



Reti di Calcolatori

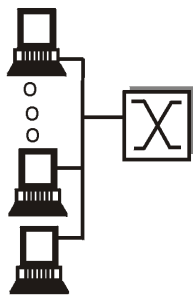
Protocolli data link layer per reti LAN



Collegamenti di rete

Esistono due tipi di collegamenti di rete:

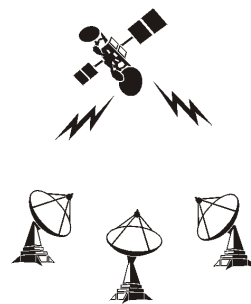
- **Collegamento punto-punto (PPP)**
 - Impiegato per connessioni telefoniche.
 - Collegamenti punto-punto tra Ethernet e host.
- **Collegamento broadcast** (cavo o canale condiviso)
 - Ethernet tradizionale
 - HFC in upstream
 - Wireless LAN 802.11



shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)



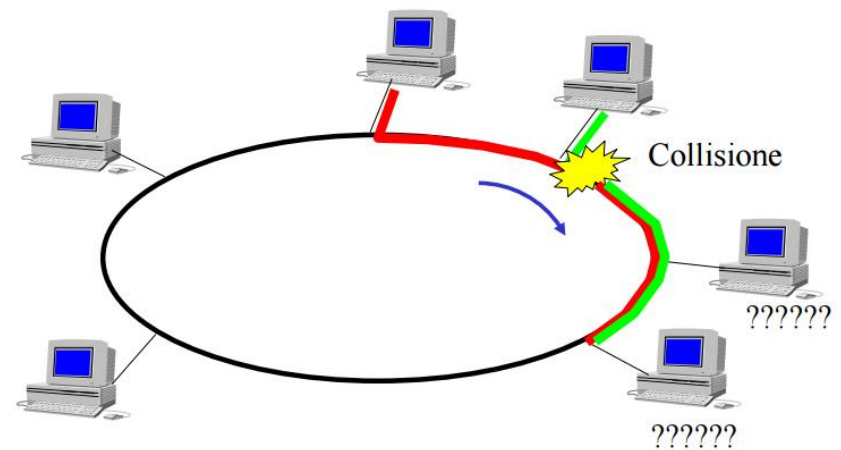
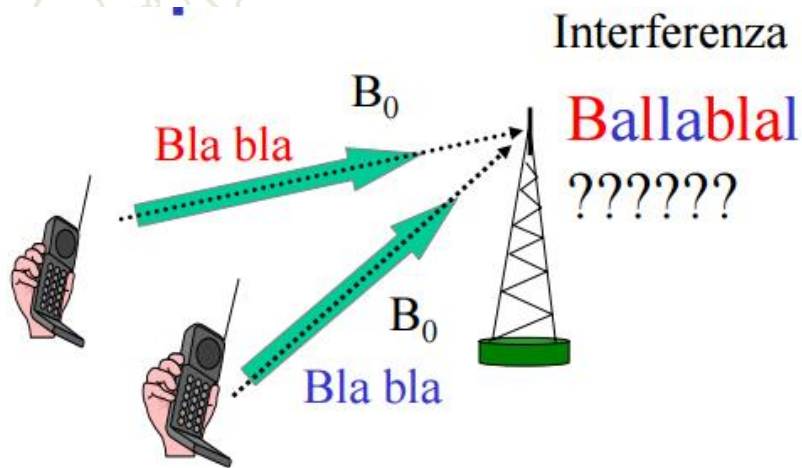
satellite



cocktail party

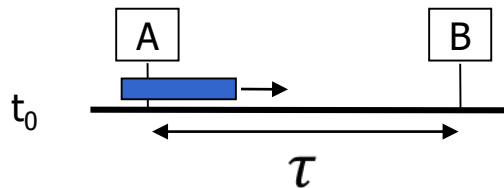
Interferenza e Collisione

- Connessione a un canale broadcast condiviso.
- Centinaia o anche migliaia di nodi possono comunicare direttamente su un canale broadcast:
 - Si genera una *collisione* quando i nodi ricevono due o più frame contemporaneamente.

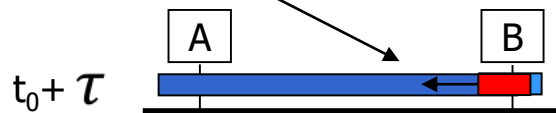


Ritardi di linea e collisioni

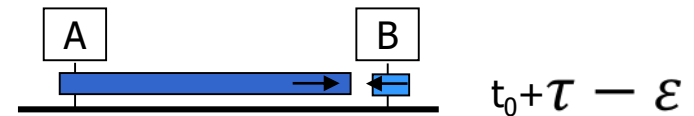
- Sia τ il tempo di propagazione fra le stazioni più lontane
- Il tempo massimo per la rilevazione di una collisione è 2τ slot di contesa)



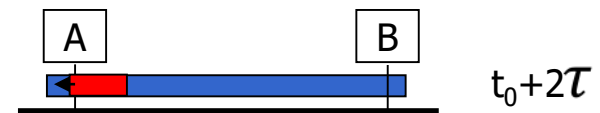
Il pacchetto deve durare più di 2τ



B rileva la collisione e sospende la trasmissione. Il disturbo generato dalla collisione si propaga verso A

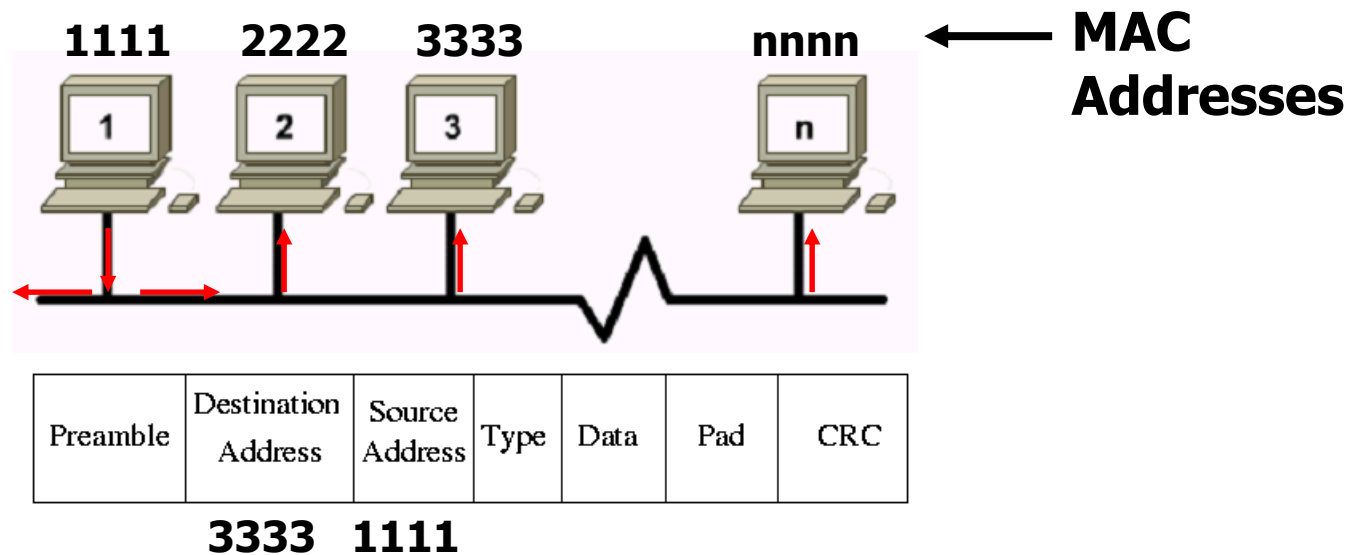


B inizia a trasmettere poco prima che l'inizio del pacchetto di A arrivi in B



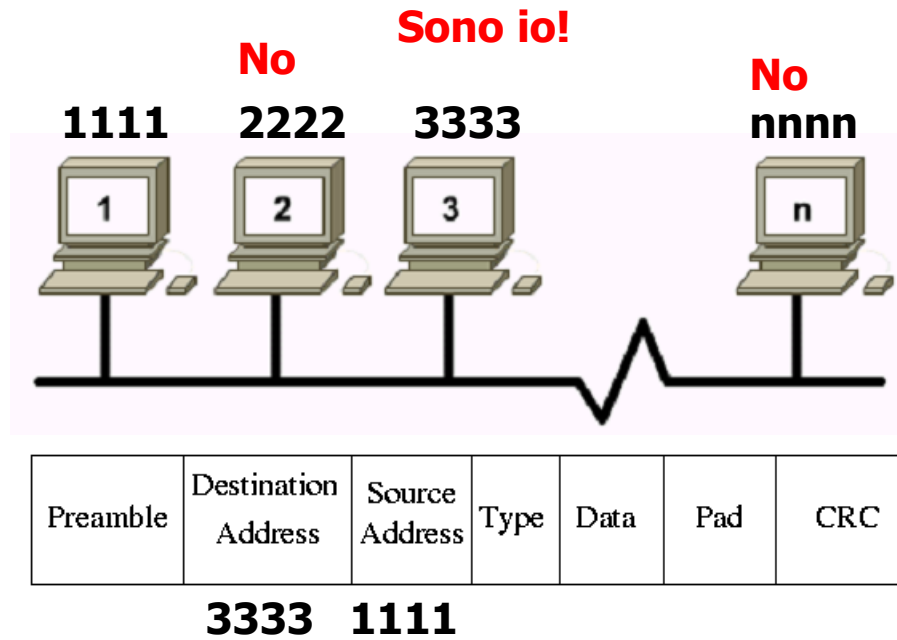
A rileva la collisione e sospende la trasmissione.

Condivisione su Bus



- Quando una trama viene inviata sul mezzo condiviso tutti i dispositivi collegati la ricevono.
- Cosa ne fanno?

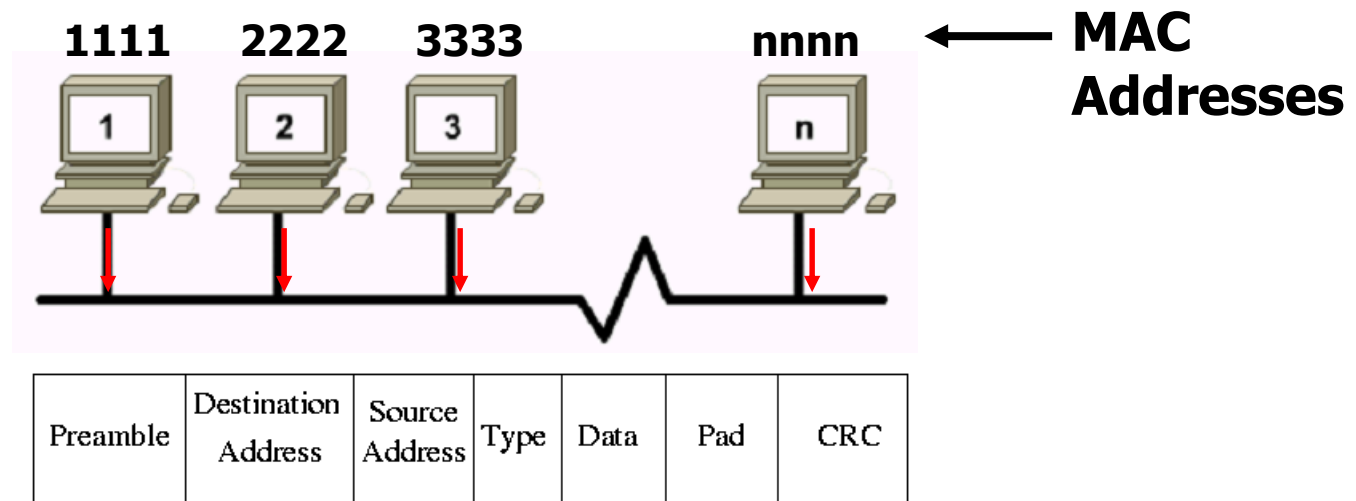
Condivisione su Bus



Abbreviated MAC Addresses

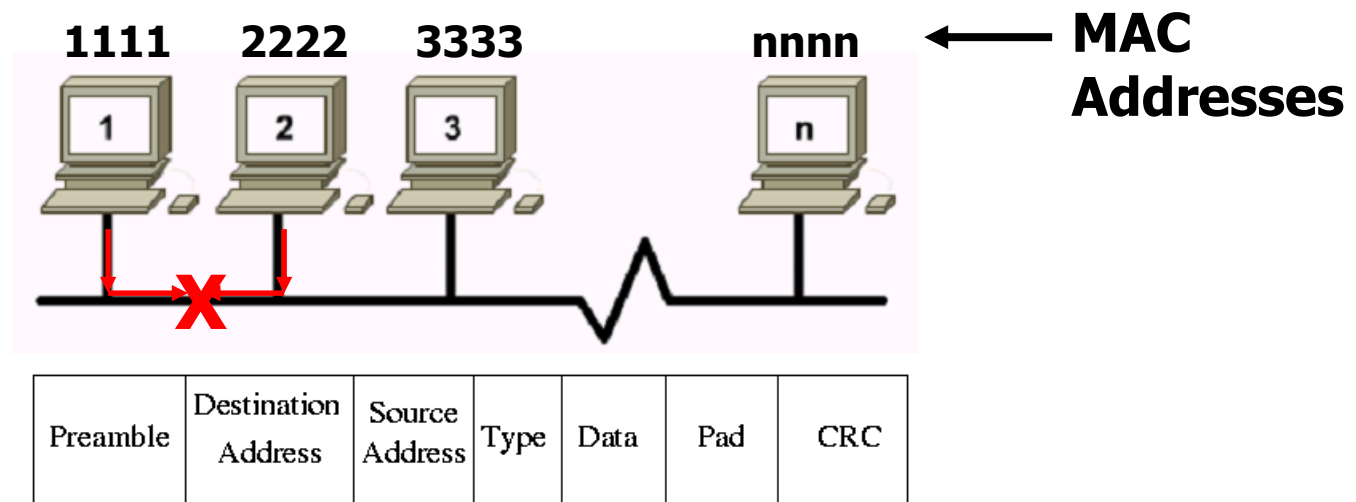
- Ogni NIC confronta il suo MAC address con quello il destination address sulla trama.
- Se corrispondono, copia il resto della trama.
- Altrimenti, la ignora, a meno che la NIC non operi in **modalità promiscua** (Il NIC fa passare ogni trama)

Condivisione su Bus



- Ma che accade quando più stazioni tentano di trasmettere allo stesso tempo?

Condivisione su Bus



Una Collisione!

Protocolli di accesso multiplo

- Connessione a un canale broadcast condiviso.
- Centinaia o anche migliaia di nodi possono comunicare direttamente su un canale broadcast:
 - Si genera una *collisione* quando i nodi ricevono due o più frame contemporaneamente.

Protocolli di accesso multiplo

- Protocolli che fissano le modalità con cui i nodi regolano le loro trasmissioni sul canale condiviso.
- La comunicazione relativa al canale condiviso deve utilizzare lo stesso canale!
 - non c'è un canale “out-of-band” per la coordinazione

Protocolli di accesso multiplo ideali

Canale broadcast con velocità di R bit al sec:

1. Quando un nodo deve inviare dati, questo dispone di un tasso trasmissivo pari a R bps.
2. Quando M nodi devono inviare dati, questi dispongono di un tasso trasmissivo pari a R/M bps.
3. Il protocollo è decentralizzato:
 - non ci sono nodi master
 - non c'è sincronizzazione dei clock
4. Il protocollo è semplice.

Protocolli di accesso multiplo

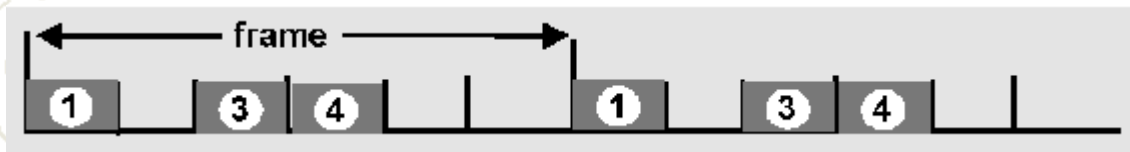
Si possono classificare in una di queste tre categorie:

- **Protocolli a suddivisione del canale** (*channel partitioning*)
 - Suddivide un canale in “parti più piccole” (slot di tempo, frequenza, codice).
- **Protocolli ad accesso casuale** (*random access*)
 - I canali non vengono divisi e si può verificare una collisione.
 - I nodi coinvolti ritrasmettono ripetutamente i pacchetti.
- **Protocolli a rotazione** (“*taking-turn*” o “collision-free”)
 - Ciascun nodo ha il suo turno di trasmissione, ma i nodi che hanno molto da trasmettere possono avere turni più lunghi.

Protocolli a suddivisione del canale: TDMA

TDMA: accesso multiplo a divisione di tempo.

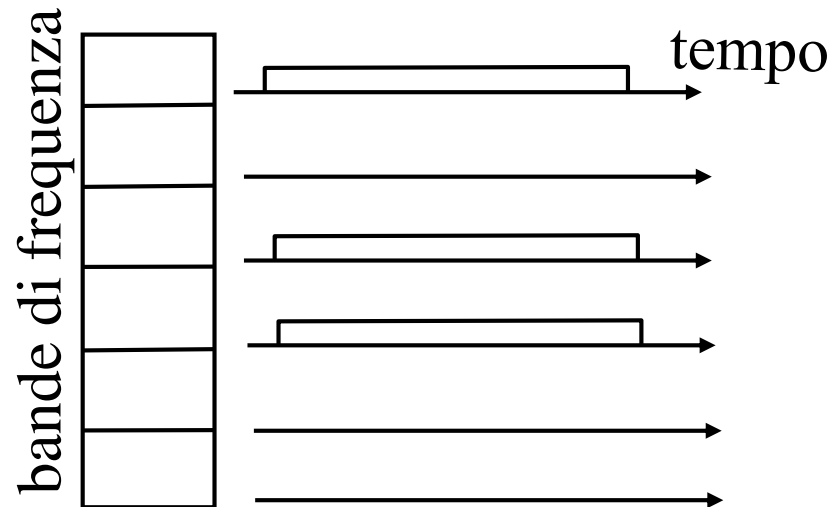
- Suddivide il canale condiviso in *intervalli di tempo*.
- Gli slot non usati rimangono inattivi
- Esempio: gli slot 1, 3 e 4 hanno un pacchetto, 2, 5 e 6 sono inattivi.



Protocolli a suddivisione del canale: FDMA

FDMA: accesso multiplo a divisione di frequenza.

- Suddivide il canale in bande di frequenza.
- A ciascuna stazione è assegnata una banda di frequenza prefissata.
- Esempio: gli slot 1, 3 e 4 hanno un pacchetto, 2, 5 e 6 sono inattivi.



Protocolli ad accesso casuale

- Quando un nodo deve inviare un pacchetto:
 - trasmette sempre alla massima velocità consentita dal canale, cioè R bps
 - non vi è coordinazione a priori tra i nodi
- Due o più nodi trasmittenti → “collisione”
- Il protocollo ad accesso casuale definisce:
 - Come rilevare un’eventuale collisione.
 - Come ritrasmettere se si è verificata una collisione.
- Esempi di protocolli ad accesso casuale:
 - slotted ALOHA
 - ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

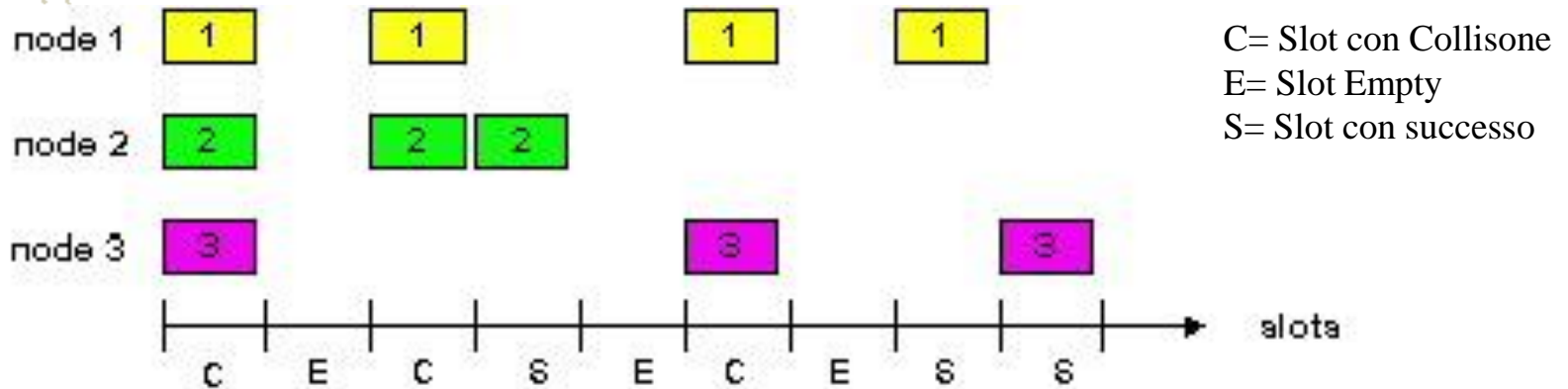
Assumiamo che:

- Tutti i pacchetti hanno la stessa dimensione.
- Il tempo è suddiviso in slot; ogni slot equivale al tempo di trasmissione di un pacchetto.
- I nodi iniziano la trasmissione dei pacchetti solo all'inizio degli slot.
- I nodi sono sincronizzati.
- Se in uno slot due o più pacchetti collidono, i nodi coinvolti rilevano l'evento prima del termine dello slot.

Operazioni:

- Quando a un nodo arriva un nuovo pacchetto da spedire, il nodo attende fino all'inizio dello slot successivo.
- *Se non si verifica una collisione:* il nodo può trasmettere un nuovo pacchetto nello slot successivo.
- *Se si verifica una collisione:* il nodo la rileva prima della fine dello slot e ritrasmette con probabilità p il suo pacchetto durante gli slot successivi.

Slotted ALOHA



Pro

- Consente a un singolo nodo di trasmettere continuamente pacchetti alla massima velocità del canale.
- È fortemente decentralizzato, ciascun nodo rileva le collisioni e decide indipendentemente quando ritrasmettere.
- È estremamente semplice.

Contro

- Una certa frazione degli slot presenterà collisioni e di conseguenza andrà "sprecata".
- Un'alta frazione degli slot rimane vuota, quindi inattiva.

L'efficienza di Slotted Aloha

L'**efficienza** è definita come la frazione di slot vincenti in presenza di un elevato numero di nodi attivi, che hanno sempre un elevato numero pacchetti da spedire.

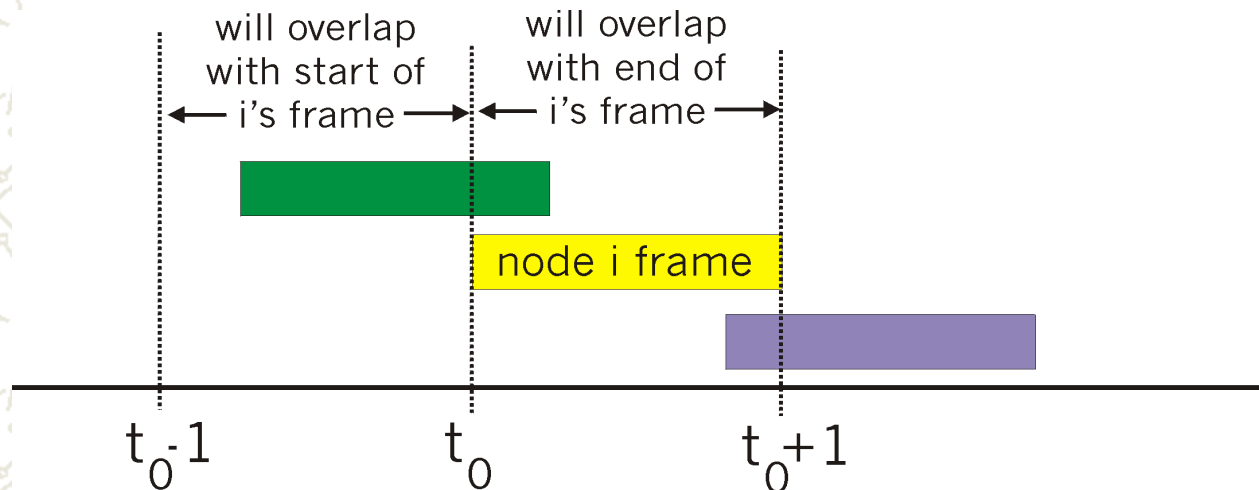
- Supponiamo N nodi con pacchetti da spedire, ognuno trasmette i pacchetti in uno slot con probabilità p .
- La probabilità di successo di un dato nodo = $p(1-p)^{N-1}$
- La probabilità che un nodo arbitrario abbia successo
= $Np(1-p)^{N-1}$

- Per ottenere la massima efficienza con N nodi attivi, bisogna trovare il valore p^* che massimizza $Np(1-p)^{N-1}$
- Per un elevato numero di nodi, ricaviamo il limite di $Np^*(1-p^*)^{N-1}$ per N che tende all'infinito, e otterremo $1/e = 0,37$

Nel caso migliore:
solo il 37% degli slot
compie lavoro utile.

ALOHA puro

- Aloha puro: più semplice, non sincronizzato.
- Quando arriva il primo pacchetto:
 - lo trasmette immediatamente e integralmente nel canale broadcast.
- Elevate probabilità di collisione:
 - Il pacchetto trasmesso a t_0 si sovrappone con la trasmissione dell'altro pacchetto inviato in $[t_0-1, t_0+1]$.



L'efficienza di Aloha puro

$P(\text{trasmissione con successo da un dato nodo}) = P(\text{il nodo trasmette}) \cdot$

$P(\text{nessun altro nodo trasmette in } [t_0-1, t_0]) \cdot$

$P(\text{nessun altro nodo trasmette in } [t_0, t_0+1])$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$

... scegliendo p migliore e lasciando $n \rightarrow \text{infinito}$...

$$= 1/(2e) = 0,18$$

Peggior di prima !

Accesso multiplo a rilevazione della portante (CSMA)

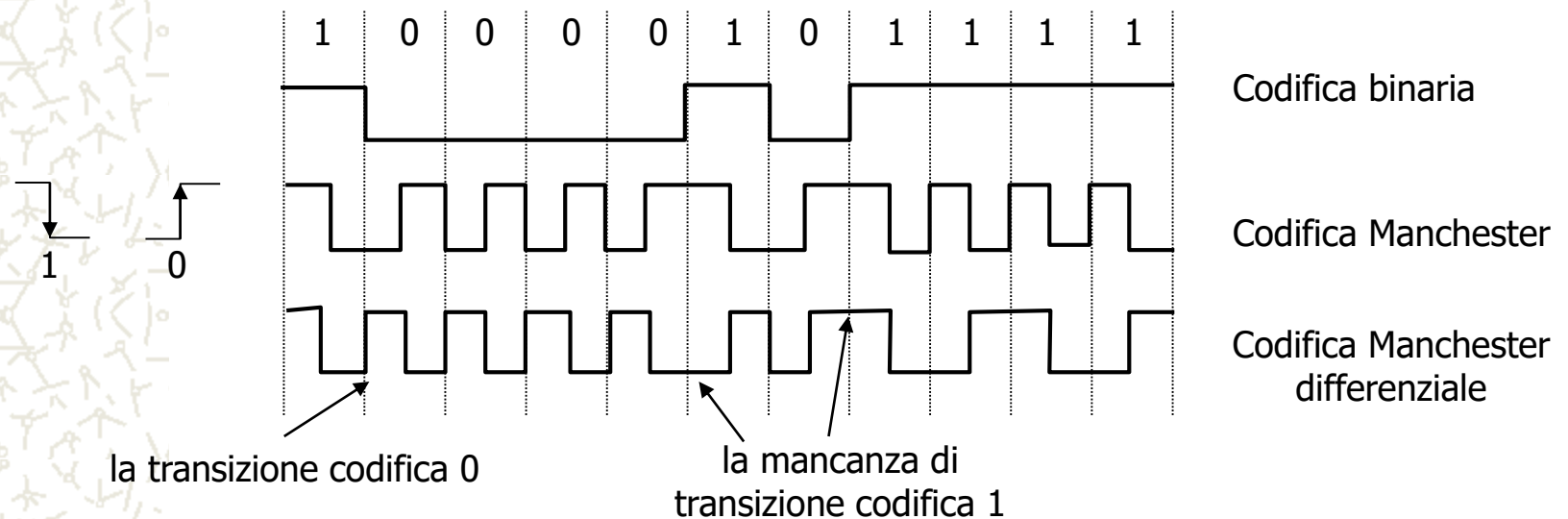
CSMA: Carrier Sense Multiple Access

CSMA: si pone in ascolto prima di trasmettere:

- Se rileva che il canale è libero, trasmette l'intero pacchetto.
- Se il canale sta già trasmettendo, il nodo aspetta un altro intervallo di tempo.
- Analogia: se qualcun altro sta parlando, aspettate finché abbia concluso!

Rilevazione della portante

- Si utilizza la **codifica Manchester** per i segnali
i bit sono codificati da transizioni (usa il doppio della banda)
- I livelli in presenza di segnale sono standardizzati
IEEE 802.3: -0.85, +0.85
- L'assenza di portante è codificata dal segnale nullo (linea **idle**)



CSMA con trasmissioni in collisione

Le collisioni *possono* ancora verificarsi:

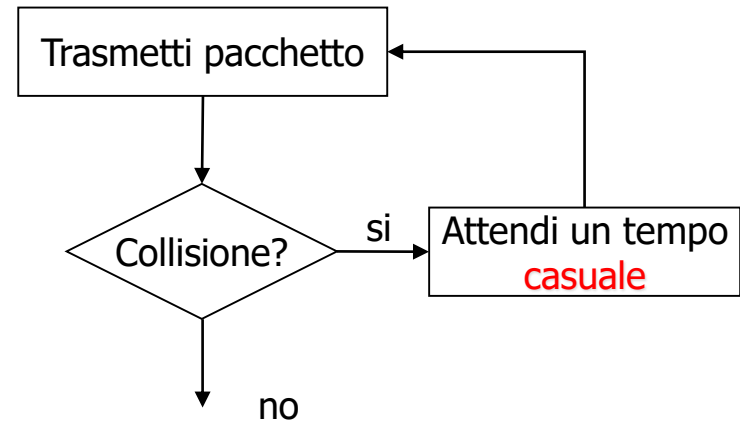
Il ritardo di propagazione fa sì che due nodi non rilevino la reciproca trasmissione

collisione:

Quando un nodo rileva una collisione, cessa immediatamente la trasmissione.

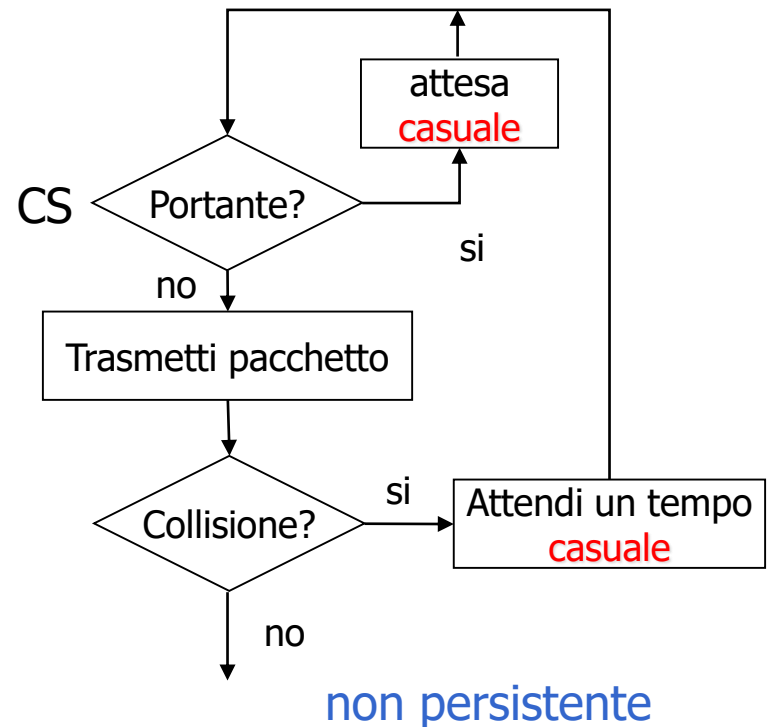
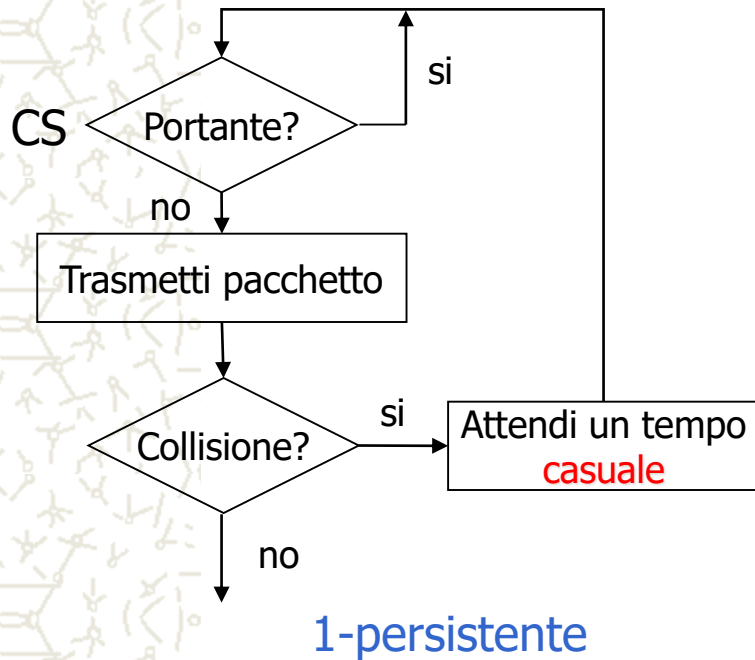
nota:

La distanza e il ritardo di propagazione giocano un ruolo importante nel determinare la probabilità di collisione.



La collisione viene rilevata ascoltando il canale e verificando che il segnale ricevuto corrisponda a quello trasmesso senza interferenze. Basta una minima sovrapposizione dei due pacchetti per farli andare persi.

CSMA: persistenza



CSMA 1-persistente

- Il più semplice di questi protocolli ha il seguente funzionamento:
 - quando un calcolatore ha dati da trasmettere, ascolta il **segnale presente** sul mezzo trasmissivo
 - se trova il canale **libero**, **trasmette** il frame
 - se trova il canale **occupato**, **continua ad ascoltare** fino a che il canale non si libera, e poi trasmette il frame
 - in caso di **collisione**, la stazione aspetta un tempo **casuale** e ripete l'algoritmo
- Il protocollo si chiama **1-persistente** perchè quando trova il canale occupato, resta in ascolto **continuamente**, ed appena il canale si libera trasmette con **probabilità 1** (sempre)

CSMA 1-persistente (cont.)

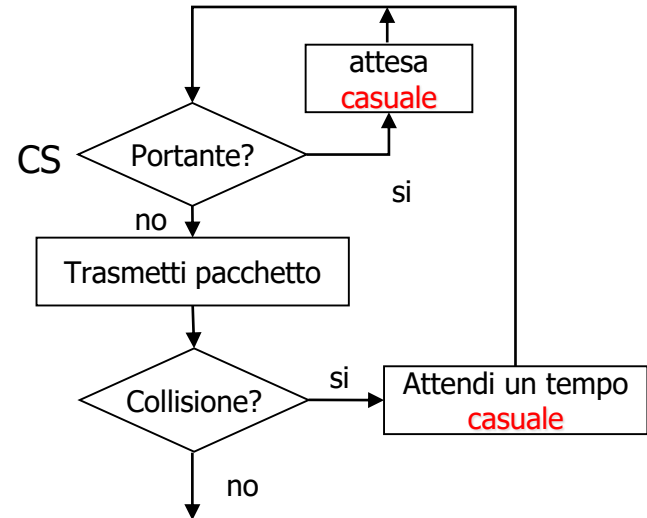
- Con questo protocollo acquista grande importanza il **ritardo di propagazione** del segnale tra due stazioni
 - infatti, quando una stazione **inizia** a trasmettere, una seconda stazione **potrebbe** voler trasmettere, ed ascolta il canale
 - se il segnale trasmesso dalla **prima** stazione non ha ancora avuto **il tempo di propagarsi** fino alla seconda stazione, questa troverà il canale libero e trasmetterà, generando una **collisione**
- Maggiore è il ritardo di propagazione, più **numerose** saranno le collisioni dovute alla eventualità sopra descritta
 - nota: questa situazione si presenterà **sempre** ed indipendentemente dal ritardo di propagazione qualora **due stazioni** volessero trasmettere mentre una terza stà trasmettendo: alla fine della trasmissione della terza stazione, le due stazioni in attesa si metteranno **sempre** a trasmettere **contemporaneamente**

CSMA 1-persistente (cont.)

- Come slotted aloha, questo protocollo **non interferisce** con le trasmissioni già in atto
- A differenza di slotted aloha, questo protocollo **non prevede** di dover attendere la time slot successiva, evitando ad esempio di lasciare **inutilizzata** una slot temporale per il tempo di durata della slot stessa
- Inoltre CSMA 1-persistente **non richiede** la **sincronizzazione** delle stazioni connesse alla rete

CSMA non persistente

- Si differenzia dal precedente per il fatto che una stazione, quando **vuole trasmettere** ma trova il canale **occupato**, **non resta** ad ascoltare in continuazione, ma attende un **tempo casuale** e riprova
- Questo meccanismo **riduce sensibilmente** le collisioni dovute al fatto che due stazioni vogliano trasmettere durante la trasmissione di una terza:
 - ora le stazioni attenderanno generalmente **tempi diversi** prima di ritentare
 - la prima che ritenta troverà il canale **libero** e trasmetterà
 - la seconda troverà **nuovamente** il canale occupato, quindi non interferirà ed **aspetterà ancora**



- Questo protocollo **alza notevolmente** l'efficienza di utilizzo del canale con **l'aumento del carico**, cioè delle stazioni connesse alla rete
- Il problema principale di questo protocollo è che in condizioni di **elevato carico** il tempo che intercorre tra l'istante in cui la stazione vuole trasmettere e l'istante in cui riesce a trasmettere può crescere **enormemente**

CSMA p-persistente

- In questa ultima versione del protocollo a rilevamento della portante, il tempo è **suddiviso in slot temporali** come nello slotted aloha
- In questo caso, chi desidera trasmettere **ascolta** il canale continuamente e quando lo trova libero
 - trasmette con **probabilità p**, oppure attende la slot **successiva** con probabilità $(1-p)$
 - alla slot successiva, **se libera**, trasmette **nuovamente** con probabilità p o aspetta la successiva con probabilità $1-p$, e così via
 - in caso di **collisione**, o se durante i tentativi di trasmissione qualche altra stazione inizia a trasmettere, la stazione attende un **tempo casuale** e ripete l'algoritmo
- Questo protocollo è una via di mezzo tra il protocollo 1-persistente (a cui tende per p che tende ad 1) e quello **non persistente**
- Come nel caso di CSMA non persistente, ad **elevato carico** e per **bassi** valori di p **cresce l'efficienza** di utilizzo della linea ma cresce il **ritardo** di trasmissione rispetto all'arrivo dei dati dallo strato di rete
- Per **alti valori di p** l'efficienza di utilizzo della linea **descende** rapidamente con l'aumentare del carico

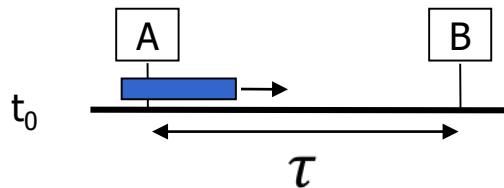
CSMA/CD [Collision Detection]

Il protocollo opera in tre diverse fasi:

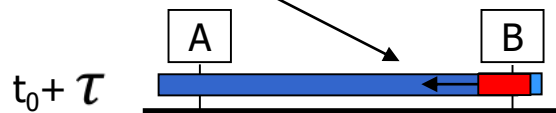
- **carrier sense**: (rilevazione della trasmissione): ogni stazione che deve trasmettere ascolta il bus e decide di trasmettere solo se questo è libero (*listen before talking*);
- **multiple access**: nonostante il carrier sense è possibile che due stazioni, trovando il mezzo trasmissivo libero, decidano contemporaneamente di trasmettere; la probabilità di questo evento è aumentata dal fatto che il tempo di propagazione dei segnali sul cavo non è nullo, e quindi una stazione può credere che il mezzo sia ancora libero anche quando un'altra ha già iniziato la trasmissione;
- **collision detection**: se si verifica la sovrapposizione di due trasmissioni si ha una "collisione"; per rilevarla, ogni stazione, mentre trasmette un pacchetto, ascolta i segnali sul mezzo trasmissivo, confrontandoli con quelli da lei generati (*listen while talking*).

Ritardi di linea e collisioni

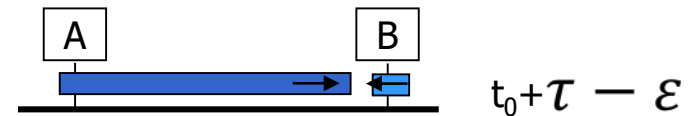
- Sia τ il tempo di propagazione fra le stazioni più lontane
- Il tempo massimo per la rilevazione di una collisione è 2τ slot di contesa)



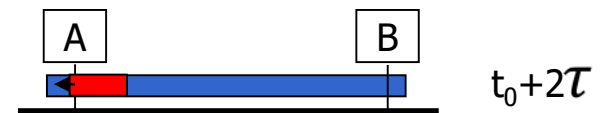
Il pacchetto deve durare più di 2τ



B rileva la collisione e sospende la trasmissione. Il disturbo generato dalla collisione si propaga verso A



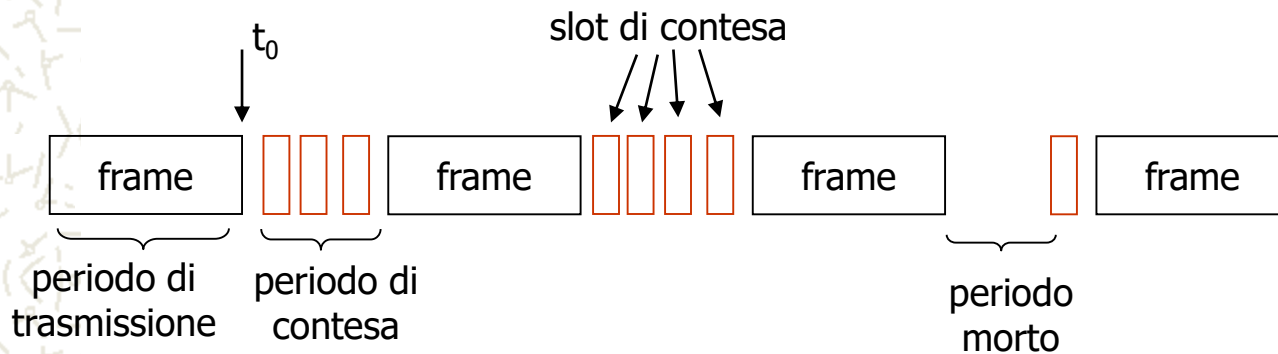
B inizia a trasmettere poco prima che l'inizio del pacchetto di A arrivi in B



A rileva la collisione e sospende la trasmissione.

CSMA/CD [Collision Detection]

- **Le stazioni bloccano la trasmissione quando rilevano una collisione**



- Alla fine di una trasmissione si può avere un periodo di contesa se 2 o più stazioni iniziano a trasmettere
- L'ampiezza di ciascun slot di contesa dipende dal ritardo di propagazione (dalla distanza delle stazioni che trasmettono in contemporanea)
- I periodi morti si hanno quando nessuna stazione ha frame da trasmettere

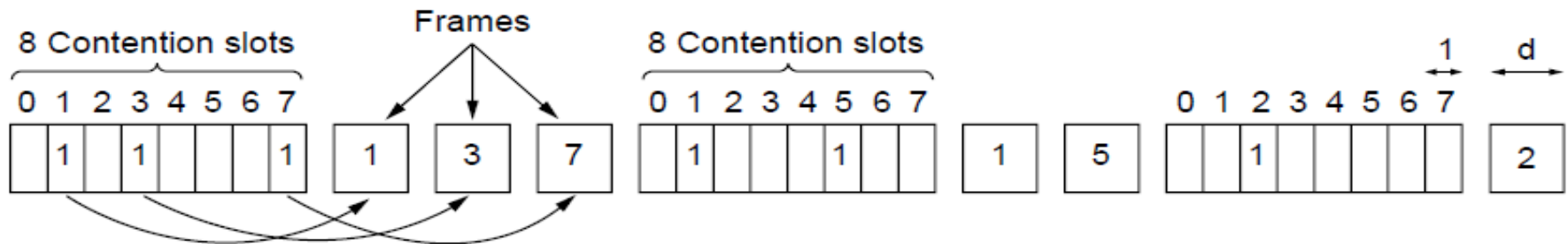
CSMA/CD: jamming

A seguito di un'avvenuta collisione si intraprendono le seguenti azioni:

- la stazione trasmittente sospende la trasmissione e trasmette una sequenza di *jamming* (interferenza trasmissiva) per comunicare a tutte le stazioni di rilevare l'avvenuta collisione
 - composta da 32 bit per 802.3 ed un numero di bit compreso tra 32 e 48 per Ethernet v.2.0;
- le stazioni in ascolto, riconoscendo il frammento di collisione costituito dalla parte di pacchetto trasmessa più la sequenza di jamming, scartano i bit ricevuti;
- la stazione trasmittente ripete il tentativo di trasmissione dopo un tempo pseudo-casuale per un numero di volte non superiore a 16.

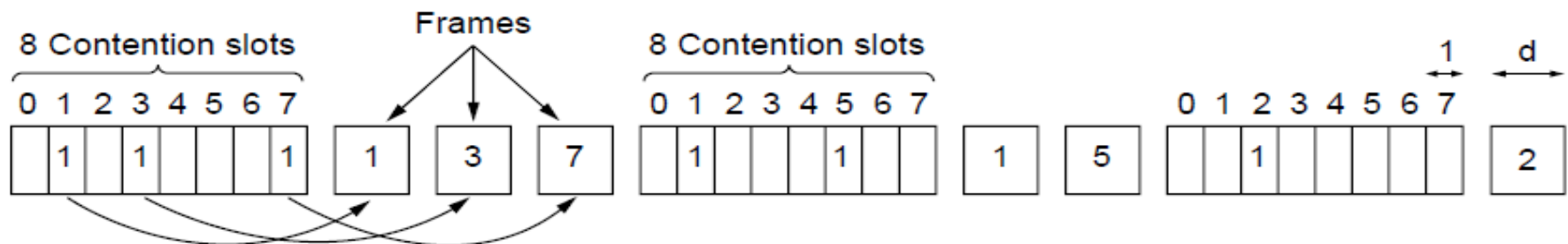
Protocolli collision free: prenotazione

- Protocollo a mappa di bit elementare:
 - sulla rete ci sono **N stazioni**, numerate da 0 a N-1
 - alla fine della trasmissione di un frame inizia un **periodo di contesa**, in cui ogni stazione, andando per **ordine di indirizzo**, trasmette un bit che vale 1 se la stazione deve trasmettere, 0 altrimenti
 - al termine del periodo di contesa (privo di collisioni in quanto ogni stazione **aspetta il suo turno**) tutti hanno appreso quali stazioni devono trasmettere, e le trasmissioni procedono **un frame alla volta** sempre andando per ordine
 - se una stazione riceve dati da trasmettere quando la fase di prenotazione è **terminata**, deve attendere il **successivo periodo di contesa** per prenotare la propria trasmissione



Protocolli collision free: prenotazione

- L'efficienza di questo protocollo è **bassa** per **grandi valori di N** e **basso carico** trasmissivo;
 - in queste condizioni una stazione **deve attendere** tutti gli N bit delle altre stazioni (delle quali la maggior parte o la totalita' non desidera trasmettere) prima di poter trasmettere
- In condizioni di **carico elevato** l'overhead dovuto agli N bit di prenotazione si **distribuisce** sui $\sim N$ frame da trasmettere, riducendo l'inefficienza complessiva del protocollo



Protocolli collision free: prenotazione

- Efficienza di canale = (carico informativo) / (carico complessivo)
- Carico informativo (d) = il numero di bit trasmessi con un frame
- Carico complessivo = ($d + N$), con N numero di stazioni (carico)
 - Con N stazioni la contention slot è lunga N bit

$$\text{Eff. Canale} = d / (d + N)$$

$$d / (d + N)$$

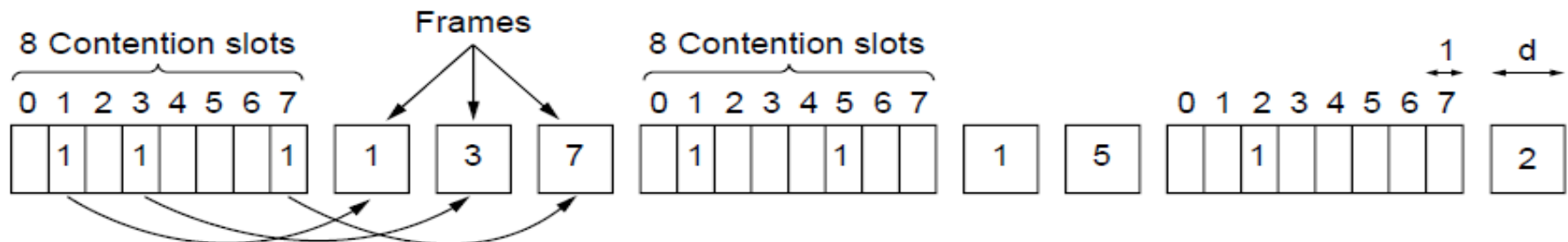
Basso carico

(Es.: 1 frame trasmesso di d bit)

$$N * d / (N * d + N)$$

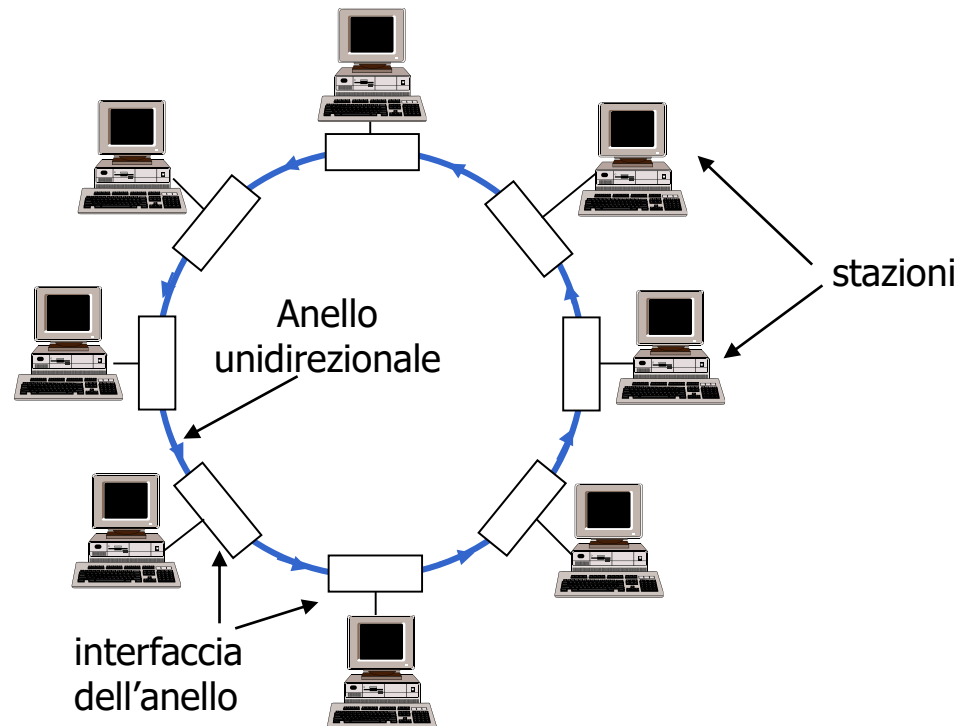
Alto carico

(Tutte le staz. trasmettono)



Protocolli Collision-free: Token Ring

Non utilizza un mezzo broadcast ma un insieme di collegamenti punto-punto associati in successione per realizzare una topologia ad anello



Protocolli Collision-free: Token Ring

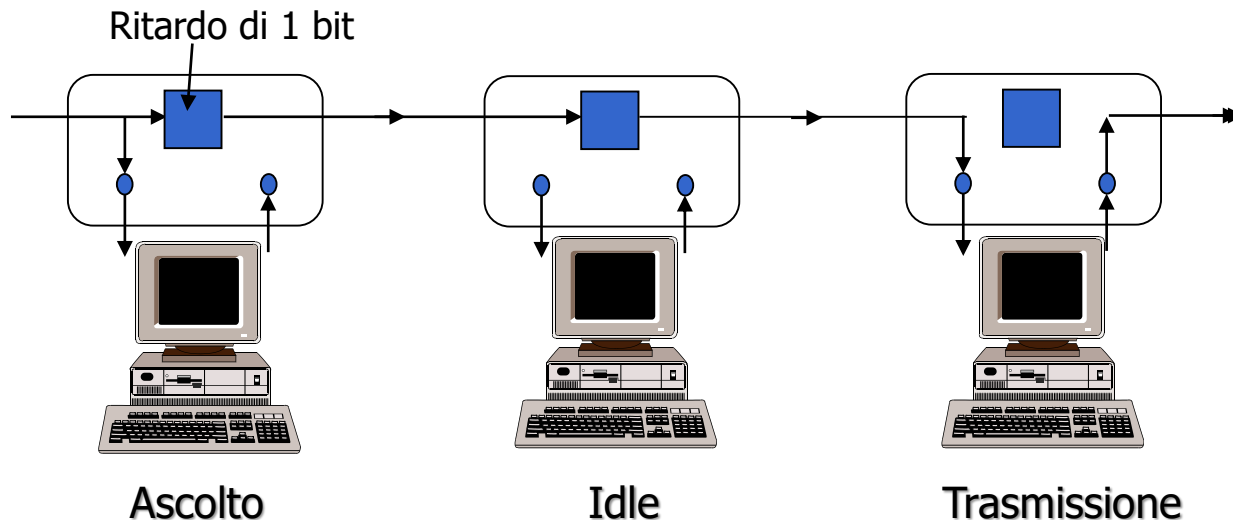
- Token ring (standard IEEE 802.5)
 - questo protocollo prevede l'utilizzo di una topologia ad anello
 - sull'anello circola un piccolo frame, detto token (gettone) che le stazioni ricevono da una parte e ritrasmettono dall'altra in continuazione
 - una stazione è autorizzata a trasmettere dati solo quando è in possesso del token
 - la stazione riceve il token, lo trattiene ed inizia a trasmettere dati
 - terminata la trasmissione, ritrasmette il token in coda ai frame di dati
 - esistono specifiche a 4 e 16 Mbps
- Esiste una versione modificata del token ring standardizzata per trasmissione su doppio anello in fibra ottica, detto FDDI (Fiber Distributed Data Interface) a 100 Mbps
- L'IEEE ha sviluppato uno standard molto simile, dedicato alle topologie a bus (token bus: IEEE 802.4)
 - in questo protocollo il problema aggiuntivo è determinato dalla necessità di configurare un ordine sequenziale delle stazioni, che viene fatto in una fase di inizializzazione del protocollo

Interfaccia Token Ring

Ogni bit che raggiunge l'interfaccia è copiato in un buffer di 1 bit

Il bit viene ritrasmesso sull'anello dopo un'eventuale controllo (**ascolto**) o modifica (**trasmissione**)

Si ha un ritardo di 1 bit per ogni interfaccia

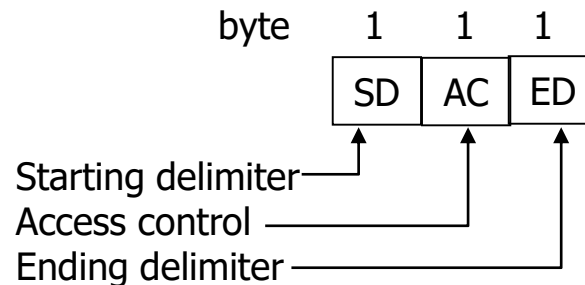


Il Token

Il **token** è una sequenza particolare di bit che circola sull'anello quando tutte le stazioni sono inattive.

Quando una stazione vuole trasmettere, si impossessa del token e lo rimuove dall'anello.

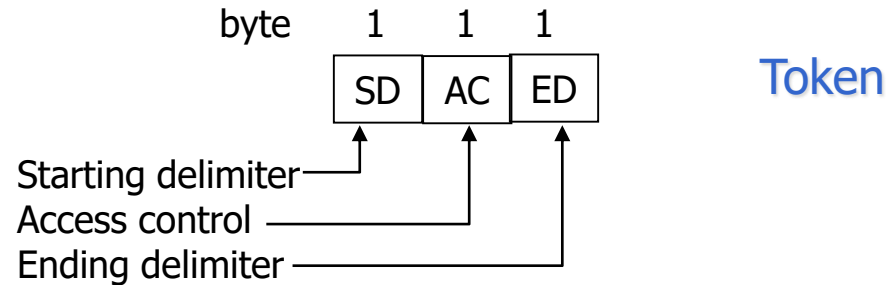
Una sola stazione può trasmettere (quella che possiede il token)



Token

Il token è acquisito semplicemente cambiando un bit nel byte Access Control

II Token



Starting Delimiter (SD)

J	K	0	J	K	0	0	0
---	---	---	---	---	---	---	---

J = non-data J
K = non-data K
0 = valore del bit

Ending Delimiter (ED)

J	K	1	J	K	1	I	E
---	---	---	---	---	---	---	---

J = non-data J
K = non-data K
1 = valore del bit
I = intermediate frame bit
E = error detected bit

Access control (AC)

P	P	P	T	M	R	R	R
---	---	---	---	---	---	---	---

PPP = priority bit
T = token bit
M = monitor bit
RRR = reservation bit

I byte di start e end hanno dei bit che corrispondono a violazioni della codifica Manchester differenziale (J,K)

Il token bit (in AC) assume valore 0 nel caso del token e valore 1 nel caso di un pacchetto

Il pacchetto

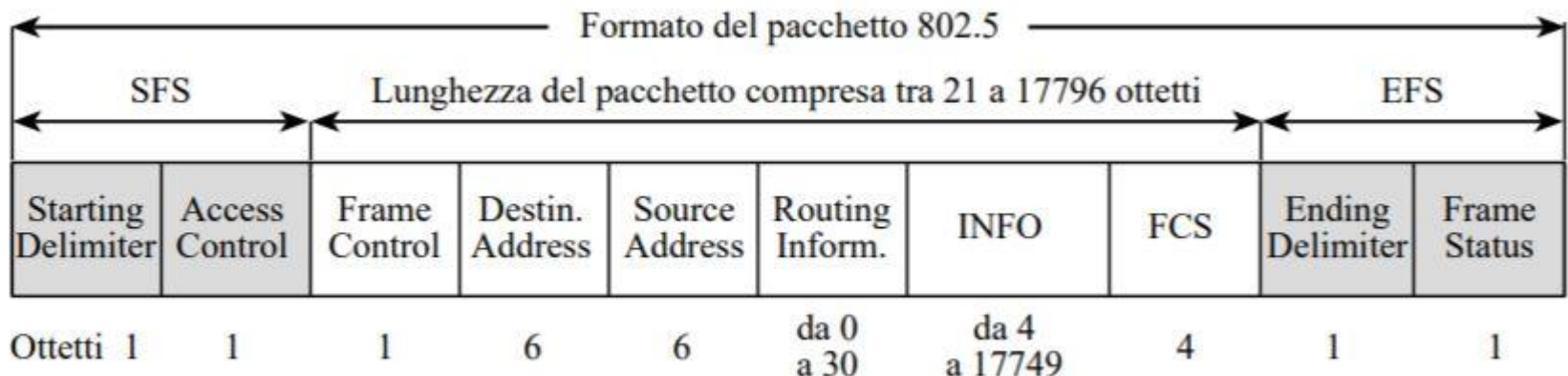
Il token bit (in AC) assume valore 0 nel caso del token e valore 1 nel caso di un pacchetto.

Il pacchetto è delimitato da due sequenze:

- SFS (Start of Frame Sequence), indica l'inizio del pacchetto
- EFS (End of Frame Sequence), indica la fine del pacchetto

Il pacchetto vero e proprio inizia dopo la SFS.

- INFO contiene i dati da inviare.
- DA e SA sono rispettivamente il Destination ed il Source Address
- FCS contiene il CRC
- FC definisce il contenuto del pacchetto
- RI contiene le informazioni d'instradamento per una WAN



Esempio di funzionamento

Es. "A" vuole inviare un'immagine al nodo "E"

- "A" sovrascrive il token, inserendo la propria "priorità" di invio all'interno di AC
- Il token gira tra tutti i nodi e torna ad A
- Se la priorità di A risulta essere superiore a quella inserita dagli altri, allora A è autorizzato a trasmettere.
- A sovrascrive il token, che diventa un pacchetto, inserendo DA, SA, ... Dati.
- Il pacchetto gira sull'anello fino ad arrivare a destinazione.
- I nodi con indirizzo diverso da DA reinviano il pacchetto al nodo successivo.
- Il nodo con indirizzo DA cattura il pacchetto, preleva i dati e sovrascrive il token con un «riscontro di avvenuta consegna»
- In DA inserisce SA ed invia il pacchetto.
- Il pacchetto girando sull'anello arriva ad A che riceve il riscontro.
- A ripristina il token

La lunghezza dell'Anello

L'anello deve avere un ritardo sufficiente per contenere un token completo circolante quando tutte le stazioni sono inattive

La velocità di propagazione tipica è di $200\text{m}/\mu\text{s}$

Se la velocità di trasmissione è di $R \text{ Mbps}$, ogni bit occupa $200/R \text{ m}$

AC (Access Control)

P: Priority bit. Priorità attuale

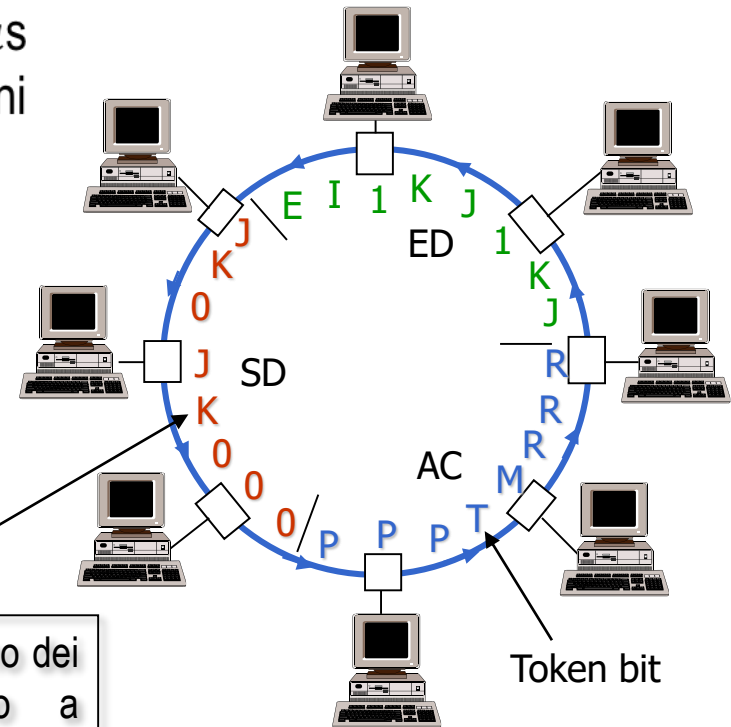
T: Segnala il tipo di Token (dati o token)

M: usato dalla stazione

Monitor per verificare se il token è orfano (senza mittente o destinatario)

R: Reservation bit (bit di prenotazione). Priorità richiesta.

I byte di start e end hanno dei bit che corrispondono a violazioni della codifica Manchester differenziale (J,K)



Protocollo token ring: efficienza

- Il protocollo token ring (come tutti quelli a turno) è **poco efficiente** in condizioni di **basso carico**
 - la stazione che deve trasmettere deve attendere di **ricevere il token** (o in generale deve attendere il suo turno) prima di poterlo fare, anche se il canale non è occupato
- In condizioni di **carico elevato**, quando tutti vogliono trasmettere, l'efficienza del protocollo sfiora l'unità
 - il solo overhead è dovuto alla necessità che ha una stazione di **identificare** il token prima di poter trasmettere
 - in questi protocolli il token è scelto in **modo opportuno** per minimizzare l'overhead
- Una importante caratteristica di questo genere di protocolli è la possibilità di valutare un tempo **massimo** di **ritardo** per le trasmissioni
 - una stazione che desidera trasmettere dovrà attendere **al più** N tempi di trasmissione (uno per stazione, nel caso tutti debbano trasmettere) prima che **tocchi nuovamente ad essa**
 - questo permette l'utilizzo del protocollo in situazioni in cui i tempi di risposta possono essere **determinanti** (ad esempio una catena di montaggio)