



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Data-Link

Parte I : Protocolli

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il Livello Data-Link: sommario

PARTE I

- ▶ Scopi del livello Data-Link
- ▶ Framing, Rilevazione e correzione degli errori, controllo di flusso
- ▶ Protocolli per reti Punto-punto: PPP.
- ▶ Protocolli per reti MultiAccesso: Aloha, CSMA, CSMA/CD, CSMA/CA

PARTE II

- ▶ Gli standard IEEE802
- ▶ Ethernet: Sottoliv. MAC e LLC, tecnologie Ethernet, il Frame, Repeater, Switch, Bridge
- ▶ Spanning Tree Protocol.
- ▶ Lan Virtuali

PARTE III

- ▶ Lan Wireless

RIFERIMENTI

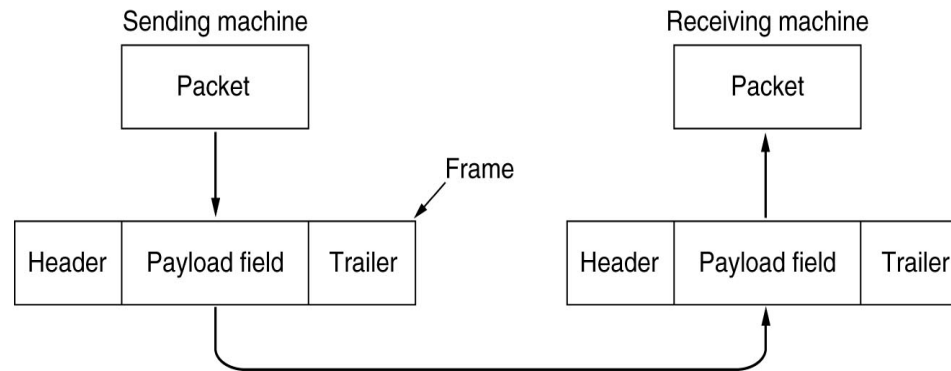
- ▶ *Reti di Calcolatori*, A. Tanenbaum, ed. Pearson
- ▶ *Reti di calcolatori e Internet*, Forouzan , Ed. McGraw-Hill

Scopi del Livello Data-Link

Il livello Data-Link ha la responsabilità di trasferire in modo sufficientemente affidabile i dati tra nodi adiacenti, ovvero su canali punto-punto o multi-accesso.

La comunicazione affidabile è realizzata mediante la **Suddivisione del flusso di dati in “frame”** con lunghezza massima fissata.

Il Frame contiene un payload che viene riempito con i dati da trasportare provenienti dal livello superiore e informazioni di servizio poste in testa e/o in coda al payload (header/trailer).



Le informazioni di servizio vengono utilizzate principalmente per:

- ▶ Delimitare inizio e fine del frame (framing)
- ▶ Gestire la **rilevazione ed eventualmente la correzione degli errori**.
- ▶ Gestire (eventualmente) il **controllo del flusso**
- ▶ Gestire l'**accesso al mezzo trasmissivo** (nei canali Multi-Accesso)

Servizi offerti al livello Network

I servizi forniti al livello network possono essere:

- ▶ **Senza connessione e senza conferma**

- Semplice e veloce, è adatto a mezzi trasmissivi affidabili (Es. Ethernet)

- ▶ **Senza connessione ma con conferma**

- Viene inviato un Frame di conferma per ogni Frame inviato.
- Utile se il mezzo è poco affidabile (es. LAN Wireless)

- ▶ **Con connessione e con conferma**

- Ogni frame inviato è parte di una connessione e quindi dotato di una numerazione
- Garantisce che ogni Frame viene ricevuto una sola volta e riordinato.
- 3 fasi distinte:
 - attivazione della connessione
 - invio dati numerati e conferme
 - chiusura connessione
- Overhead elevato. Raramente utilizzato a livello Link. Implementato nei livelli superiori (TCP)

Impacchettamento (Framing)

Il primo problema da risolvere è come delimitare inizio e termine di un frame.

I frame possono avere dimensione fissa o variabile. Se la dimensione è fissa non è necessario delimitare il frame (vedi la rete telefonica ATM). Se la dimensione è variabile occorre una strategia per distinguere i frame.

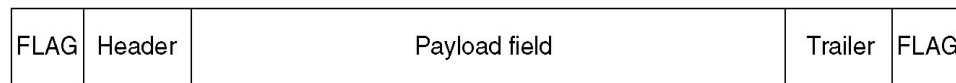
Possibili strategie:

- ▶ Un intervallo **temporale** tra un frame ed il successivo
- ▶ Far precedere ogni Frame con il **numero di byte del frame**.
- ▶ Delimitare il Frame con caratteri speciali (**Flag**)

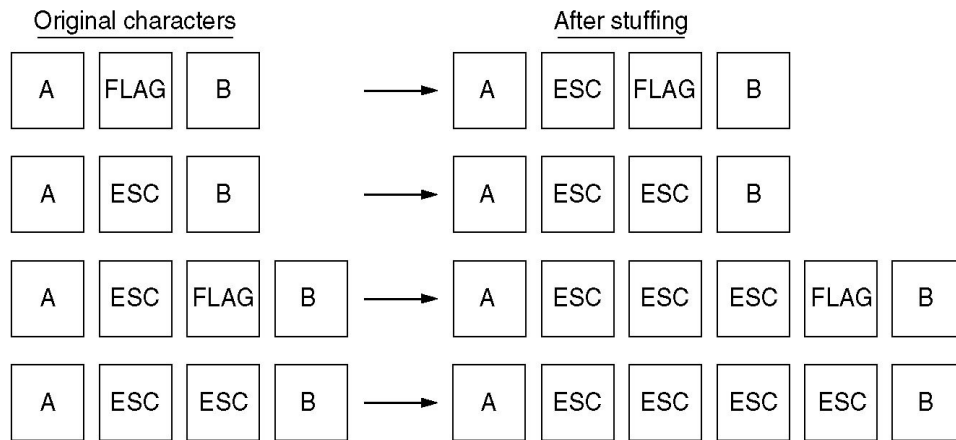
Gran parte dei protocolli di Data-Link usano l'abbinamento del Flag e dell'intervallo temporale per aumentare la ridondanza.

Framing con “ESC Stuffing”

La delimitazione del Frame è marcata da un Byte speciale denominato **FLAG**.
Un problema potrebbe nascere se all'interno del Frame è presente la sequenza di FLAG.
Per flussi **Byte-Oriented**, si può inserire un Byte di Escape (ESC) appena prima dell'occorrenza accidentale del Flag (o dell'ESC).
Il destinatario dovrà realizzare l'operazione di “destuffing”.

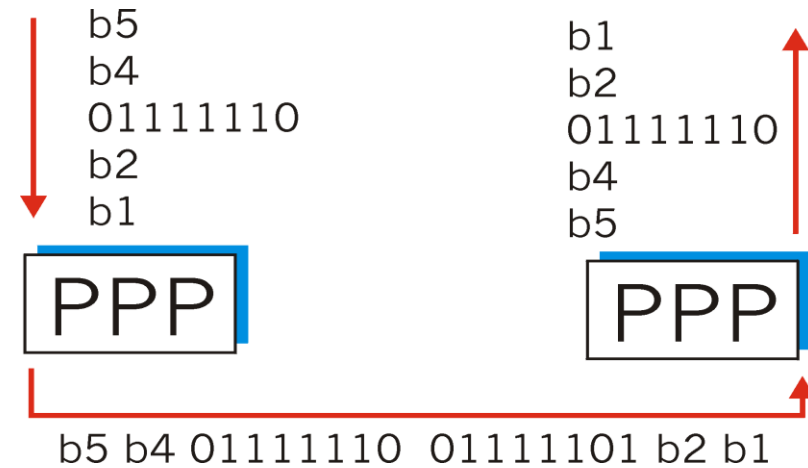


(a)



(b)

Esempio PPP (ESC = 01111101).



Bit stuffing


Se i flussi sono **bit-oriented** si può utilizzare il “Bit-stuffing”:

Ogni Frame inizia e termina con la sequenza 01111110 (Flag)

Ogni volta che nella trama si incontra la sequenza 11111 (5 uni) viene aggiunto un bit 0 (bit stuffing) per non confondere il destinatari.

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0



Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(a) Dati originali

(b) Dati elaborati dal mittente che aggiunge i bit di stuffing

(c) Dati elaborati dal destinatario che elimina i bit di stuffing.

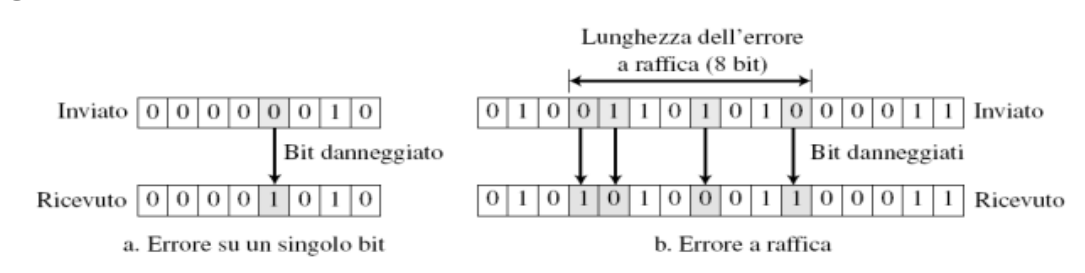
Rilevazione e Correzione degli errori

Durante la trasmissione di un Frame possono verificarsi disturbi o rumore termico che possono cambiare la forma del segnale e quindi alterare la ricezione dei bit.

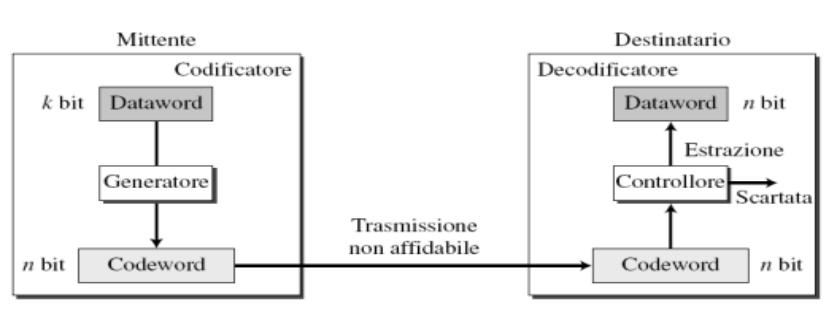
Gli errori sono rari su linee ottiche, mentre possono essere frequenti su canali come wireless o “ultimo miglio” sulla linea ADSL.

Tipi di errori:

- ▶ a bit singolo
- ▶ a raffica (“burst”)



Per individuare gli errori si utilizza la **Ridondanza**: il mittente, attraverso un opportuno algoritmo, determina una breve codice (Frame Control Sequence - FCS), che verrà inviata assieme al Frame. Se il destinatario riapplicando l'algoritmo otterrà una sequenza FCS diversa capirà che si è verificato un errore.



Rilevazione e Correzione degli errori: algoritmi

Esistono 2 strategie possibili:

Rilevazione degli errori (senza correzioni): richiede algoritmi più semplici ed un FCS più breve. Si utilizza su canali affidabili (es. Fibra Ottica), in cui gli errori sono rari e conviene eventualmente **ritrasmettere** il Frame.

Nota: La richiesta di ritrasmissione può essere

- esplicita: il ricevente manda un NACK in caso di errore
- automatica con protocollo ARQ (Automatic Repeat ReQuest): il mittente attiva un timer, il ricevente invia un ACK per i dati ricevuti correttamente e scarta i dati con errore; allo scadere del timer il mittente rispedisce il frame.

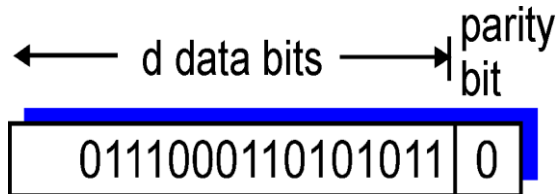
ARQ è in grado di gestire sia frame errati che frame perduti

Rilevazione e correzione degli errori: richiede algoritmi più complessi e maggiore ridondanza nel FCS. Si utilizza raramente, in reti poco affidabili o in trasmissioni Simplex, in cui non è possibile inviare al mittente la richiesta di ritrasmissione.

Codifica a blocchi: bit di parità

Semplice algoritmo per rilevazione dell'errore.

Il numero totale di 1 nella sequenza, compreso il bit di parità, deve essere dispari (o pari)



Applicato alla sequenza di un Frame determina l'esistenza di un singolo errore all'interno della sequenza.

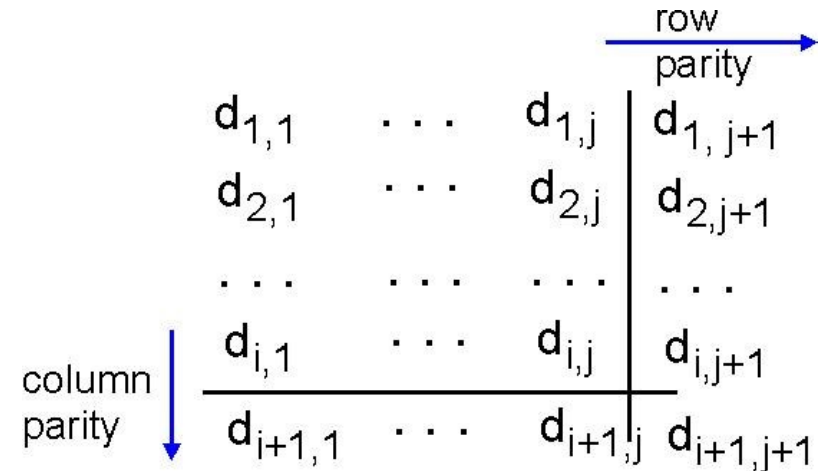
Il bit di parità si usa in molti dispositivi hardware come ad esempio nei bus SCSI e USB e in molte cache di microprocessori

Codifica a blocchi: bit di parità 2D

Semplice algoritmo **esemplificativo** per la correzione dell'errore.

Suddividendo il Frame in più sotto-sequenze di uguale lunghezza possiamo calcolare la parità in 2 dimensioni e quindi individuare e correggere il singolo bit errato.

Questa tecnica è però poco efficiente e **non trova impieghi reali**.



1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

no errors

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

parity error
parity error

*correctable
single bit error*

Cyclic Redundancy Check (CRC)

Un Frame di d bit è visto come una lista di coefficienti di un polinomio D con d termini (di grado $d-1$). Per esempio 110001 rappresenta $x^5 + x^4 + x^0$

Trasmittitore e Ricevitore si mettono d'accordo su di un polinomio comune G di $r+1$ bit (grado r) detto "generatore", che deve essere un numero primo.

Il **Trasmittitore** aggiunge r bit (il CRC) al termine della sequenza del Frame in modo che il nuovo Frame M (di grado $r+d-1$) sia divisibile per G .

Procedura:

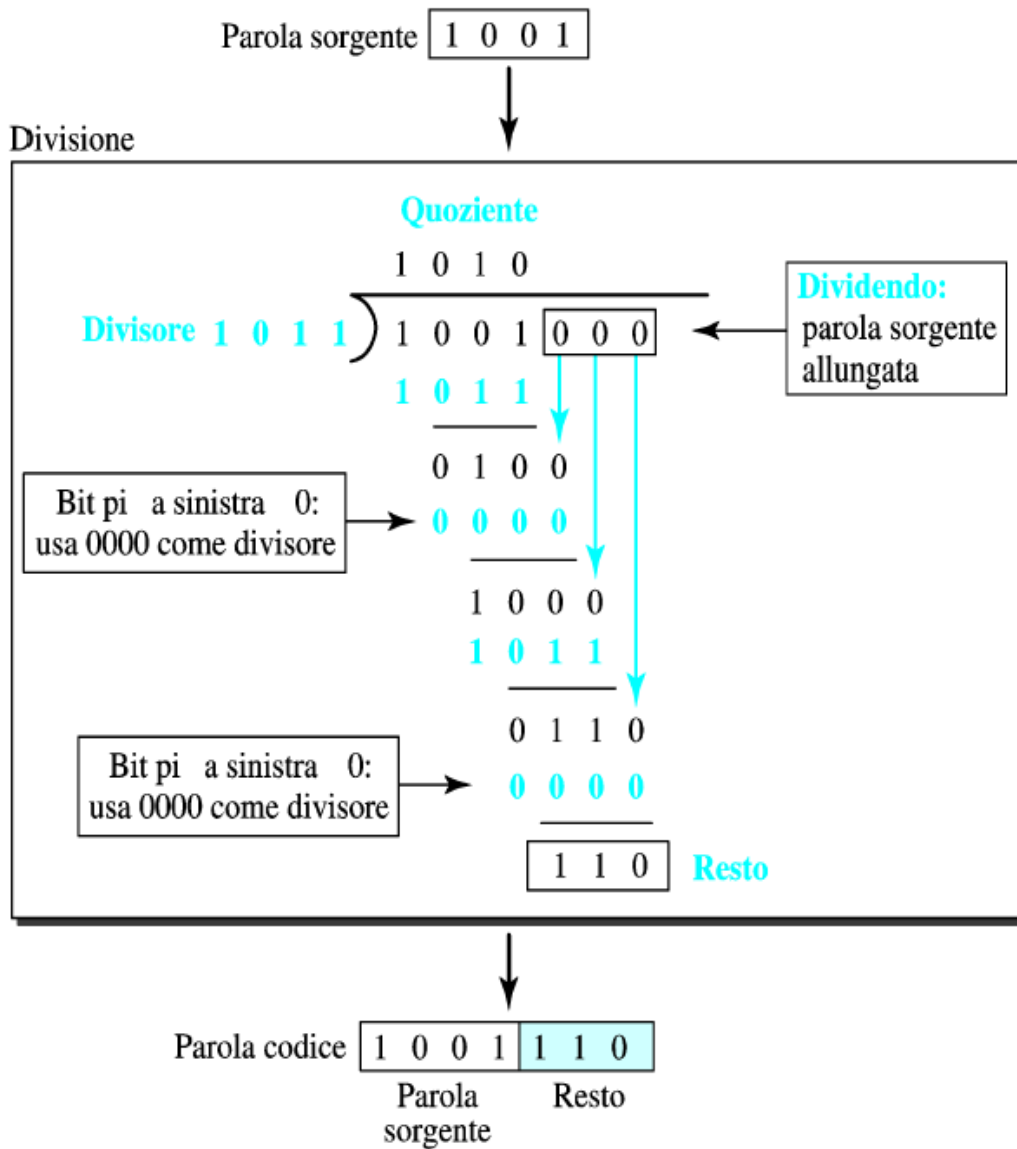
- 1) $N = D \cdot x^r$ (vengono aggiunti r zeri al termine di D)
- 2) $R = N/G$ (viene determinato il resto R della divisione)
- 3) $M = N - R = N \text{ xor } R$ ($N - R$ è divisibile per G . La sottrazione in mod2 si fa con XOR).

Il **Ricevitore** divide M/G . Se il resto è diverso da zero si è verificato un errore.

Procedura:

- 1) $R = M/G$ (viene determinato il resto R della divisione)
- 2) if ($R \neq 0$) then ERROR

CRC: esempio di calcolo



Il polinomio generatore (divisore) è di 4 bit (grado 3)

Si calcola l'XOR tra i primi 4 bit del dividendo e divisore.

Al risultato si aggiungono progressivamente i bit rimanenti del divisore e si ripete l'XOR con il divisore.

Se il bit più a sinistra del dividendo è 0, in quel passo occorre un divisore fatto di tutti 0.

CRC

Uno dei vantaggi del CRC è che i moduli di codifica e decodifica possono essere **facilmente implementati in hardware** usando componenti elettronici poco costosi.

La codifica polinomiale CRC con r bit di controllo è in grado di rilevare sequenze di errori di lunghezza fino a r .

Il CRC è il codice più utilizzato nei protocolli data-link:

Ethernet: Usa il CRC-32 ($x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1$) riesce a rilevare fino a 32 errori e tutti quelli che toccano un numero dispari di bit.

PPP: Usa CRC-16

ATM: Usa CRC-8

Checksum (somme di controllo)

Il checksum è un'altra tecnica, basata sulla rappresentazione dei numeri in complemento a 1, per individuare differenze tra i bit inviati da mittente e quelli ricevuti dal destinatario.

Il mittente divide il segmento in blocchi da 16 bit e li somma, quindi ne fa il complemento e inserisce il risultato nel campo checksum.

Il ricevente ricalcola il checksum (incluso il checksum ricevuto) senza complemento finale.

Il risultato, senza errori di trasmissione è una sequenza di 1, altrimenti c'è un errore.

E' adatto per implementazioni software e per questo non è usato a livello Link ma nei livelli superiori (IP, ICMP, TCP, UDP)

Il termine **Checksum** è spesso utilizzato per intendere in generale le tecniche per verificare l'integrità di un dato o di un messaggio.

Checksum: operazioni

Esempio a 8 bit:

Mittente:	Destinatario:
10101000	10101000
01010010	01010010
10001110	10001110
-----	01110111 (checksum)
10001000 (somma)	-----
01110111 (checksum)	11111111 (checksum OK)

La funzione mittente a 16 bit:

```
chsum(u_short *buf, int count)
{
    register u_long sum=0;
    while (count --)
    {
        sum += buf++;
        if (sum & 0xffff0000)    {sum &= 0xffff; sum++;}
    }
    return ~(sum & 0xffff);
}
```


Protocolli per il controllo del flusso

Protocolli condivisi tra mittente e destinatario per garantire il corretto invio del flusso dei dati. Possono essere implementati a livelli Link o ai livelli superiori.

Protocolli in modalità non connessa

Per canali senza rumore

- ▶ Semplice
- ▶ Stop-and-wait (con conferma)

Protocolli in modalità connessa

Per canali con rumore

Si basano sulla numerazione dei frame

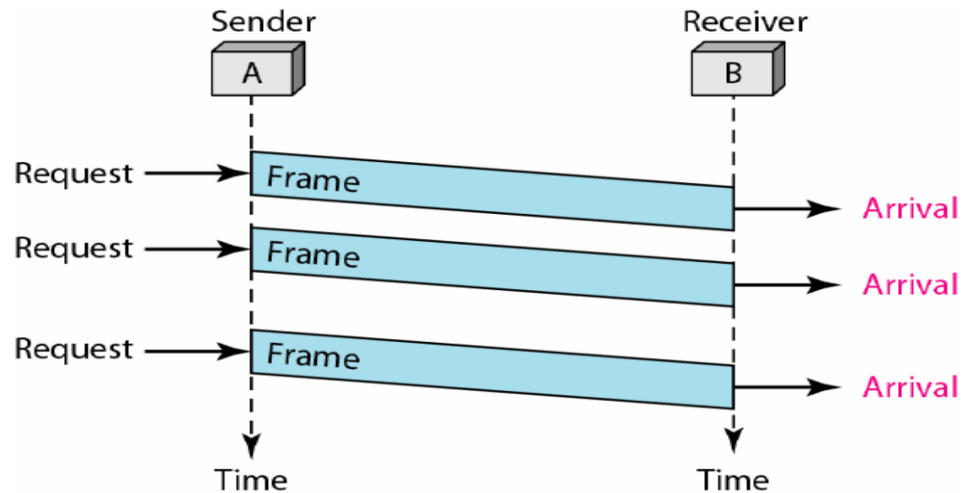
- ▶ Stop-and-wait ARQ
- ▶ Protocolli a finestra scorrevole (Sliding Window)
 - Go-back-N ARQ
 - Ripetizione selettiva ARQ

Protocollo semplice (simplex) senza restrizioni

Scenario (ideale):

- ▶ Il destinatario è sempre pronto a ricevere e a gestire i frame ricevuti.
- ▶ I dati arrivano senza errori

Protocollo Semplice:



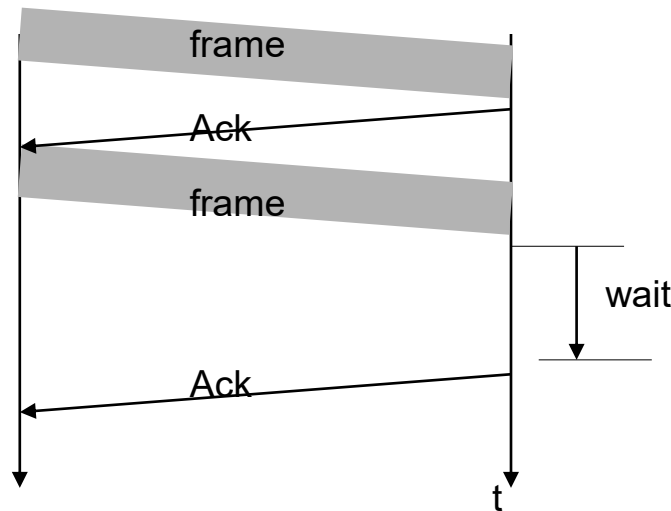
Protocollo stop-and-wait

Scenario:

- ▶ Il ricevente ha bisogno di tempo per elaborare i dati ricevuti

Protocollo Stop-and-Wait:

- ▶ Prima di inviare il prossimo dato il mittente deve ricevere una **conferma (Ack)**, per cui il destinatario, se sovraccarico, può moderare il tasso di invio dei dati ritardando l'invio di Ack (controllo di flusso con conferma).



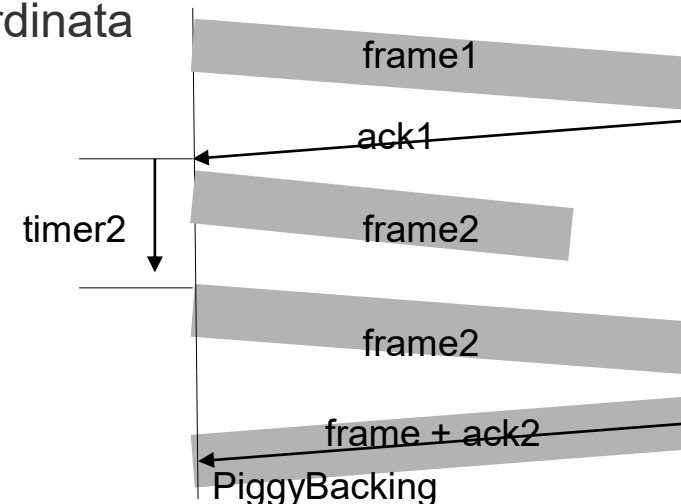
Protocollo stop-and-wait ARQ

Scenario:

- ▶ Il ricevente ha bisogno di tempo per elaborare i dati ricevuti
- ▶ I frame possono essere danneggiati o perduti (canale rumoroso o disturbato)

Protocollo Stop-and-Wait ARQ (Automatic Repeat reQuest):

- ▶ Poiché i frame possono andare perduti viene gestito l'invio di un frame di servizio, denominato ACK, a conferma della corretta ricezione.
- ▶ I frame danneggiati vengono scartati, oppure viene inviato un NACK (Not ACK).
- ▶ Per gestire la perdita di frame, il mittente attiva un timer per ogni frame inviato. Se il mittente non riceve un ACK in un certo tempo il frame viene **rispedito**.
- ▶ **Servizio di Connessione** : consegna garantita e ordinata grazie a **numerazione** dei frame e dei relativi ACK
- ▶ Il mittente deve mantenere copia dei frame fino all'ACK
- ▶ Se il traffico è bidirezionale l'ACK può viaggiare in un frame dati inviato in senso opposto (**PiggyBacking**).
- ▶ Nei protocolli Stop-and-wait la capacità del canale non è sfruttata al meglio poiché occorre attendere l'arrivo dell'ACK.



Protocolli “Sliding Window”

I protocolli “**Sliding Window**” **migliorano l'efficienza del canale** consentendo al trasmettitore di poter inviare fino SWS (Sender Window Size) frame senza attendere il riscontro ACK.

La finestra si sposta in avanti man mano che i riscontri arrivano. I Frame appartenenti alla finestra vengono memorizzati dal mittente per eventuali ritrasmissioni.

Il mittente assegna ad ogni Frame un **numero di Sequenza**.

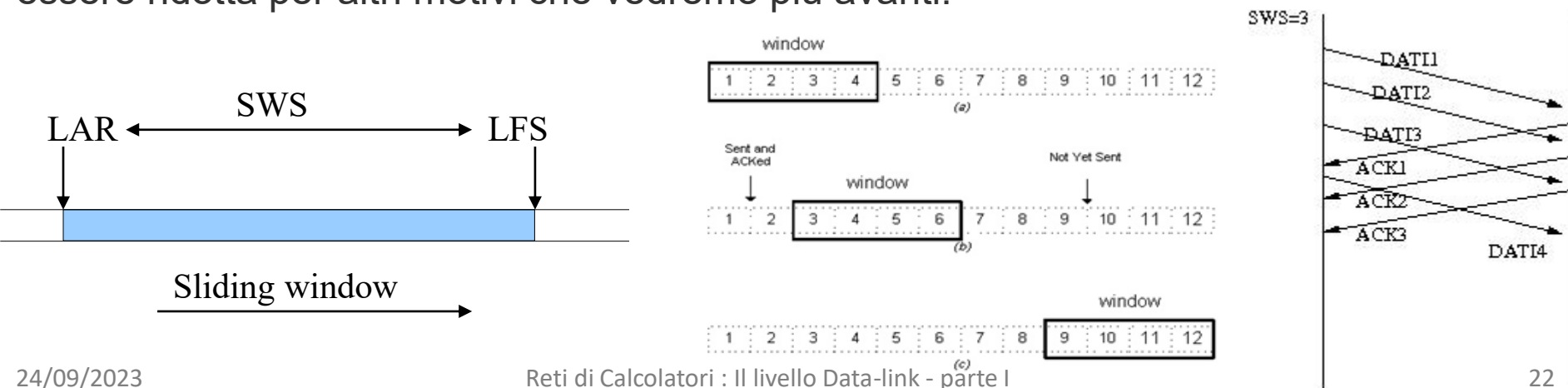
L'indice LFS (Last Frame Sent) contiene il numero dell'ultimo Frame inviato,

L'indice LAR (Last Ack Received) contiene l'indice dell'ultimo ACK ricevuto

Deve valere la regola $LFS - LAR \leq SWS$.

Il destinatario può comunicare al mittente la **finestra del destinatario**, che specifica il numero di dati che il destinatario può ricevere in quel momento. In genere corrisponde allo **spazio libero nel buffer del ricevente**.

La **finestra utilizzata del mittente** non può superare la finestra del destinatario ma può essere ridotta per altri motivi che vedremo più avanti.



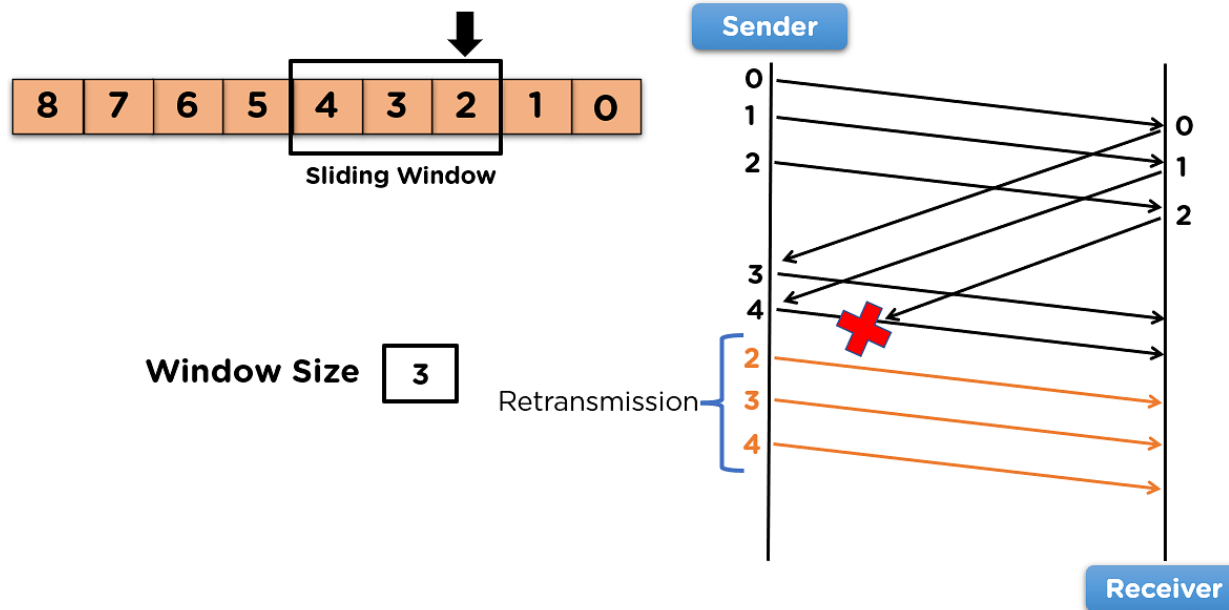
Protocollo per il recupero di errori

Go-Back-N ARQ

Con il protocollo Sliding window può succedere che scada il timer di frame (errato o perduto) quando diversi altri Frame successivi sono stati consegnati correttamente. Se al destinatario arrivano Frame fuori ordine (successivi a Frame non ancora riscontrati) li scarta.

Solo i Frame arrivati correttamente (senza errori e in ordine) vengono mantenuti nel buffer del destinatario fino a quando l'applicazione non li gestisce.

Quando scade il timer del Frame errato/perduto il mittente torna indietro e ricomincia a spedire tutti i Frame a partire da quello errato (**Go-Back-N**)



Protocollo per il recupero degli errori

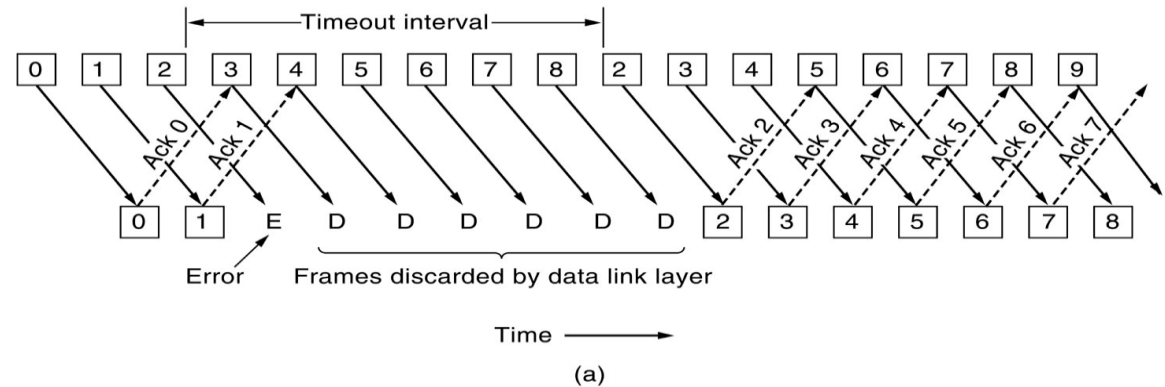
Ripetizione Selettiva

Con il protocollo a **Ripetizione Selettiva** i frame ricevuti correttamente, successivi a quello errato o perduto, vengono bufferizzati dal ricevente il quale sollecita il mittente al reinvio dei frame mancanti, tramite l'invio di un NACK.

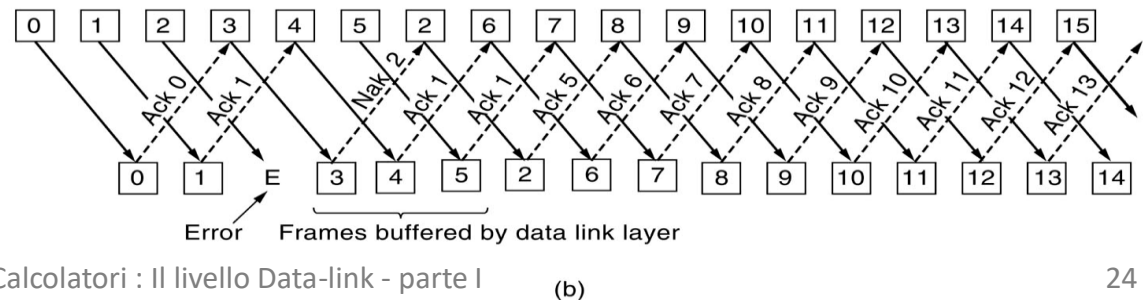
- ▶ Il Mittente rispedisce il Frame richiesto
- ▶ Il Ricevente riscontra il Frame errato e tutti i frame successivi memorizzati nel buffer.

*Nota: Per il funzionamento di questo protocollo il destinatario deve gestire un **buffer** in cui vengono memorizzati i frame ricevuti.*

Esempio GO-BACK-N



Esempio Ripetizione Selettiva



Esempio di protocollo data-link: PPP

Evoluzione di HDLC per Internet. Utilizzato in ADSL.

NOTA: HDLC è un protocollo data link bit-oriented nato per comunicazioni punto-punto o multi-punto, con supporto sia alla modalità non connessa (unNumbered) che connessa.

PPP è protocollo Byte Oriented (Byte Stuffing) ed è definito in RFC 1661

(<http://www.ietf.org/rfc.html>)

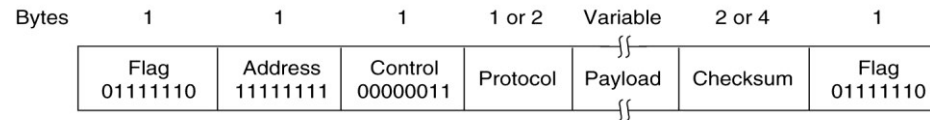
Supporto solo la modalità non connessa (solo UnNumbered)

Supporta vari protocolli dello strato rete (IP, AppleTalk, ..)

Gestisce protocolli ausiliari (LCP e NCP) per l'autenticazione, la configurazione degli indirizzi di rete (IP via DHCP), la concatenazione di diversi link.

I campi dell'header PPP

Il frame PPP aggiunge una intestazione di 6 (o 8) byte al payload, in cui vengono definiti alcuni campi originariamente ideati per HDLC.



Il framing è gestito con il Flag 01111110.

I campi **Address** e **Control** derivano da HDLC e in PPP hanno un valore fisso.

Il campo **Protocol** è stato aggiunto per supportare diversi protocolli a livello rete (IP, IPX, AppleTalk, ..) o protocolli ausiliari (LCP e NCP).

Nota: Il protocollo ausiliario **LCP** è utilizzato per configurare e verificare la connessione a livello data-link e consente di concatenare diversi link PPP.

Il protocollo ausiliario **NCP** serve per configurare i diversi protocolli di livello Network come DHCP per gli indirizzi di rete, per la compressione.

Payload: contiene un numero variabile di byte

MTU (Maximum Transfer Unit) è il payload massimo del protocollo. In PPP l'MTU standard è di 296 byte, ma può essere adattato (vedi PPPoA).

FCS (Gestione Errori): CRC-16, negoziabile fino a CRC-32. Il caso di errori il pacchetto viene scartato senza notifica. In caso di errori eccessivi viene abbattuta la connessione.

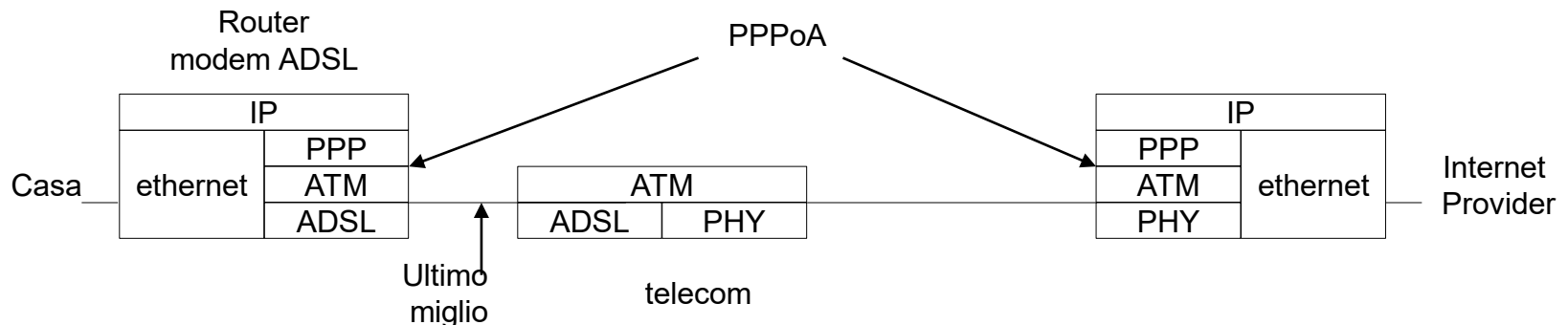
PPP over ATM (PPPoA)

Le telecom forniscono connessioni geografiche utilizzando le proprie reti commutate basate su tecnologia ATM.

ATM è una rete a commutazione di pacchetti, dette celle, di lunghezza fissa di 53 byte, di cui 48 di payload (vedi slide successiva).

Lo standard PPPoA (PPPoA over ATM) definisce le modalità per trasportare pacchetti PPP all'interno di celle ATM. PPP riceve frame Ethernet, per questo utilizza lo stesso MTU di 1500 byte.

Il frame PPP viene suddiviso in celle da 48 byte e riassembleato all'uscita della rete ATM.



Reti a circuito Virtuale: ATM

ATM (Asynchronous Transfer Mode) è una tecnologia, sviluppata da ITU-T (organismo internazionale che si occupa delle trasmissioni telefoniche) a partire dai primi anni 90, che realizza una infrastruttura di rete per trasmissioni a commutazione di pacchetto dedicata al sistema telefonico, ma con l'ambizione di essere utilizzato anche per i le comunicazioni Internet.

Nasce dal mondo della fonia, quindi i principi base dell'architettura sono adattati a questo tipo di esigenza:

- **commutazione di pacchetto a circuito virtuale** (simile alla commutazione di circuito)
- **Qualità del Servizio** (le trasmissioni telefoniche vengono integrate nelle trasmissioni dati, ma hanno diversi requisiti di qualità)
- **pacchetti (celle) di lunghezza fissa di 53 byte**
di cui 5 di intestazione e 48 di payload

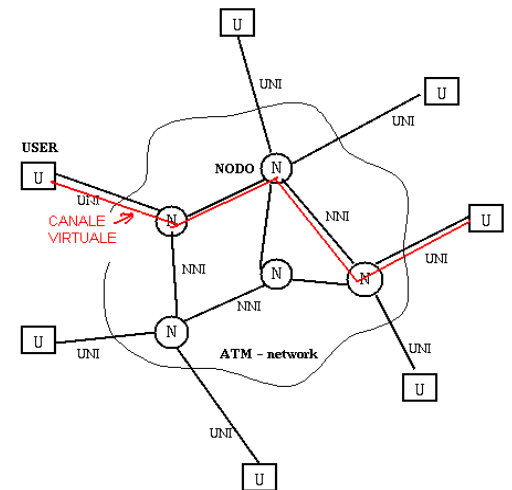
Nota: ogni comunicazione telefonica trasporta la stessa quantità di dati per unità di tempo:

PCM genera 1Byte a 8KHz = 8 KB/s = 64 Kb/s.

Una cella trasporta $48/8K = 6$ ms di conversazione.

ATM non ha avuto successo al di fuori delle reti telefoniche, se non per la realizzazione di reti WAN.

Viceversa la fonia sta diventando sempre più una applicazione di Internet (VoIP).

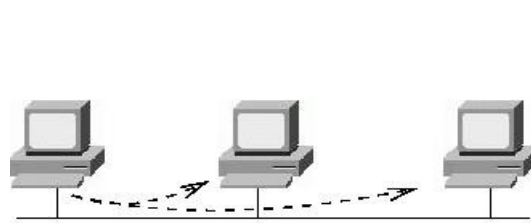


Reti Data-Link: Local Area Network

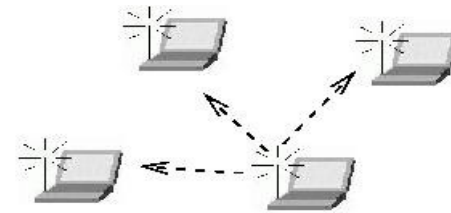
Un canale Multi-Accesso (o canale broadcast) è un canale condiviso per l'accesso diretto tra più terminali ed è il modo più semplice per realizzare una rete di calcolatori a livello Data-Link (LAN) che fanno parte dello stesso **dominio di broadcast** in cui i terminali possono scambiare tra loro messaggio unicast o broadcast.

Le problematiche delle reti LAN richiedono la definizione di un protocollo specifico per:

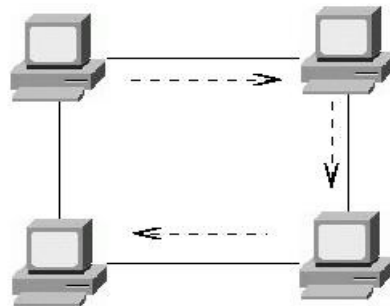
- Disciplinare l'accesso al canale (se fisicamente broadcast)
- Gestire gli indirizzamenti unicast, broadcast e multicast



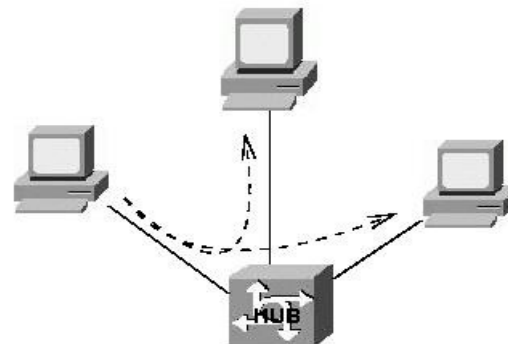
CANALE FISICAMENTE BROADCAST



CANALE FISICAMENTE BROADCAST



CANALE LOGICAMENTE BROADCAST



CANALE LOGICAMENTE BROADCAST

Canali Broadcast: Accesso al canale

Il canale può essere assegnato agli utenti in **modo statico o dinamico**.

Allocazione **statica** del canale

Si può realizzare con tecniche **FDM e TDM** suddividendo la capacità trasmissiva del canale in sotto-canali di numero e dimensione prestabilita

Se il numero di utenti è inferiore al numero di canali ho uno spreco di banda

Se il numero di utenti è superiore alcuni utenti non possono parlare, anche se altri stanno sottoutilizzando il proprio slot.

Questa tecnica è poco efficiente per le Reti Locali in cui gli utenti e le loro esigenze mutano rapidamente.

L'assegnazione del canale nelle principali tecnologie LAN è **dinamica**.

Assegnazione dinamica del canale : Accesso Multiplo

Un singolo canale viene condiviso da N stazioni, ma viene utilizzato solo da chi deve effettivamente inviare dati (assegnazione dinamica).

Nessuna stazione gestisce il canale, ma tutte le stazioni lo devono contendere.

Due possibili modalità di **tempo di trasmissione**:

- ▶ Tempo continuo: la trasmissione può iniziare in qualunque istante.
- ▶ Slotted: Il tempo è diviso in intervalli detti Slot. La trasmissione deve coincidere con l'inizio di un intervallo.

Collisioni: L'accesso a contesa implica che un Frame potrebbe entrare in collisione con un altro. In questo caso entrambi i Frame dovranno essere inviati nuovamente.

Un protocollo che gestisce i tempi di trasmissione e le eventuali collisioni è detto ad **Accesso Multiplo (Multiple Access – MA)**.

Verifica dell'occupazione del canale: in alcuni protocolli MA le stazioni verificano lo stato del canale (**Carrier Sense – CS**) prima di decidere se iniziare la trasmissione.

Alcuni protocolli verificano lo stato del canale anche durante la trasmissione per individuare rapidamente eventuali collisioni (**Collision Detection – CD**)

Protocolli ad Accesso Multiplo

I principali protocolli ad Accesso Multiplo (MA) sono:

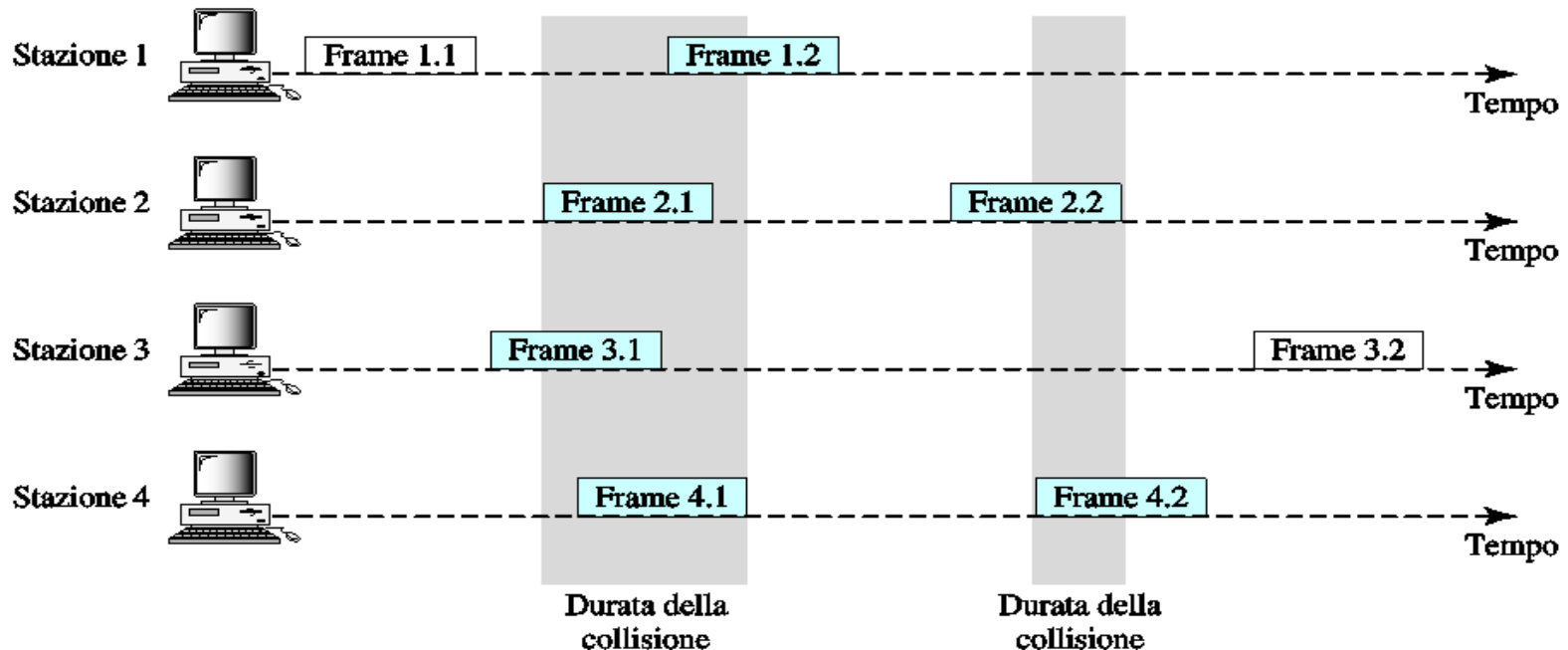
- ▶ MA puro (ALOHA) e slotted (Slotted ALOHA)
- ▶ CSMA persistente e non persistente
- ▶ CSMA/CD (802.3 - Ethernet su Rame o Fibra)
- ▶ CSMA/CA (802.11 - Ethernet Wireless)

ALOHA

ALOHA è il nome del primo protocollo Multiple Access (MA) ideato nei primi anni 70 dall'Università delle Hawaii per connettere via radio Honolulu con le altre isole dell'arcipelago.

ALOHA PURO (Norman Abramson 1970)

Ogni terminale invia i Frame senza accordo con gli altri (MA); l'assenza di conferma viene considerata una collisione con altri trasmettitori, per cui il Frame viene ritrasmesso dopo un **intervallo casuale** di tempo (tempo di backoff).



Algoritmo di Backoff

Il caso di collisione parte un algoritmo (detto algoritmo di Backoff) che determina un tempo di attesa prima di riprovare.

L'algoritmo di Backoff più utilizzato (Ethernet) è l'**esponenziale binario**:

- ▶ Dopo n collisioni consecutive si attende un numero di slot random tra 0 e $2^n - 1$.

Ethernet ammette un valore massimo di $n=10$

Dopo la prima collisione l'invio può avvenire dopo 0 slot (subito) con prob. 50% oppure dopo una attesa di 1 slot con prob. 50%.

Dopo la seconda collisione la trasmissione avviene con probabilità al 25% per i 4 casi [0,1,2,3], e così via.

Nota: Se un host spedisce un Frame in un determinato slot, la probabilità di avere un collisione è data dalla somma delle probabilità di trasmissione degli altri host meno la probabilità del loro verificarsi in contemporanea (per non contarli doppi)

ALOHA: tempo di vulnerabilità

ALOHA PURO (Norman Abramson 1970)

Frame-Time T è il tempo necessario per trasmettere un frame

I Frame hanno lunghezza costante di L bits. $T = L/\text{bitrate}$

Anche con una sovrapposizione di un singolo bit entrambi i frame sono danneggiati.

Il **tempo di vulnerabilità** (intervallo di tempo in cui si può avere una collisione) è $2T$

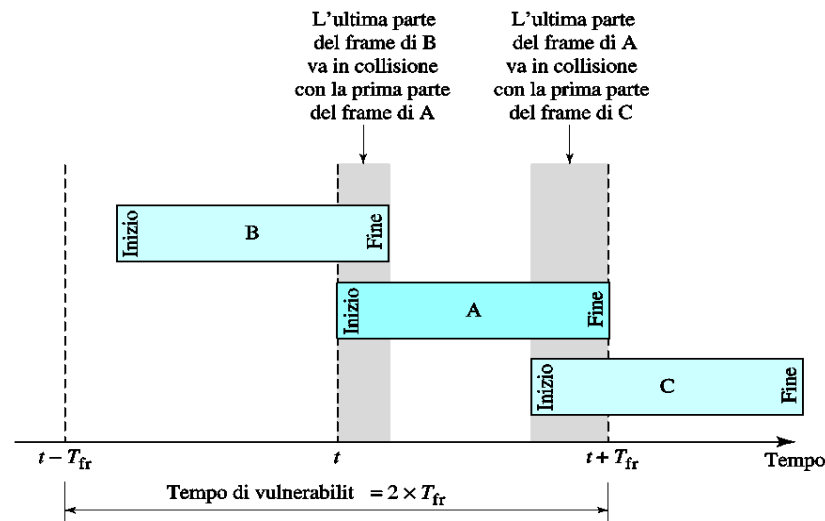
Con N frame generati mediamente nel tempo T :

- ▶ se $0 < N < 1$ ci aspettiamo un throughput ragionevole
- ▶ se $N > 1$ si va rapidamente alla paralisi

Per ogni collisione è necessario rispedire il Frame

G è il carico generato mediamente nel tempo T

$G = N + \text{frame rispediti}$



ALOHA: tempo di vulnerabilità

SLOTTED ALOHA (Roberts 1972)

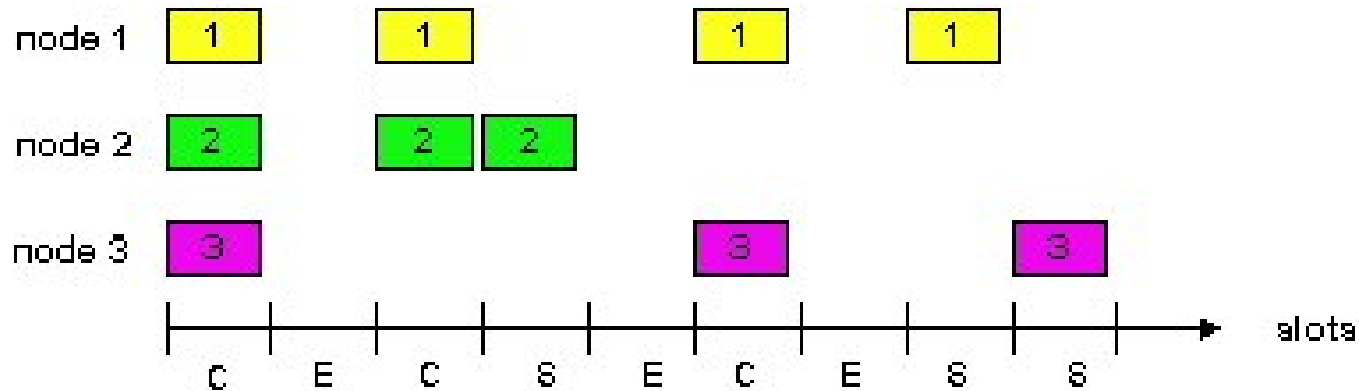
Il tempo viene diviso in intervalli discreti

le trasmissioni possono iniziare solo all'inizio di un intervallo

Una speciale stazione emette un segnale all'inizio

di ogni intervallo per sincronizzare trasmettitori

Il tempo di vulnerabilità è T (dimezzato rispetto a Pure Aloha)



ALOHA Throughput

Qualunque sia il carico G che si presenta (pacchetti trasmessi nel tempo T), la capacità di trasporto S (Throughput) è G volte la probabilità P_0 (trasmissione con successo nel **tempo di vulnerabilità**): $S = G P_0$

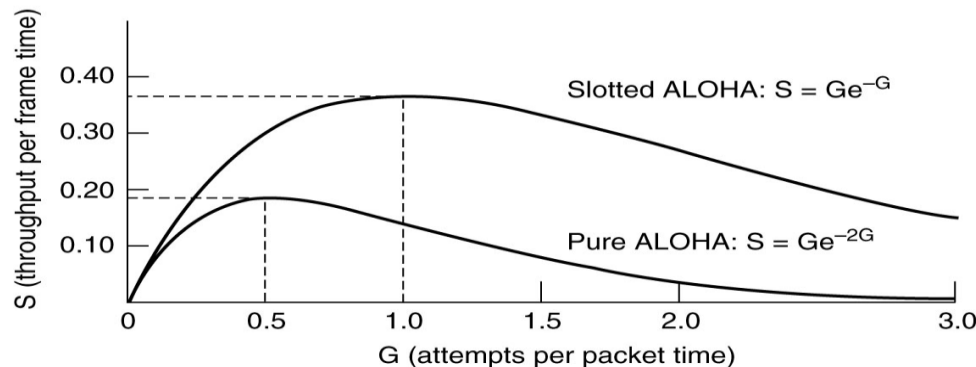
La probabilità che k frame siano generati durante il tempo T è dato dalla **distribuzione di Poisson**: $P[k] = G^k e^{-G} / k!$

ALOHA PURO: Per un periodo di vulnerabilità pari a $2T$ la probabilità che nessun altro frame venga generato durante il periodo di vulnerabilità:

$$P[0] = G^0 e^{-2G} / 0! = e^{-2G} \quad \text{Throughput } S = G e^{-2G} \quad \text{Max } S = 0.18 \text{ per } G = 0.5$$

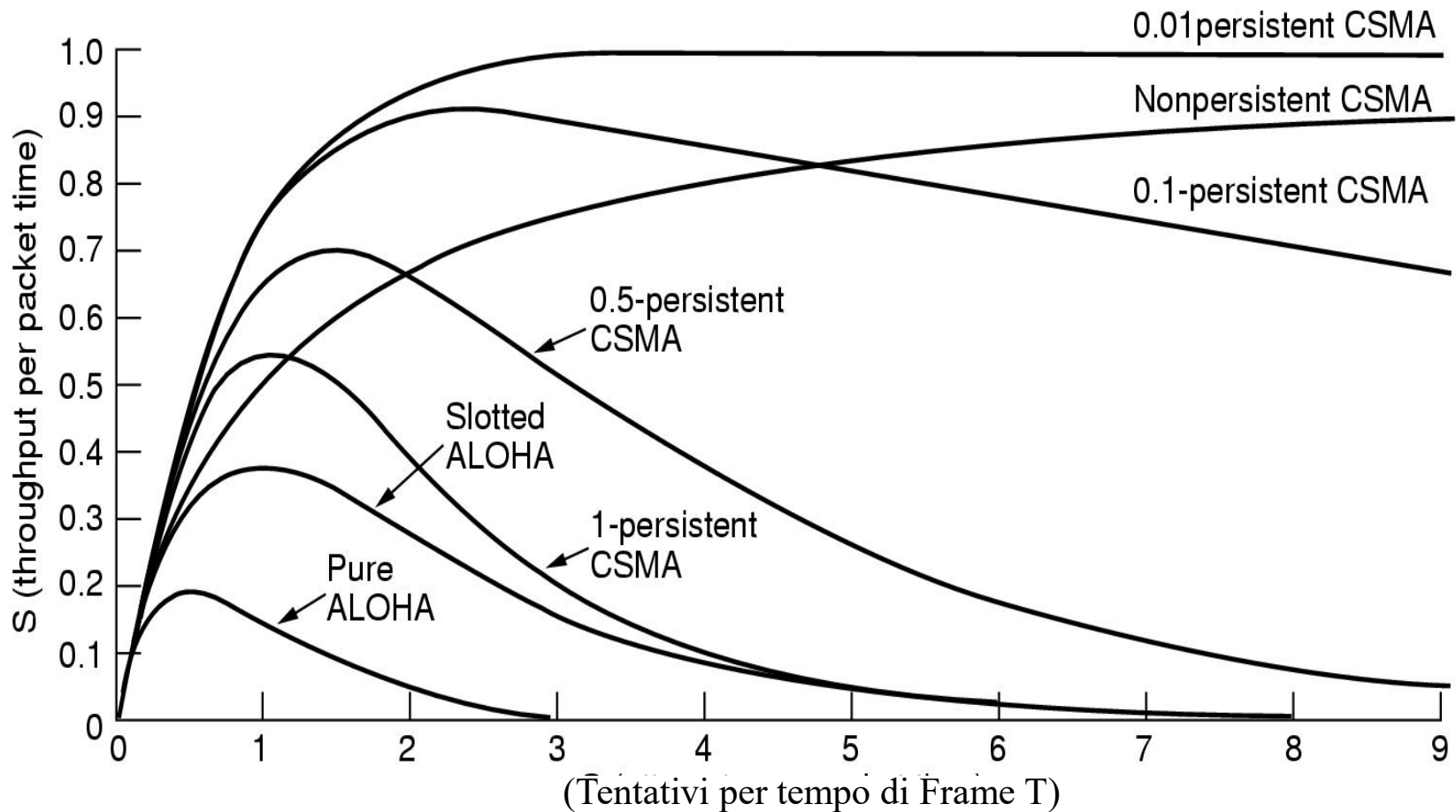
ALOHA SLOTTED: Il tempo di vulnerabilità è T (dimezzato rispetto a Pure Aloha)

$$\text{Throughput } S = G e^{-G} \quad \text{Max } S = 0.36 \text{ per } G = 1$$



Throughput

La figura mostra l'andamento del throughput mettendo a confronto diverse tecniche:



CSMA

CSMA (Carrier Sense Multiple Access)

migliora le prestazioni aggiungendo l'ascolto del canale: se il canale è occupato pospone la trasmissione

Il numero di collisione è molto ridotto (ma non azzerato) $G \approx N$

L'algoritmo che determina quando ritentare è fondamentale.

CSMA con persistenza

CSMA non persistente

se il canale è libero inizia la trasmissione altrimenti attende un tempo casuale prima di ritentare (anche in assenza di collisioni). Diminuisce la probabilità di collisione poiché è improbabile che 2 stazioni aspettino lo stesso tempo, ma aumenta il ritardo di trasmissione (anche in una rete con poco traffico).

CSMA 1-Persistente (utilizzato da Ethernet)

se il canale è libero inizia la trasmissione

altrimenti attende che si liberi prima di ritentare.

E' detto 1-persistente perché trasmette con probabilità 1 quando il canale è libero.

Problema: in caso di alto traffico è probabile che 2 nodi in attesa entrino in collisione.

CSMA p-persistente: Si applica ai canali divisi in intervalli temporali.

Se il canale è libero la trasmissione avviene con **probabilità p** e viene rimandata all'intervallo successivo con probabilità $1-p$.

Se anche questo è libero la trasmissione avviene con **probabilità p** e così via.

Se il canale è occupato si comporta come se ci fosse stata una collisione:

parte un algoritmo di Backoff (generalmente l'attesa è proporzionale al numero di collisioni consecutive)

Al crescere di p diminuisce il ritardo, ma aumenta la probabilità di collisione

CSMA/CD

CSMA/CD (Carrier Sense Multiple Access - Collision Detect)

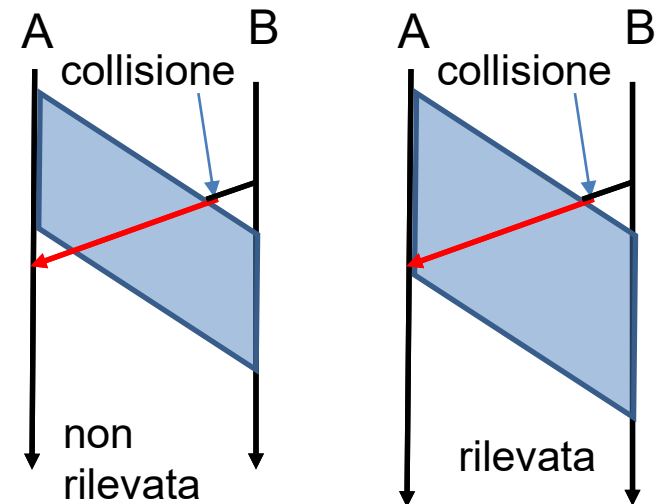
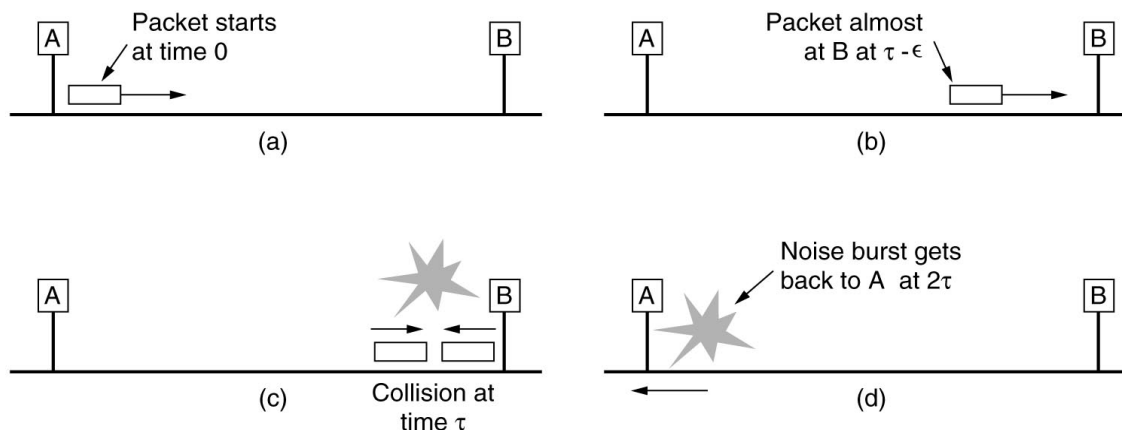
Chi spedisce rimane in ascolto del canale anche durante la trasmissione.

Vantaggi:

- in caso di collisione si interrompe la trasmissione → si riduce il tempo di vulnerabilità
- il mittente capisce se il Frame è stato inviato correttamente (senza collisioni)

Se T_{pr} è il tempo di propagazione del cavo, il massimo ritardo nell'individuare una collisione è $2T_{pr}$ (supponendo che il secondo nodo all'altro estremo inizi la trasmissione un attimo prima di ricevere il pacchetto)

Per individuare con certezza una collisione è quindi necessario che il frame abbia un tempo di trasmissione $T_{tr} \geq 2T_{pr}$



Dominio di Collisioni e dominio di Broadcast

L'insieme dei nodi che concorrono per accedere allo stesso mezzo trasmissivo costituisce un **Dominio di Collisione** (Collision Domain).

Il **dominio di Broadcast** è l'insieme dei nodi che possono comunicare direttamente, senza dover risalire al livello rete.

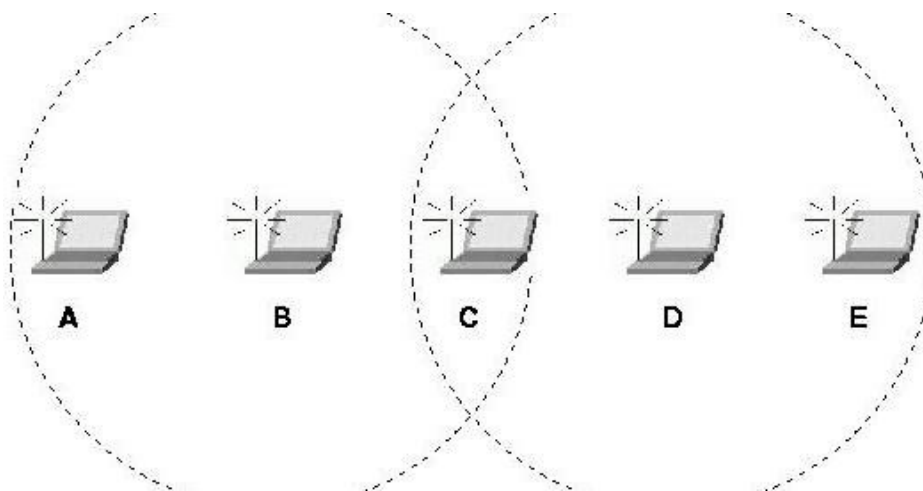
I due domini possono non coincidere per effetto di apparati di rete (Bridge) che separano i domini di collisione ma non i domini di broadcast.

Protocolli LAN Wireless

Nelle reti Wireless il dominio di collisione non è nettamente definito come nelle reti wired:

Problema del nodo nascosto: B trasmette a C. D non sente il segnale di B e trasmette contemporaneamente a B creando collisione non rilevata da B.

Problema del nodo esposto: B trasmette ad A. C vorrebbe trasmettere a D ma non lo fa perché crede erroneamente di creare una collisione.



La soluzione consiste nell'evitare le collisioni (Collision Avoidance) attraverso un opportuno protocollo (CSMA/CA) .

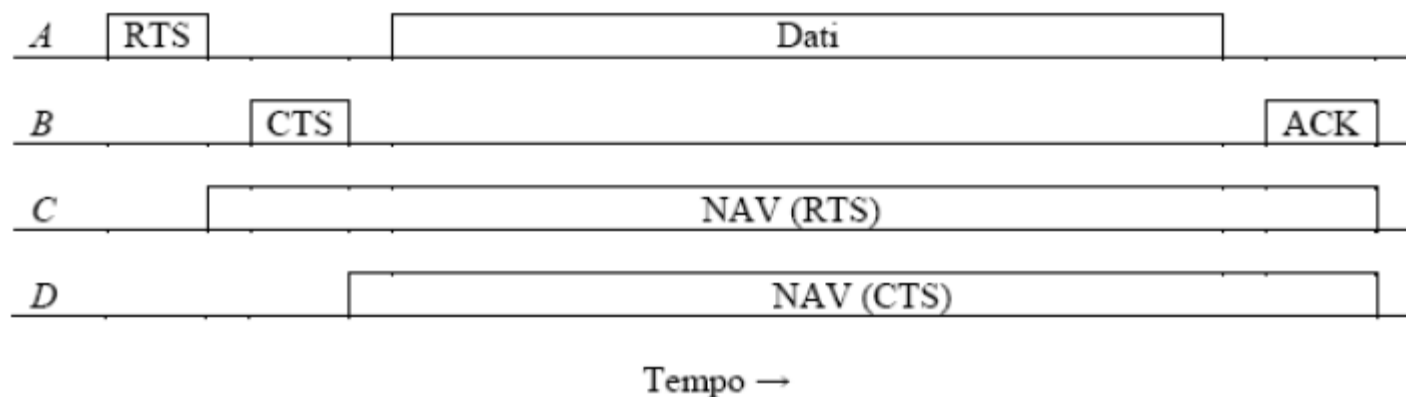
Protocolli LAN Wireless: CSMA/CA

Nel protocollo CSMA/CA (CSMA with Collision Avoidance) il trasmettitore incita il ricevitore (con un Frame RTS) a trasmettere un piccolo Frame (CTS) il modo che le stazioni che si trovano alla sua portata evitino di inviare dati. Sia RTS che CTS contengono il tempo necessario per la trasmissione.

Le stazioni che ricevono RTS e CTS attivano un Carrier Sense Virtuale detto NAV (Network Allocation Vector) che è un contatore che viene decrementato e rappresenta il tempo in cui devono considerare indisponibile in canale.

Il ricevitore invia un pacchetto ACK dopo ogni Frame ricevuto con successo. Eventuali collisioni di pacchetti RTS sono comunque possibili e sono gestite con il protocollo CSMA.

Questo protocollo è utilizzato nelle reti WiFi (IEEE 802.11) e WiMax (802.16)



Protocolli LAN Wireless: CSMA/CA

La stazione che deve trasmettere (A) valuta se il mezzo è libero consultando sia NAV che il mezzo reale.

Se il canale è considerato libero emette un RTS altrimenti avvia la procedura di Backoff Esponenziale Binario.

Quando il ricevente (B) riceve l'RTS risponde con un CTS.

Se A non riceve il CTS di B avvia la procedura di Backoff con tempo raddoppiato, altrimenti inizia la trasmissione.

Se dopo l'invio del dato A non riceve un ACK entro un tempo stabilito deve ripetere la procedura.

