenza di flag 0x7d Se si vuole evitare la sequenza 0x7e (01111110), si antepone la sequenza di controllo 0x7d (01111101); al posto della sequenza 0x7e inseriamo la sua versione in XOR con la sequenza 0x20 (00100000).

A quel punto avremo individuato la sequenza "pericolosa" all'interno del frame, ossia 0x7e, avremo un identificativo, 0x7d, che ci dice dov'è situata nel frame e la sostituiremo con il suo valore in XOR con 0x20.

Quindi quando all'interno del frame leggo il byte 0x7d, il nuovo flag che indica che c'è stato byte stuffing, non leggiamo 0x7d ma il byte successivo, quello calcolato tramite il 0x7e XOR 0x20.

In ricezione se applico nuovamente XOR 0x20 al byte precedentemente modificato con lo XOR, allora otterrò nuovamente il byte iniziale, 0x7e.

Es.

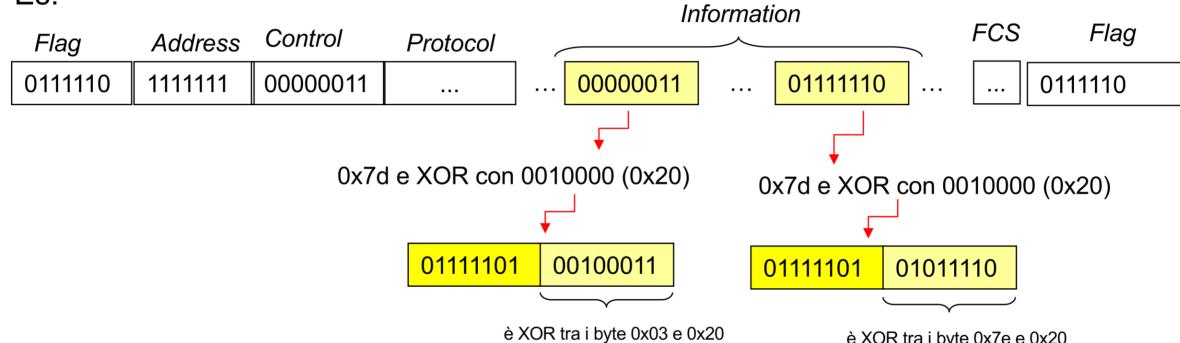
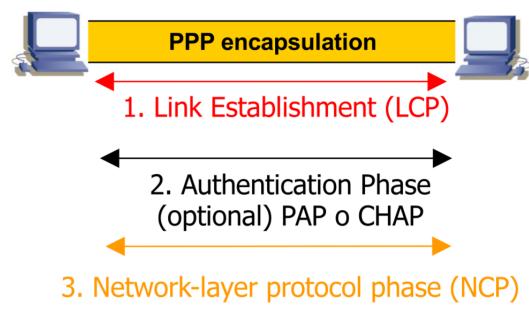


Figure 29: Esempio di byte stuffing

Nel caso in cui incappiamo con una "falsa" sequenza di flag del byte stuffing, ossia 0x7d, la trasmetteremo come 0x7d, 0x5d.

Struttura PPP

Il PPP utilizza due protocolli per la gestione del collegamento tra nodi:



- LCP(Link Control Protocol): grazie al quale si apre la connessione, si ha la verifica della qualità del collegamento e rende possibile l'autenticazione(PAP o CHAP)
- NCP(Network Control Protocol): con cui è possibile scegliere e configurare uno o più protocolli di rete(es. IP)

Figure 30: Encapsulation PPP

Autenticazioni

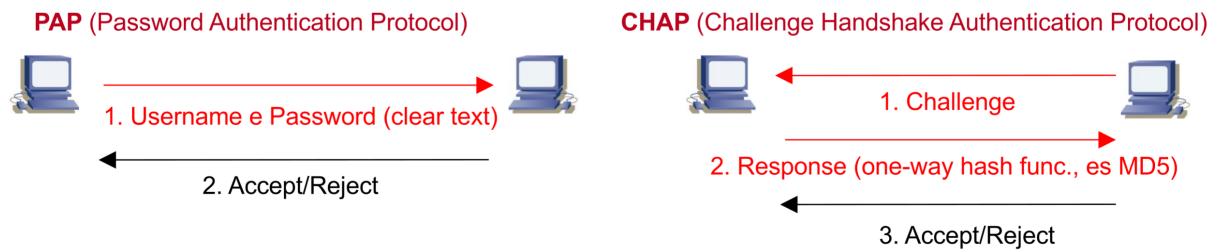


Figure 31: Autenticazione PAP e CHAP

Autenticazione PAP Password Authentication Protocol (PAP) invia username e password in chiaro. È semplice ma poco sicuro, poiché le credenziali possono essere intercettate(password e username sono solitamente criptate).

Autenticazione CHAP Challenge Handshake Authentication Protocol (CHAP) utilizza un meccanismo di challenge-response con hash per autenticare in modo più sicuro, evitando di trasmettere la password in chiaro.

0.3 Protocolli di accesso

A livello due viene svolto un ruolo importante dai protocolli di accesso, che possono essere di due tipologie:

- Accesso ordinato:
 - Ogni nodo ha un turno prestabilito per trasmettere, evitando collisioni.
 - Utilizzato in sistemi come il polling o il token passing.
 - Garantisce un utilizzo equo del canale, ma può introdurre ritardi se un nodo non ha dati da trasmettere.
- Accesso casuale:
 - senza rilevazione del canale:
 - * I nodi trasmettono senza verificare se il canale è libero.
 - * Può portare a collisioni frequenti, come nel protocollo ALOHA puro.
 - con rilevazione del canale:
 - * senza rivelazione di collisioni:
 - I nodi verificano se il canale è libero prima di trasmettere.
 - Non rilevano collisioni, quindi i dati persi devono essere ritrasmessi dopo un timeout.
 - Esempio: Carrier Sense Multiple Access (CSMA).
 - * con rilevazione di collisioni:
 - I nodi verificano se il canale è libero e rilevano eventuali collisioni durante la trasm.
 - In caso di collisione, interrompono la trasm. e ritentano dopo un intervallo casuale.
 - Esempio: CSMA/CD (utilizzato in Ethernet).

0.3.1 Troughput di rete

Per valutare le prestazioni di una rete vengono considerati parametri come ... ; il Troughput, tra questi, è uno dei più rilevanti: è definito come il numero di bit al secondo che giungono corretti; maggiore è questo valore, meglio è per le prestazioni della rete.

Questo valore è alto quando gli errori nei pacchetti che giungono a destinazione sono pochi.

Nota bene: a livello 4 si preferisce in genere utilizzare il goodput definito come bit al secondo che giungono corretti, ma riferiti alla sola informazione utile del livello applicativo. Il throughput invece tiene conto anche di tutti gli header aggiunti ai vari livelli della pila protocollare.

Calcolo del Troughput

Per il calcolo del Troughput è fondamentale definire alcuni concetti:

- frequenza di cifra C : è un parametro tipico del livello 2 (datalink), rappresenta la capacità del canale su cui sta viaggiando l'informazione, quindi la quantità di bit che possono fisicamente muoversi all'interno del canale
- Troughput massimo: è pari alla frequenza di cifra C
- Λ_s : è il numero medio di pacchetti emessi dalla sorgente
- π : la probabilità che il pacchetto arrivi errato o che non arrivi proprio
- Λ_T : Troughput medio

Troughput medio packet/s Il Troughput medio Λ_T , misurato in pacchetti al secondo, si calcola come:

$$\Lambda_T = \Lambda_s(1 - \Pi) \quad [\text{packet/s}] \quad (29)$$

Troughput medio bit/s se desiderassi calcolare il Troughput medio misurato in bit al secondo, ho bisogno di definire il parametro L , ossia la lunghezza media dei pacchetti, allora il Troughput vale:

$$\Gamma_T = \Lambda_s L (1 - \Pi) \quad [\text{bit/s}] \quad (30)$$

Troughput adimensionale è anche possibile definire un valore adimensionale del Troughput, che varia da un valore utile che va da 0 ad uno massimo di 1; lo si ottiene normalizzando Γ_s alla capacità del canale C ; inoltre definiamo:

- $T = \frac{L}{C}$ è il tempo medio di trasmissione di un singolo pacchetto
- $G = \Gamma_s T$ è il traffico medio normalizzato emesso dalla sorgente

$$S = \frac{\Lambda_s L}{C} (1 - \Pi) = \frac{\Lambda_T}{C} = G(1 - \Pi) \quad (31)$$

se S è maggiore di 1 significa che il mittente sta ricevendo più bit di quelli che il canale può supportare.

0.3.2 Protocollo ALOHA puro

ALOHA è un protocollo di accesso multiplo casuale senza rilevazione del canale.

Questo protocollo non prevede vincoli all'invio di dati e quindi all'occupazione della banda. Quando una postazione ha dati da trasmettere, li trasmette.

Poiché ogni stazione agisce indipendentemente dalle altre, il successo è determinato unicamente dalla mancata collisione con altre trasmissioni da parte di altre stazioni. Poiché i canali broadcast danno la possibilità di verificare (feedback) se il frame trasmesso è stato ricevuto correttamente oppure se si sono verificate collisioni, la stazione trasmittente ascolta il canale e determina il successo o l'insuccesso della trasmissione. Qualora non sia possibile ascoltare il canale le stazioni si mettono in attesa di un riscontro (ack) da parte del ricevente. Se ci sono collisioni (o se l'ack non arriva entro un tempo di attesa stabilito), i frame corrotti vengono distrutti. Ciò indipendentemente dal livello di corruzione dei dati; quando un frame è stato interessato da collisione, viene eliminato. In questo caso la postazione mittente reinvia il frame dopo un'attesa casuale e si rimette in ascolto sul canale (o attende un ack) fino a quando non stabilisce che il frame è stato ricevuto correttamente.

Collisioni nel canale condiviso

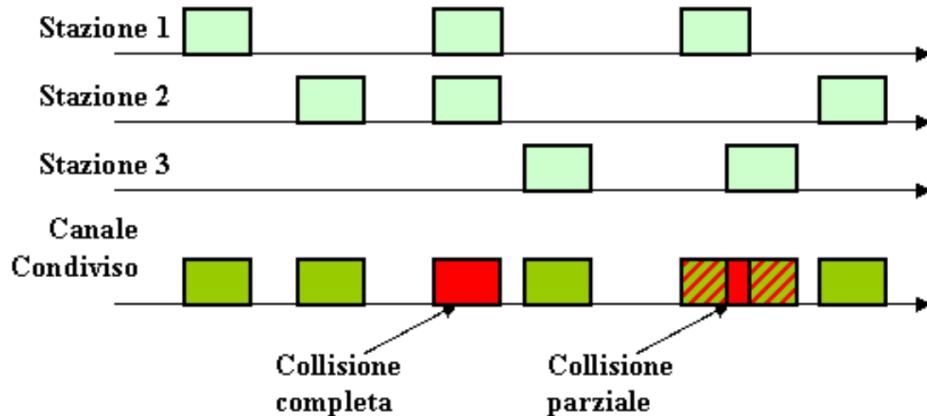


Figure 32: Collisioni nel protocollo ALOHA

Le stazioni trasmittenti inviano i loro messaggi lungo il canale condiviso. Se l'intervallo di tempo durante il quale due o più stazioni trasmettono si sovrappone, allora si avrà una collisione; quindi le stazioni coinvolte non riceveranno alcun ack e dovranno perciò ritentare la trasmissione in un istante successivo.

Per evitare che la collisione si ripeta indefinitamente è opportuno che le stazioni coinvolte tentino la loro ritrasmissione in tempi distinti, in modo da ridurre la probabilità di nuove sovrapposizioni fra i due periodi di trasmissione.

Calcolo del Troughput di ALOHA puro

Si ipotizza che le stazioni trasmettano sul canale secondo un processo di Poisson, quindi ognuna di esse è un evento indipendente dagli altri; le stazioni trasmettono pacchetti nel canale indipendentemente da ciò che fanno le altre stazioni.

Si suppone anche che ci sia un numero di stazioni molto elevato, che trasmettano tutte nel canale condiviso di interesse;

ricordando la formula del Troughput medio espresso in pacchetti al secondo:

$$\Lambda_T = \Lambda_s(1 - \pi) \quad [\text{packet/s}] \quad (32)$$

ci interessa capire quanto può valere $(1 - \pi)$ nel protocollo ALOHA puro, quindi la probabilità che i pacchetti arrivino a destinazione considerando solamente l'effetto della collisione tra pacchetti, non altri fenomeni, poiché si vuole valutare solo l'effetto del protocollo ALOHA sul Troughput.

Per fare ciò considero un nuovo parametro, il periodo di vulnerabilità.

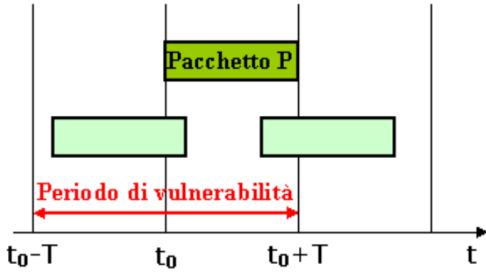


Figure 33: Periodo di vulnerabilità

Il periodo di vulnerabilità rappresenta l'intervallo di tempo durante il quale una trasmissione(pacchetto verde chiaro) può essere soggetta a collisioni(pacchetto P collide con la trasmissione se rientra nella finestra di $2T$). Nel caso del protocollo ALOHA puro, questo periodo è pari a $2T$, dove T è il tempo di trasmissione di un singolo pacchetto. Durante questo intervallo, se un'altra stazione inizia a trasmettere, si verifica una collisione.

Il valore $(1 - \pi)$ rappresenta quindi la probabilità per la quale nell'intervallo di trasmissione di un pacchetto questo arrivi correttamente(infatti $1 - \pi$, dove 1 è la probabilità massima per la quale non ci sia collisione, π invece rappresenta la probabilità di collisione, più è alta e più andrà a diminuire il valore ideale 1).

Dall'espressione di Poisson(p_0 rappresenta la probabilità che zero eventi (collisione di pacchetti) si verifichino in un certo intervallo di tempo) si calcola che la probabilità di non avere eventi di collisione nell'intervallo di vulnerabilità $2T$ è la seguente:

$$p_k = \frac{(\Lambda t)^k}{k!} e^{-\Lambda t} \Rightarrow p_0 = \frac{(\Lambda_s 2T)^0}{0!} e^{-\Lambda_s 2T} = e^{-2\Lambda_s T} = (1 - \pi) \quad (33)$$

Troughput in packet/s sostituendolo nella formula di Λ_T :

$$\Lambda_T = \Lambda_s(1 - \Pi) = \Lambda_s p_0 = \Lambda_s e^{-2\Lambda_s T} \quad \text{packet/s} \quad (34)$$

Troughput in bit/s

$$\Gamma_T = \Lambda_s L(1 - \pi) = \Lambda_s L(e^{-2\Lambda_s T}) \quad [\text{bit/s}] \quad (35)$$

Troughput adimensionale

$$S = \frac{\Lambda_s L}{C}(1 - \Pi) = \frac{\Lambda_T}{C} = G(1 - \Pi) = G(e^{-2\Lambda_s T}) \quad (36)$$

Prestazioni ALOHA puro

Andamento funzione $G(e^{-2\Lambda_s T})$ analizzando la funzione

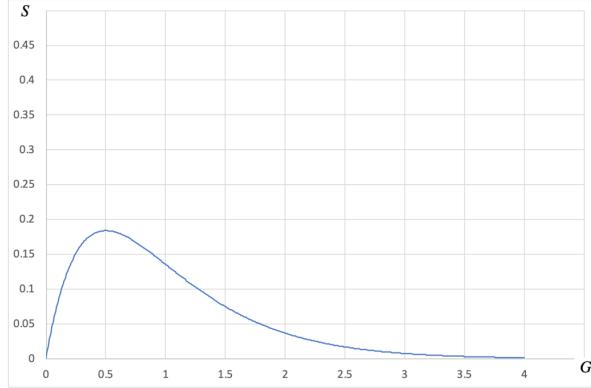


Figure 34: Andamento Troughput normalizzato ALOHA puro

La funzione $S = Ge^{-2G}$ ha un massimo per $G = 0.5$, che corrisponde a $S = 0.184$.

Questo significa che il massimo Troughput normalizzato ottenibile con ALOHA puro è circa il 18.4% della capacità del canale, un valore molto basso dovuto all'elevata probabilità di collisione.

Le prestazioni basse sono dovute al fatto che le stazioni trasmettono senza alcun controllo sul canale.

Appendice matematica - distribuzione di Poisson

La distribuzione di Poisson $\mathcal{P}_\lambda(n)$ è una distribuzione di probabilità discreta data da

$$\mathcal{P}_\lambda(n) = \frac{\lambda^n e^{-\lambda}}{n!} \quad \text{per ogni } n \in N, \quad (37)$$

dove λ è il numero medio di eventi per intervallo di tempo, mentre n è il numero di eventi per intervallo di tempo (lo stesso col quale si misura λ) di cui si vuole la probabilità.

Esempio Un call center riceve in media 4 chiamate all'ora. Qual è la probabilità che riceva esattamente 2 chiamate in un'ora? ($\lambda = 4$, $n = 2$)

$$\mathcal{P}_4(2) = \frac{4^2 \cdot e^{-4}}{2!} = \frac{16 \cdot e^{-4}}{2} \approx \frac{16 \cdot 0.0183}{2} \approx \frac{0.2928}{2} \approx 0.1464$$

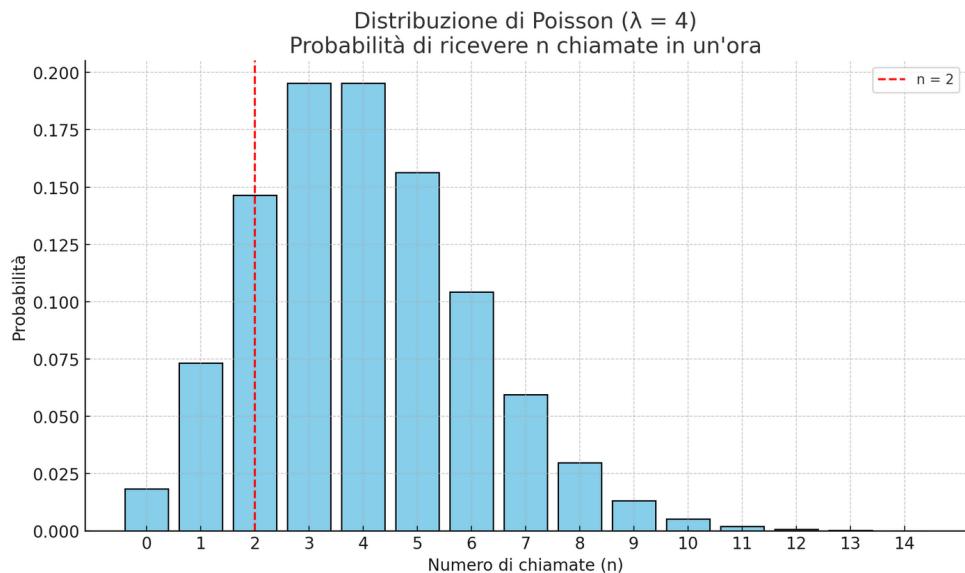


Figure 35: Grafico della distribuzione di Poisson

0.3.3 Protocollo ALOHA slotted - prestazioni

Il protocollo Slotted Aloha aggiunge al protocollo Aloha (da cui deriva) un'ulteriore caratteristica, ovvero la suddivisione del tempo in intervalli discreti chiamati slot.

Ogni stazione è vincolata a cominciare la propria trasmissione all'inizio di uno slot temporale.

Se una stazione ad un certo istante è pronta a trasmettere dovrà attendere necessariamente l'inizio del successivo slot. La conseguenza di tale caratteristica è che due trasmissioni o collidono completamente all'interno dello stesso slot oppure non collidono affatto; il problema delle collisioni parziali non c'è.

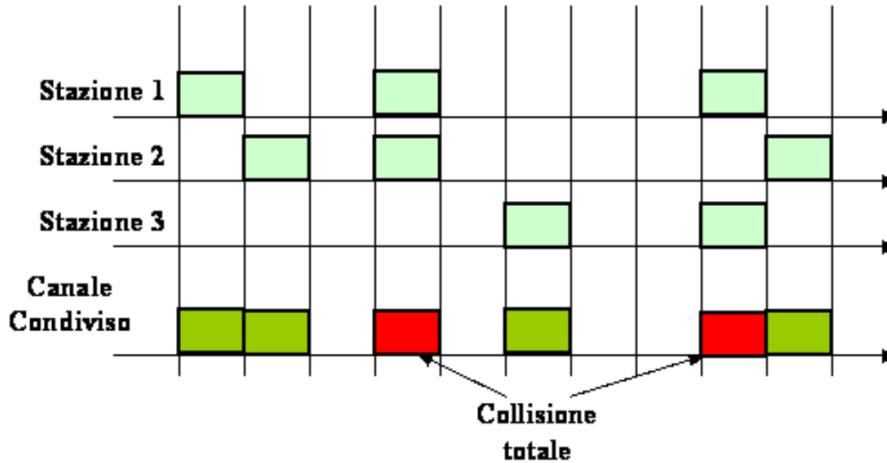


Figure 36: Protocollo ALOHA slotted

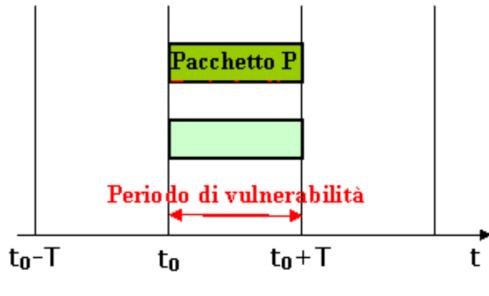


Figure 37: Protocollo ALOHA slotted

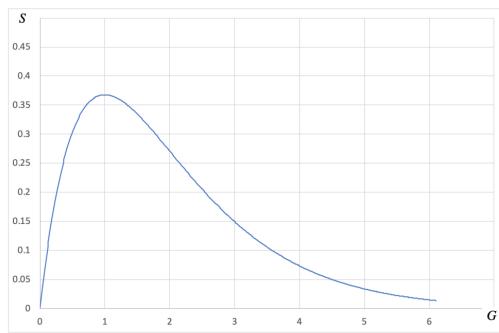


Figure 38: Protocollo ALOHA slotted

Probabilità di collisione Il modello è identico a quello visto per ALOHA, ma stavolta il tempo di vulnerabilità si è dimezzato per cui la probabilità di non avere tentativi di trasmissione in T

$$p_0 = \frac{(\Lambda_s T)^0}{0!} e^{-\Lambda_s T} = e^{-\Lambda_s T} \quad (38)$$

Troughput normalizzato

$$S = \Lambda_s \frac{L}{C} p_0 = \Lambda_s \frac{L}{C} e^{-\Lambda_s T} = G e^{-\Lambda_s T} \quad (39)$$

Il protocollo Slotted Aloha ha come conseguenza il dimezzamento del periodo di vulnerabilità, che in tal caso è pari a T. L'efficienza massima risulta conseguentemente raddoppiata, pari quindi al 36,8%.

0.3.4 Protocolli Carrier Sense Multiple Access - CSMA (rilevazione canale)

I protocolli CSMA sono quei protocolli di accesso che rilevano il canale prima di trasmettere, per migliorare l'efficienza ed evitare collisioni con altre trasmissioni.

Se rilevando il canale, risulta occupato, allora la stazione può attuare diverse strategie:

- 1-persistent: continuare a "sentire" il canale finché non lo rileva libero per trasmettere subito dopo
- no-persistent: rinviare la trasmissione ad un momento successivo, generalmente scegliendo un intervallo di attesa casuale
- p-persistent: scegliere di comportarsi con probabilità p in una delle due modalità precedenti

"Può accadere di rilevare il canale come libero, ma nel tempo in cui l'ho rilevato e ho iniziato ad utilizzarlo qualcun'altro potrebbe iniziare a sua volta la comunicazione, c'è quindi da considerare il tempo che impiega l'onda elettromagnetica a viaggiare nel mezzo v e noi a rilevarla T_p ".

CSMA/CD - CSMA with collision detection

Per realizzare il CSMA/CD è necessario che la stazione possa a livello fisico realizzare una comunicazione full duplex (per essere in grado a livello fisico di trasmettere e ricevere contemporaneamente, per permettere il listening while talking spiegato successivamente).

La stazione che intende trasmettere verifica che il canale sia libero prima di trasmettere (sensing del canale del CSMA);

Listening while talking se il canale è libero si trasmette la trama e durante la trasmissione della stessa (tra l'invio del primo e dell'ultimo bit) si continua ad ascoltare il canale (Listening while talking)(ascolta se stessa).

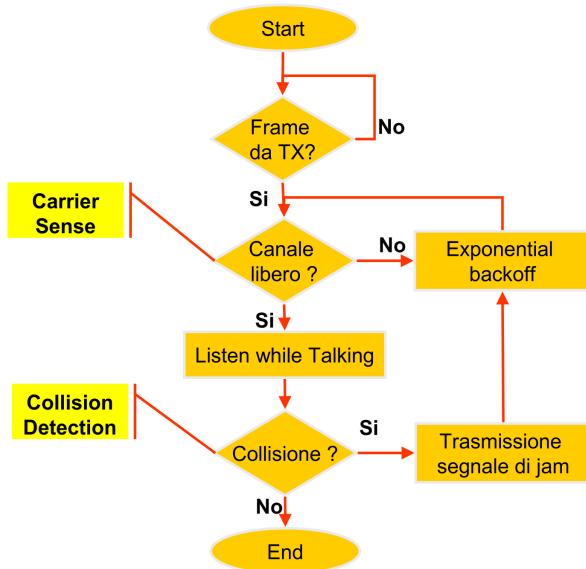


Figure 39: Esempio CSMA/CD

Rilevazione collisione Se il segnale "ascoltato" (confronto tra le forme d'onda a livello fisico) è diverso da quello trasmesso, si rileva la collisione (altra stazione che ha iniziato a trasmettere sentendo il canale libero).

In tal caso si può:

- interrompere la trasmissione
- inviare un segnale di jam per informare chi sta ascoltando il canale di scartare nel buffer di ricezione tutti i bit della trama parziale ricevuta

Exponential backoff Il tempo di attesa per la rilevazione del canale in seguito ad una collisione viene calcolato in un intervallo $T_{slot} * [0; 2^m - 1]$ tramite un algoritmo di exponential backoff, per il quale $m = \min(n; 10)$, m è il numero di ritrasmissioni, n è il numero di collisioni consecutive, T_{slot} è il tempo necessario a trasmettere un frame di lunghezza minima(512 bit).

Dimostrazione funzionamento collision detection

Utilizzando il CSMA/CD è necessario che i frame abbiano una dimensione superiore ad un valore minimo per garantire il funzionamento del collision detection.

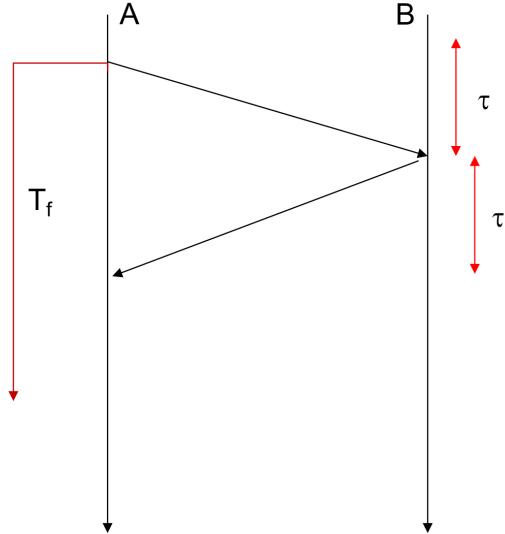


Figure 40: Dimostrazione collision detection

A e B comunicano nei tanti τ (valore massimo, caso peggiore); il primo bit inviato da A arriva a B dopo τ secondi.

Nel caso peggiore B ascolta il canale un istante(ϵ) prima che il bit di A arrivi a B, perciò B vedrà che il canale è libero, non sapendo che un'istante dopo arriveranno i bit di A, causando quindi una collisione. B si accorge della collisione e manda un jam, impiegando τ per "avvisare" A.

A può aprire che il jam che riceve è dovuto ad una collisione con la sua trasmissione solo se la sua trasmissione (T_f) dura almeno quanto il tempo che ha impiegato il suo primo bit di trasmissione ed il jam di B, perciò: $T_f \geq 2\tau$

$$T_f = \frac{L_f}{C} \geq 2\tau = \frac{2d}{v} \implies L_f \geq \frac{2dC}{v} \quad (40)$$

CSMA/CA - CSMA with collision avoidance

Duration L'obiettivo del CSMA/CA è diminuire il numero di collisioni.

In questa versione di CSMA, in cui c'è rilevazione del canale prima di trasmettere, se il canale è libero inizia la trasmissione inserendo la durata stimata della trasmissione del singolo frame (considerando anche i tempi di propagazione e la trasmissione del frame di riscontro) in un campo "duration" dell'header del frame.

Network allocation vector(NAV) - virtual sensing Tutte le altre stazioni che condividono il canale ascoltano il frame e settano un loro orologio interno, detto NAV, pari al valore del duration letto.

Finchè il NAV non si azzera non iniziano altre trasmissioni poiché si sa che il canale è occupato.

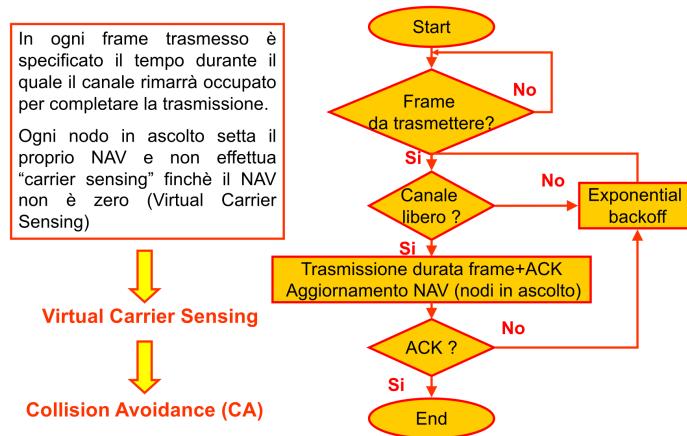


Figure 41: CSMA/CA