

Reti di Calcolatori AA 2009/2010

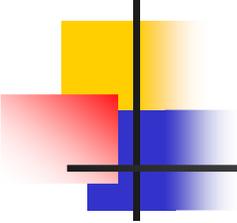


UNIVERSITÀ DEGLI STUDI DI TRENTO

<http://disi.unitn.it/locigno/index.php/teaching-duties/computer-networks>

Livello 2 OSI: Data Link e MAC

Renato Lo Cigno



Copyright

Quest'opera è protetta dalla licenza:

Creative Commons

Attribuzione-Non commerciale-Non opere derivate

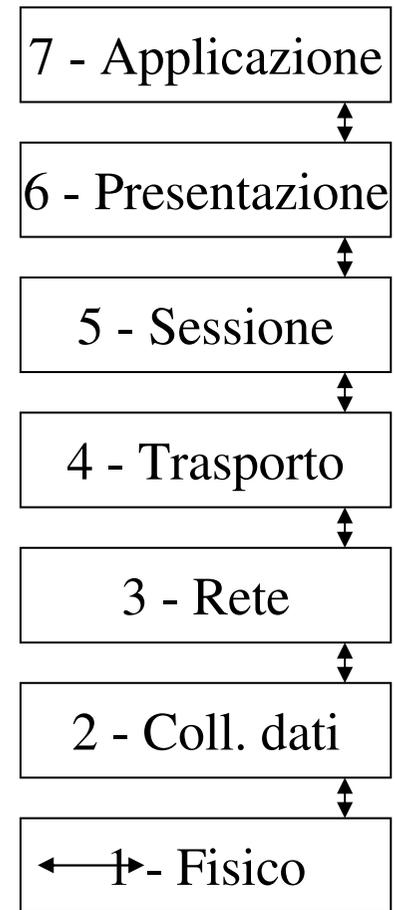
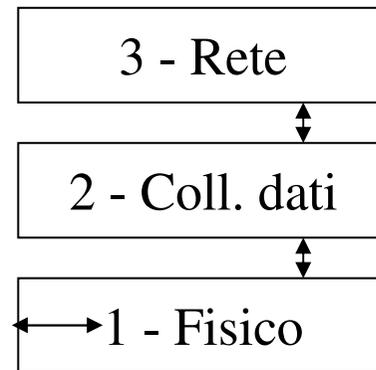
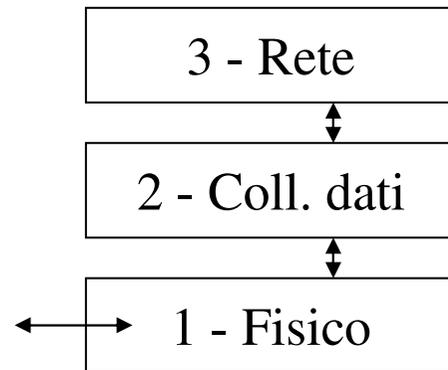
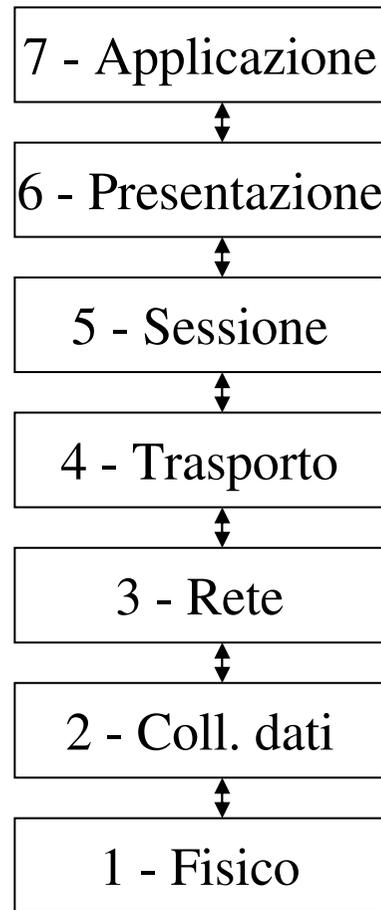
2.5 Italia License

Per i dettagli, consultare

<http://creativecommons.org/licenses/by-nc-nd/2.5/it/>

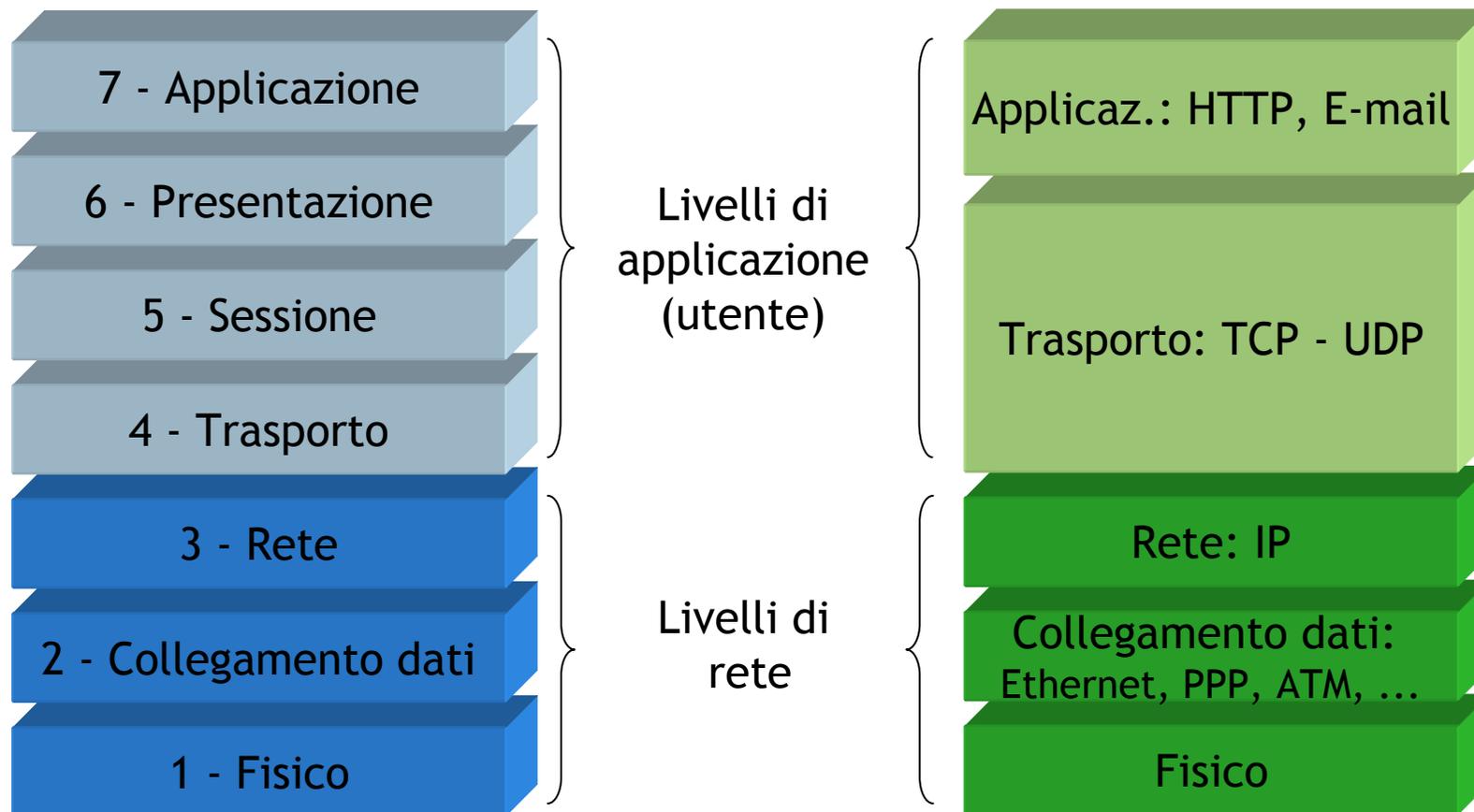


Modello a strati

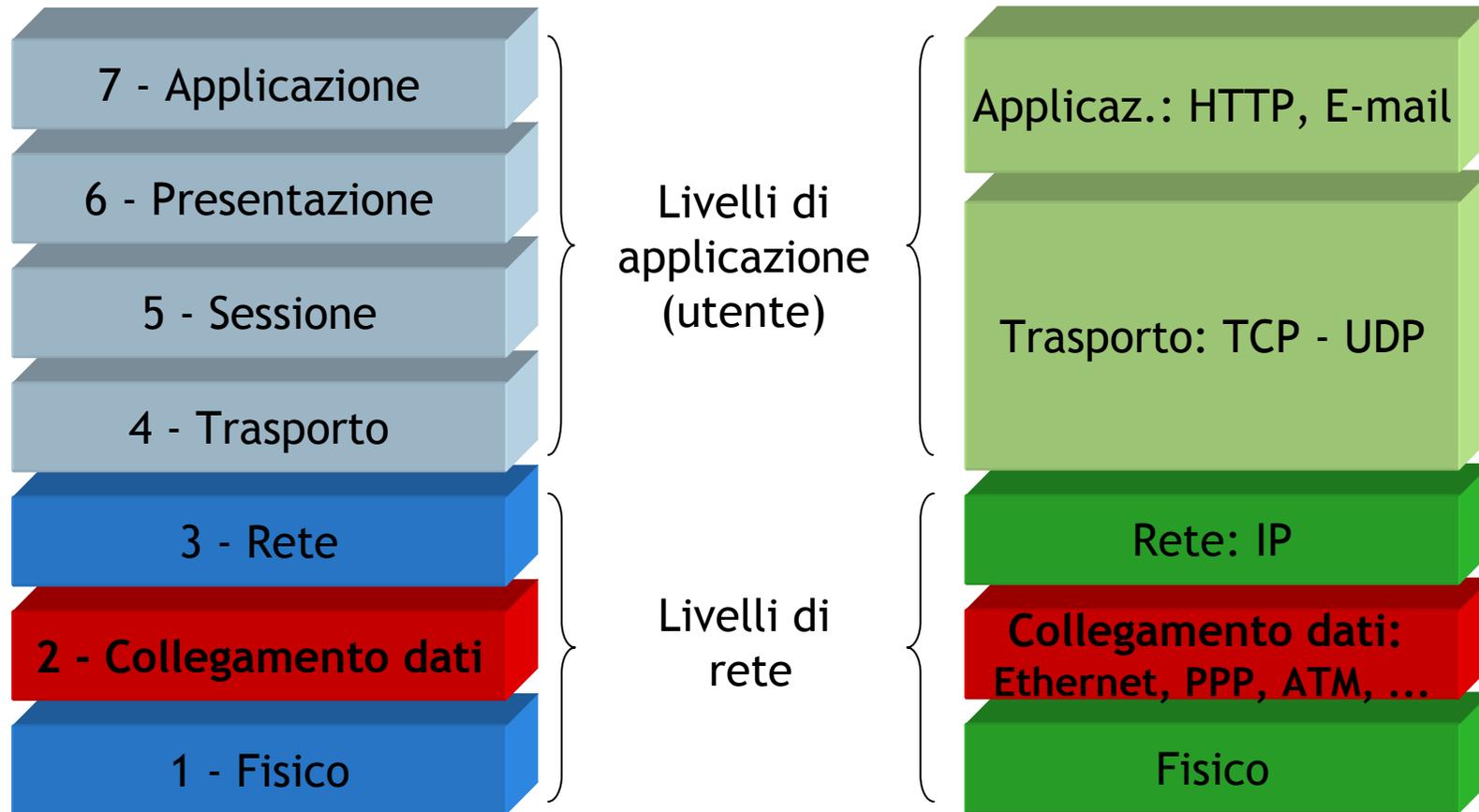


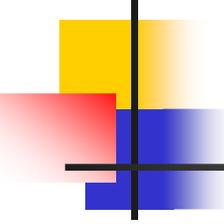
Stack OSI...

...e Stack TCP/IP



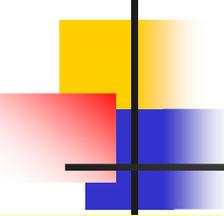
Livello Data Link





Livello Data Link

- Obiettivo principale: fornire al livello di rete di due macchine adiacenti un **canale di comunicazione** il più possibile affidabile.
 - macchine adiacenti → fisicamente connesse da un canale di comunicazione (es. un cavo coassiale, doppino telefonico)
 - canale di comunicazione → “tubo digitale”, ovvero i bit sono ricevuti nello stesso ordine in cui sono inviati
- Per compiere questo obiettivo, come tutti i livelli OSI, il livello 2 offre dei servizi al livello superiore (livello di rete) e svolge una serie di funzioni
- Problematiche: il canale fisico non è ideale
 - errori di trasmissione tra sorgente e destinazione
 - necessità di dover gestire la velocità di trasmissione dei dati
 - ritardo di propagazione non nullo

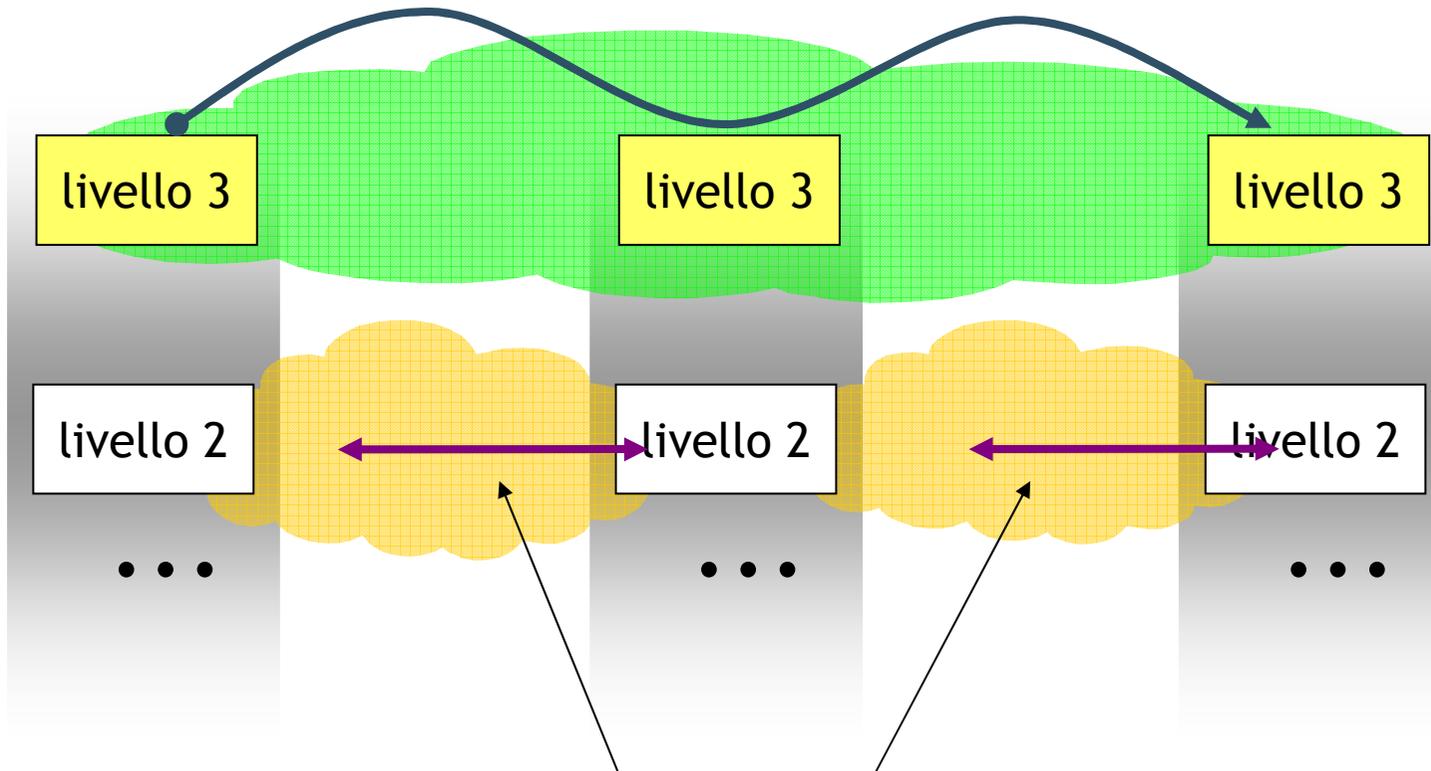


Tipologia di servizi offerti al livello superiore

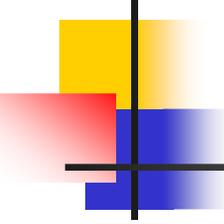
- Servizio connectionless senza acknowledge
 - non viene attivata nessuna connessione
 - invio delle trame senza attendere alcun *feedback* dalla destinazione
 - Se una trama viene persa non ci sono tentativi per recuperarla, il compito viene lasciato ai livelli superiori
 - **la maggior parte delle LAN utilizzano questa tipologia di servizio**
- Servizio connectionless con acknowledge
 - non viene attivata nessuna connessione
 - ogni trama inviata viene “riscontrata” in modo individuale
- Servizio connection-oriented con acknowledge
 - viene attivata una connessione e, al termine del trasferimento, essa viene abbattuta
 - ogni trama inviata viene “riscontrata” in modo individuale

Visibilità della rete del livello 2

Visibilità estesa a tutta la rete

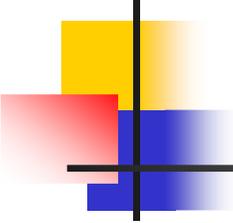


Visibilità limitata al singolo link



Funzioni di competenza del livello 2

- Le principali funzioni svolte dal livello 2 sono:
 - framing
 - delimitazione delle trame
 - rilevazione/gestione errori
 - controlla se la trama contiene errori ed eventualmente gestisce il recupero
 - controllo di flusso
 - gestisce la velocità di trasmissione

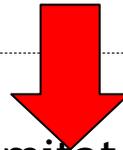
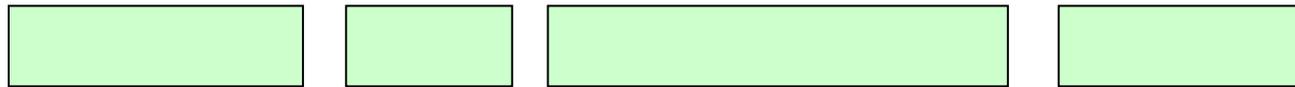


Framing

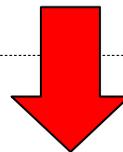
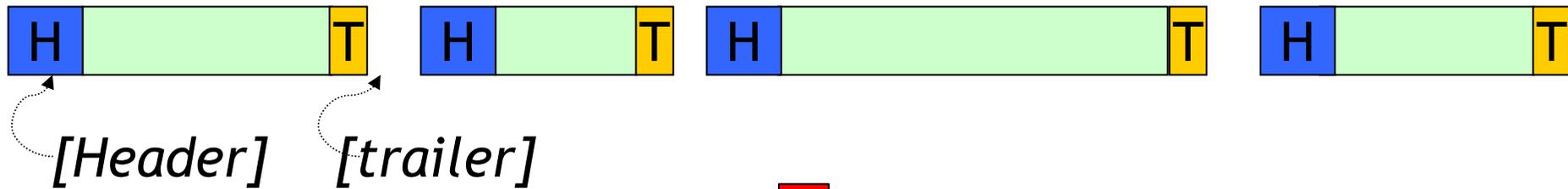
- Il livello 2 riceve dal livello superiore (rete) dei pacchetti
- Considerando che:
 - la lunghezza dei pacchetti (di livello 3) e delle corrispondenti trame (livello 2) è variabile
 - i sistemi non sono sincronizzati tra loro, ovvero non hanno un orologio comune che segna la stessa ora per tutti
 - il **livello 1 tratta solo bit**, e quindi non è in grado di distinguere se un bit appartiene ad una trama o a quella successiva
- ... nasce il problema della **delimitazione delle trame**
- La funzionalità di *framing* (frame = trama) è dunque di rendere distinguibile una trama dall'altra attraverso l'utilizzo di opportuni codici all'inizio e alla fine della trama stessa

Esempio

pacchetti dal livello 3

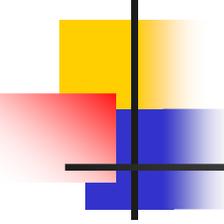


trame/frame del livello 2 con delimitatori



flusso di bit del livello 1



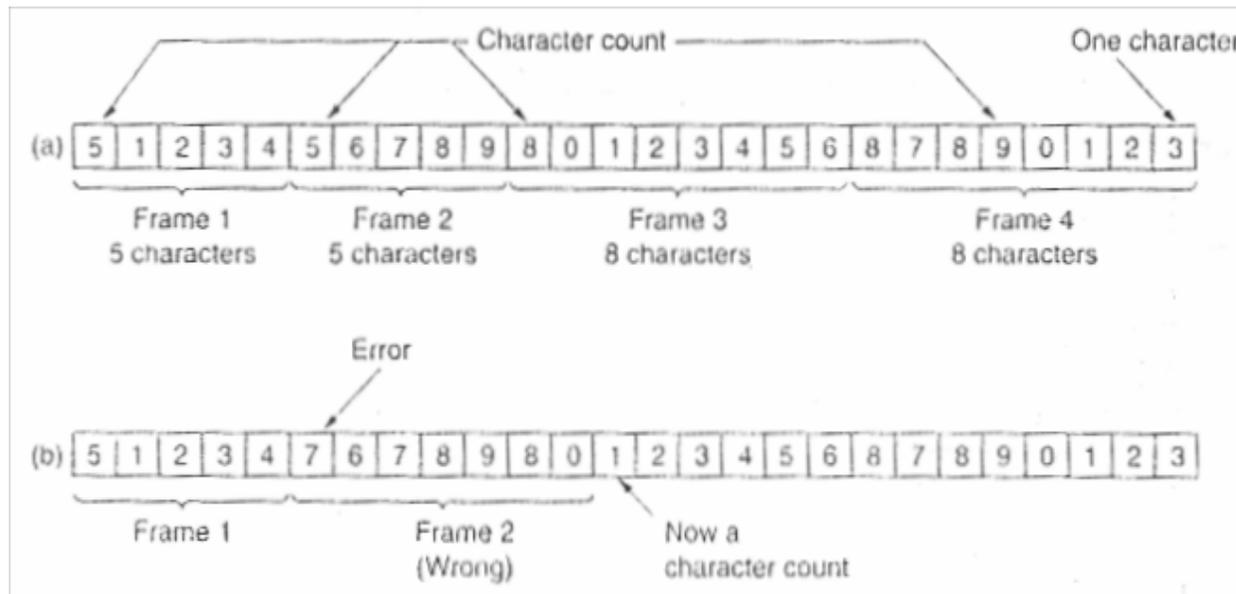


Modalità di Framing

- Esistono diverse tecniche per implementare il framing:
 - inserire intervalli temporali fra trame consecutive
 - problema: per natura intrinseca le reti di telecomunicazione non danno garanzie sul rispetto delle caratteristiche temporali delle informazioni trasmesse
 - gli intervalli inseriti potrebbero essere espansi o ridotti generando problemi di ricezione
 - marcare inizio e termine di ogni trama
 1. Character count
 2. Character stuffing
 3. Starting and ending flags (bit stuffing)
 4. Physical layer coding violations

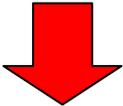
Framing: Character Count

- Un campo nell'header del frame indica il numero di 'caratteri' nel frame stesso

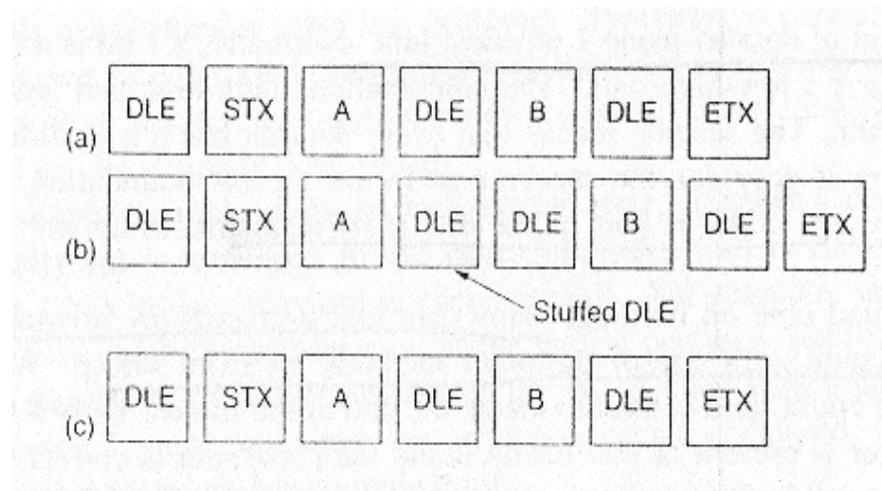


(fonte A.Tanenbaum, *Computr Networks*)

Framing: Character stuffing (1)

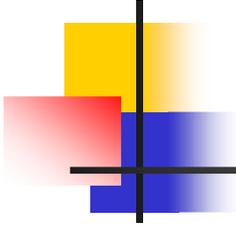
- Ogni trama inizia e termina con una sequenza di caratteri ASCII ben definita
 - DLE (Data Link Escape) + STX (Start of TeXt)
 - DLE (Data Link Escape) + ETX (End of TeXt)
- Se nella trasmissione di dati binari, una sottosequenza di bit corrisponde ai caratteri speciali...

- ...la sorgente duplica il carattere DLE
 - **character stuffing**

Framing: Character Stuffing (2)



(fonte A.Tanenbaum, *Computr Networks*)

- Svantaggio principale: soluzione legata al modulo base dei caratteri ad 8 bit e alla codifica ASCII

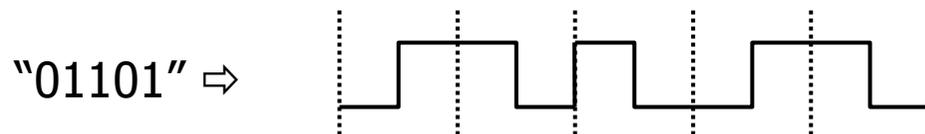


Framing: Bit Stuffing

- Ogni trama può includere un numero arbitrario di bit
- Ogni trama inizia e termina con uno speciale pattern di bit, 01111110, chiamato **byte di flag**
- In trasmissione se la sorgente incontra 5 bit "1" consecutivi, aggiunge uno "0"
 - **bit stuffing**
 - es. la sequenza "01111110" è trasmessa come "011111010"

Framing: Physical medium coding violations

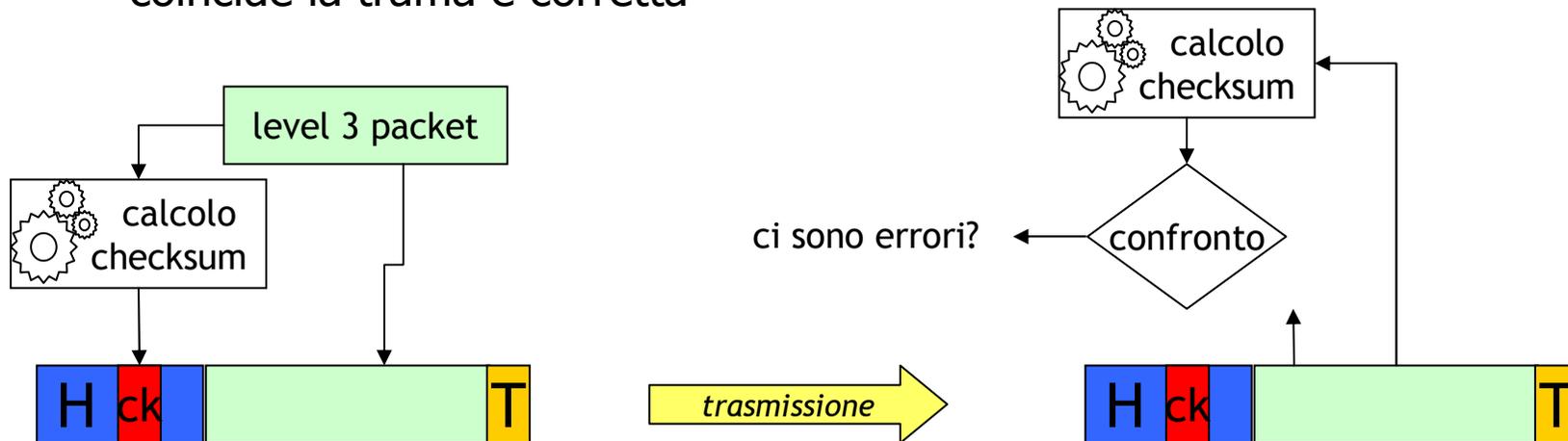
- E' una tecnica basata su sistemi che utilizzano ridondanza a livello fisico
 - es. ogni bit di informazione viene trasmesso utilizzando una combinazione di due bit a livello fisico
 - '1' \Rightarrow '10'
 - '0' \Rightarrow '01'

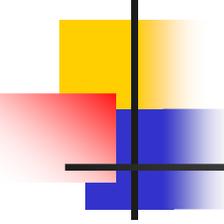


- determinate combinazioni non sono quindi usate per i dati e possono essere quindi utilizzate per il framing
 - '00' e '11'

Rilevazione dell'errore

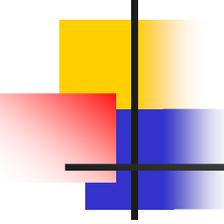
- Il livello fisico offre un canale di trasmissione **con errori**
 - errori sul singolo bit
 - replicazione di bit
 - perdita di bit
- Per la rilevazione di tali errori, nell'header di ogni trama il livello 2 inserisce un campo denominato **checksum**
 - il checksum è il risultato di un calcolo fatto utilizzando i bit della trama
 - la destinazione ripete il calcolo e confronta il risultato con il checksum: se coincide la trama è corretta





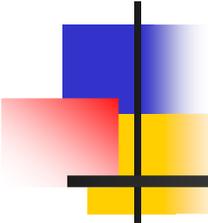
Controllo di flusso (v. Prot. a Finestra)

- Problema: la sorgente trasmette le trame ad una velocità superiore di quella che la destinazione utilizza per accettare l'informazione
 - conseguenza: congestione del nodo destinazione
- Soluzione: implementare il **controllo di flusso**
- Il controllo della velocità di trasmissione della sorgente è basato su feedback inviati alla sorgente dalla destinazione indicando
 - di bloccare la trasmissione fino a comando successivo
 - la quantità di informazione che la destinazione è ancora in grado di gestire
- I feedback possono essere
 - nei servizi con riscontro, gli ack stessi
 - nei servizi senza riscontro, dei pacchetti appositi

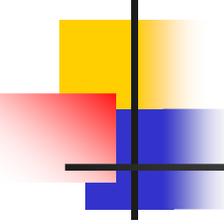


Controllo errori

- Spesso assente nelle reti locali (ma presente invece nelle reti wireless LAN)
- Presente nelle reti tradizionali di tipo geografico
- Come a livello trasporto basato su protocolli a finestra
 - normalmente stop&wait
 - sul singolo canale non ho problemi di ritardo variabile
- Ritrasmissione dell'intera trama, controllo basato su CRC



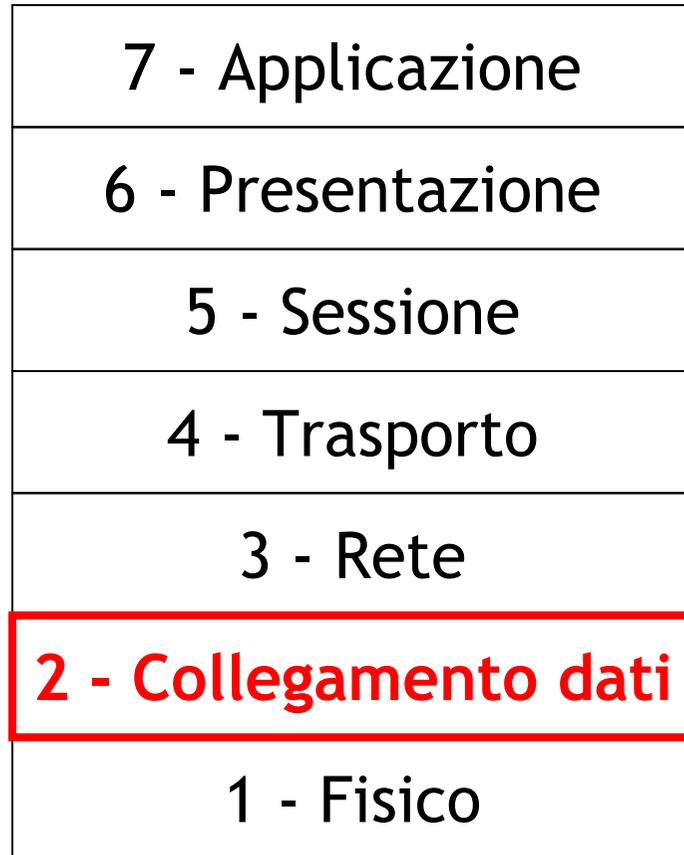
Il sotto-livello MAC



Introduzione di un nuovo sotto-livello

- Abbiamo visto che il livello 2 gestisce un insieme di problematiche svolgendo le funzioni di framing, rivelazione degli errori, controllo di flusso
- Bisogna considerare però che il livello 2 ha a che fare con il livello 1, ovvero il livello fisico (direttamente collegato al mezzo fisico)
- Il mezzo fisico può essere:
 - dedicato (reti punto-punto)
 - condiviso (reti broadcast)
- Se il mezzo fisico è condiviso, nascono una serie di problematiche relative all'accesso a tale mezzo
 - selezione dell'host che ha il diritto di trasmettere sul mezzo condiviso
 - situazione di competizione per la risorsa trasmissiva
- Viene introdotto un sotto-livello al livello 2 che gestisce queste problematiche
 - **MAC (Medium Access Control)**

Livello MAC



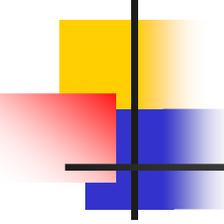
Gestisce le altre funzionalità del livello 2, in particolare il controllo di flusso

2high - Collegamento dati

2low - Medium Access Control

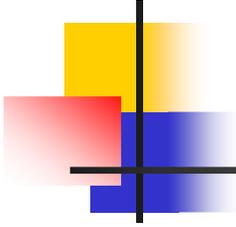
Gestisce le politiche/regole di accesso ad un mezzo condiviso

NOTA: anche se in linea di principio il livello MAC gestisce l'accesso al mezzo e il livello "high" gestisce le altre funzionalità, nella pratica il livello MAC gestisce anche il framing e il controllo di errore, mentre il livello 2 "high" si occupa del controllo di flusso. **Nello stack TCP/IP ove il livello 2 non fa controllo di flusso, il livello 2 "high" è completamente assente o, se c'è, non svolge nessuna funzione**



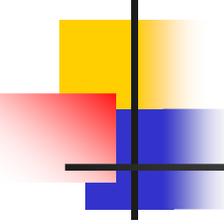
Definizione del problema

- Per mezzo **condiviso** si intende che un unico canale trasmissivo può essere usato da più sorgenti
 - esempio: stanza piena di persone che vogliono parlare tra di loro
 - se tutti parlano contemporaneamente, non potrà esserci scambio di informazione
 - l'opposto è avere un mezzo dedicato per ogni coppia di persone che vuole parlare (ad esempio un tubo o una coppia di walkie-talkie)
- E' necessario definire una serie di regole per poter utilizzare il mezzo (tecniche di allocazione del canale)
 - se due sorgenti parlano contemporaneamente vi sarà collisione e l'informazione andrà persa



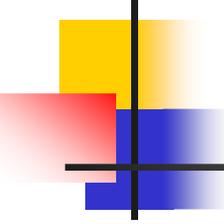
Tecniche di allocazione del canale

- Esistono due categorie in cui rientrano le tecniche di allocazione del canale trasmissivo
 - allocazione statica
 - il mezzo trasmissivo viene “partizionato” e ogni porzione viene data alle diverse sorgenti
 - il partizionamento può avvenire in base:
 - al tempo: ogni sorgente ha a disposizione il mezzo per un determinato periodo
 - alla frequenza: ogni sorgente ha a disposizione una determinata frequenza (si pensi alle stazioni radiofoniche ove il canale trasmissivo è l’aria...)
 - allocazione dinamica
 - il canale viene assegnato di volta in volta a chi ne fa richiesta e può essere utilizzato una volta che questi ha finito di usarlo e lo libera



Allocazione statica

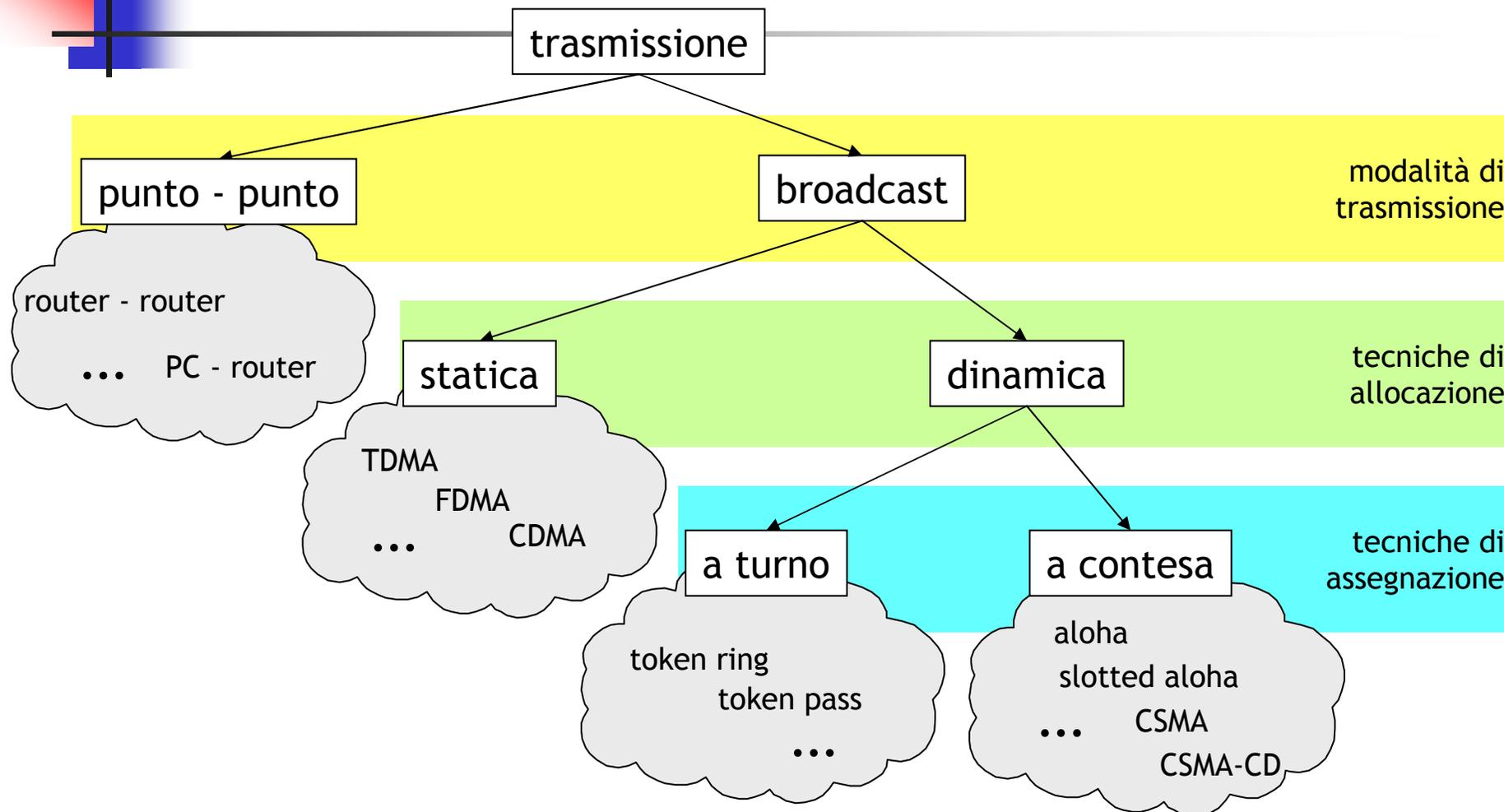
- Soluzioni “tradizionali”
 - Frequency Division Multiple Access (FDMA)
 - Time Division Multiple Access (TDMA)
 - Code Division Multiple Access (CDMA)
- Buona efficienza in situazioni di **pochi utenti con molto carico costante nel tempo**
- Meccanismi di semplice implementazione (FDM)
- Tuttavia...
 - molti utenti
 - traffico discontinuo
- ...generano una scarsa efficienza di utilizzo delle risorse trasmissive
 - le risorse dedicate agli utenti “momentaneamente silenziosi” sono perse



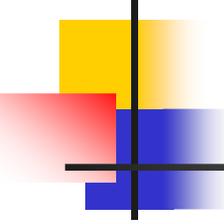
Allocazione dinamica

- Il canale trasmissivo può essere assegnato:
 - a turno
 - viene distribuito il “permesso” di trasmettere; la durata viene decisa dalla sorgente
 - a contesa
 - ciascuna sorgente prova a trasmettere indipendentemente dalle altre
- Nel primo caso si presuppone la presenza di meccanismi per l’assegnazione del permesso di trasmettere
 - overhead di gestione
- Nel secondo caso non sono previsti meccanismi particolari
 - sorgente e destinazione sono il più semplici possibile
- I protocolli che gestiscono la trasmissione a contesa sono generalmente i più utilizzati

Riassunto



In generale: se le risorse sono scarse rispetto alle esigenze delle stazioni (tante stazioni con molti dati), un accesso statico (*multiplazione*) è preferibile; viceversa, ovvero con tante risorse rispetto alle necessità delle stazioni e traffico generato discontinuo, l'allocazione dinamica (*accesso multiplo*) risulta più efficiente



Allocazione dinamica con contesa: ipotesi

- Analizziamo in dettaglio le prestazioni ottenibili da protocolli (protocollo: insieme di regole...) progettati per gestire l'allocazione dinamica del canale con contesa della risorsa. Seguono una serie di ipotesi per semplificare il problema

Single channel assumption

- unico canale per tutte le comunicazioni

Station model

- N stazioni indipendenti ognuna delle quali è sorgente di trame di livello 2
- le trame sono generate secondo la distribuzione di Poisson con media **S**
- la lunghezza delle trame è fissa, ovvero il tempo di trasmissione è costante e pari a **T** (tempo di trama)
- una volta generata una trama, la stazione è bloccata fino al momento di corretta trasmissione

Collision assumption

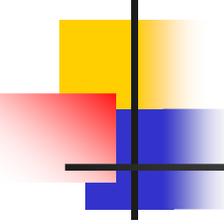
- due trame contemporaneamente presenti sul canale generano collisione
- non sono presenti altre forme di errore

Tempo...

- continuo: la trasmissione della trama può iniziare in qualunque istante
- *slotted*: la trasmissione della trama può iniziare solo in istanti discreti

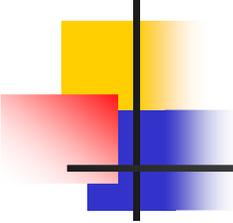
Ascolto del canale...

- *carrier sense*: le stazioni sono in grado di verificare se il canale è in uso prima di iniziare la trasmissione di una trama (questo equivale a dire che il tempo di propagazione t è $= < T$)



Protocolli di accesso multiplo

- In letteratura sono disponibili molti algoritmi di accesso multiplo al mezzo condiviso con contesa
- Principali algoritmi (utilizzati dai protocolli):
 - ALOHA
 - Pure ALOHA
 - Slotted ALOHA
 - Carrier Sense Multiple Access Protocols
 - CSMA
 - CSMA-CD (Collision Detection: con rilevazione della collisione)
 - CSMA-CA (Collision Avoidance: con tecniche per ridurre la probabilità di collisione)

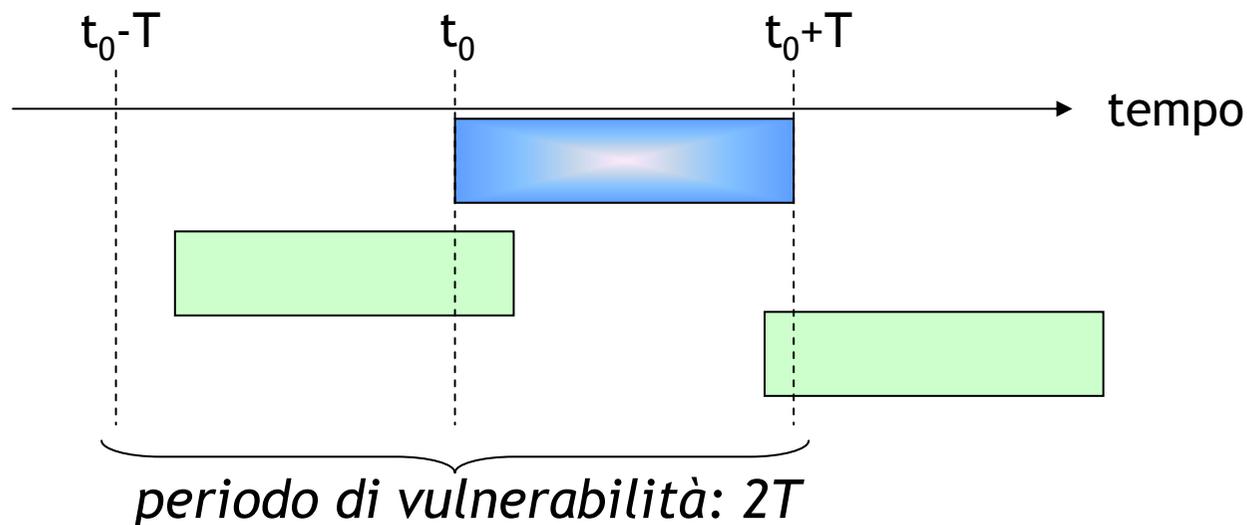


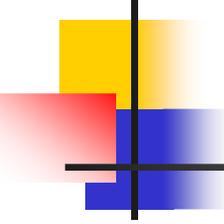
Pure ALOHA

- Definito nel 1970 da N. Abramson all'università delle Hawaii
- Algoritmo:
 - una sorgente può trasmettere una trama ogniqualvolta vi sono dati da inviare (*continuous time*)
 - se il canale è cablato la sorgente ascolta il canale per rilevare **collisioni**, se wireless il ricevitore invia esplicitamente un ACK di ricezione
 - **collisione** \Rightarrow la sorgente aspetta un tempo **casuale** e ritrasmette la trama
 - un tempo deterministico porterebbe ad una situazione di collisione all'infinito

Periodo di vulnerabilità

- Si definisce "periodo di vulnerabilità" l'intervallo di tempo in cui può avvenire una collisione che invalida una trasmissione
- Detto T il tempo di trama e t_0 l'inizio della trasmissione da parte di una sorgente, il periodo di vulnerabilità è pari al doppio del tempo di trama
 - nel momento in cui inizia a trasmettere (t_0), nessuna altra sorgente deve aver iniziato la trasmissione dopo l'istante di tempo $t_0 - T$ e nessuna altra sorgente deve iniziare la trasmissione fino a $t_0 + T$





Prestazioni

- Ipotesi
 - trame di lunghezza fissa
 - tempo di trama: tempo necessario per trasmettere una trama
 - popolazione ∞ che accede ad un mezzo condiviso
- Traffico generato (numero di trame per tempo di trama) segue la distribuzione di Poisson con media G
 - G ingloba anche il numero di ri-trasmissioni dovuto a collisioni
- Il throughput reale è dato da
 - numero medio di trasmissioni * probabilità che non ci siano trasmissioni per tutto il periodo di vulnerabilità (2 tempi di trama consecutivi)
→ $S = G \cdot P[0 \text{ trasmissioni per } 2T]$, ovvero

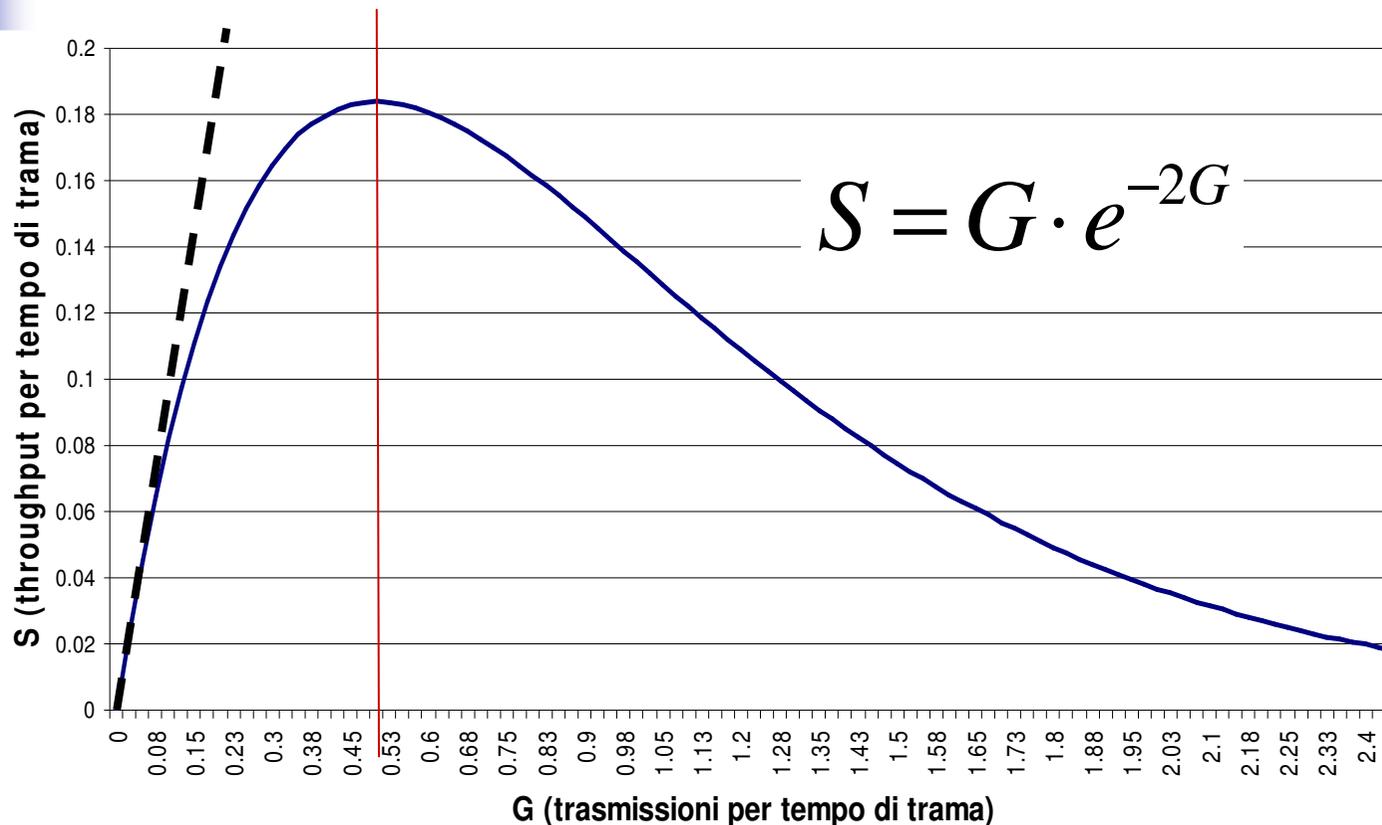
$$S = G \cdot e^{-2G}$$

G = numero medio di trame trasmesse nel tempo di trama

S = numero medio di trame trasmesse con successo (throughput)

Prestazioni

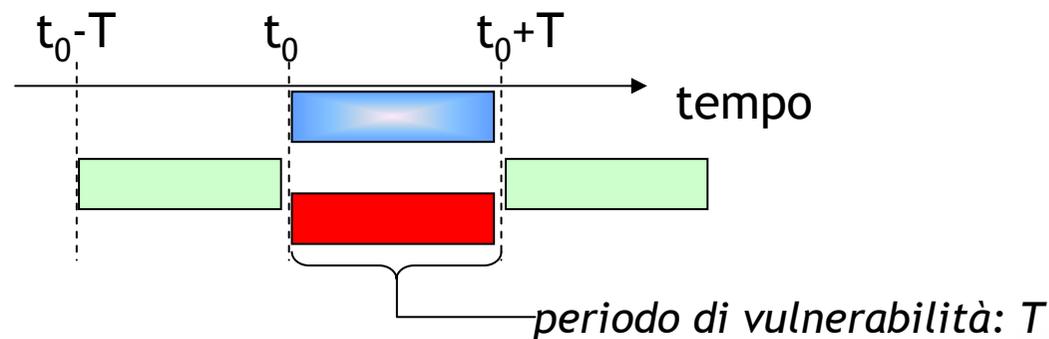
Throughput



- ALOHA permette al massimo di sfruttare il 19% del tempo, il massimo si ha quando il traffico offerto è 0.5 volte la capacità del canale. **Protocollo instabile!!**

Slotted ALOHA

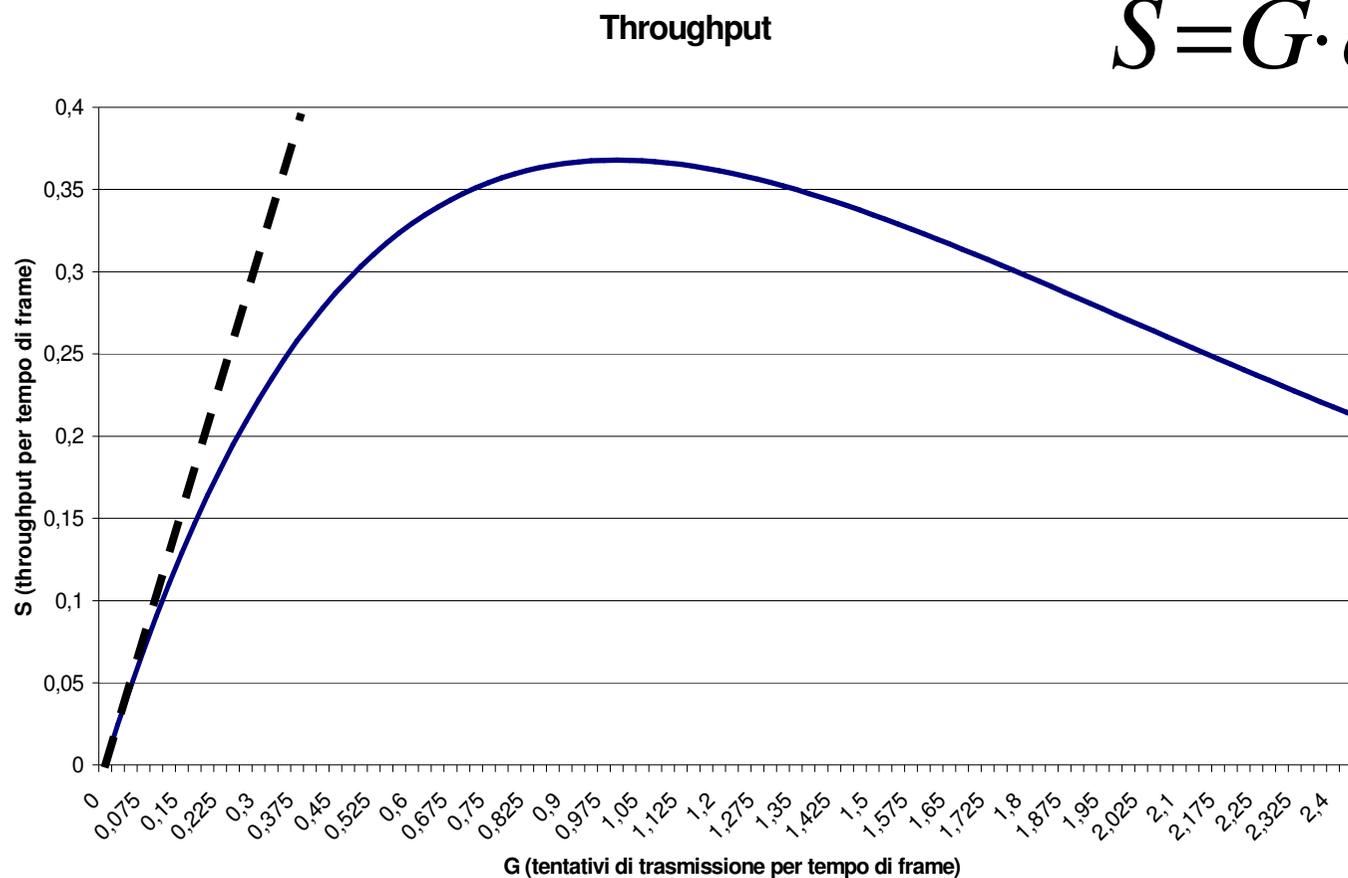
- Proposto nel 1972 da Roberts per migliorare la capacità di Pure ALOHA
- Basato su ipotesi di *slotted time* (tempo suddiviso ad intervalli discreti)
- Algoritmo:
 - Pure ALOHA
 - la trasmissione di una trama può iniziare solo ad intervalli discreti
 - necessaria sincronizzazione tra stazioni
- Periodo di vulnerabilità: T (tempo di trama)



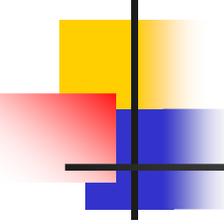
Prestazioni

- Il periodo di vulnerabilità è dimezzato, quindi il throughput reale è dato da

$$S = G \cdot e^{-G}$$

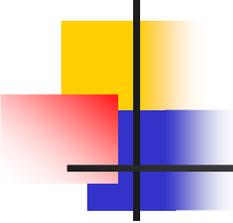


- Slotted ALOHA permette al massimo di sfruttare il 37% degli slot liberi a carico 1
- Il protocollo è instabile!!
- Bisogna distribuire un sincronismo



Carrier Sense Multiple Access (CSMA)

- Ambito LAN: le stazioni possono monitorare lo stato del canale di trasmissione (ritardi bassi)
- Le stazioni sono in grado di "ascoltare" il canale prima di iniziare a trasmettere per verificare se c'è una trasmissione in corso
- Algoritmo
 - se il canale è libero, si trasmette
 - se è occupato, sono possibili diverse varianti
 - non-persistent
 - rimanda la trasmissione ad un nuovo istante, scelto in modo casuale
 - persistent
 - nel momento in cui si libera il canale, la stazione inizia a trasmettere
 - se c'è collisione, come in ALOHA, si attende un tempo casuale e poi si cerca di ritrasmettere

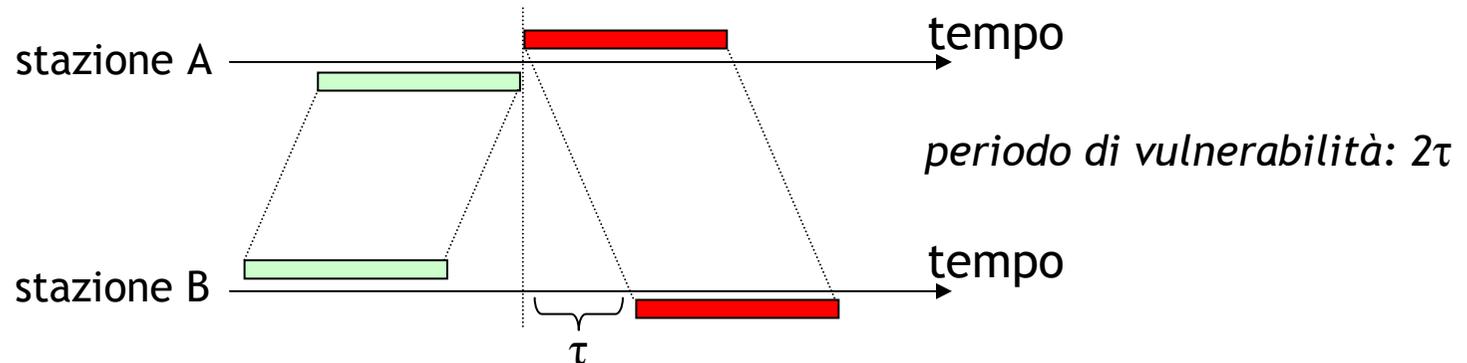


CSMA: modalità p-persistent

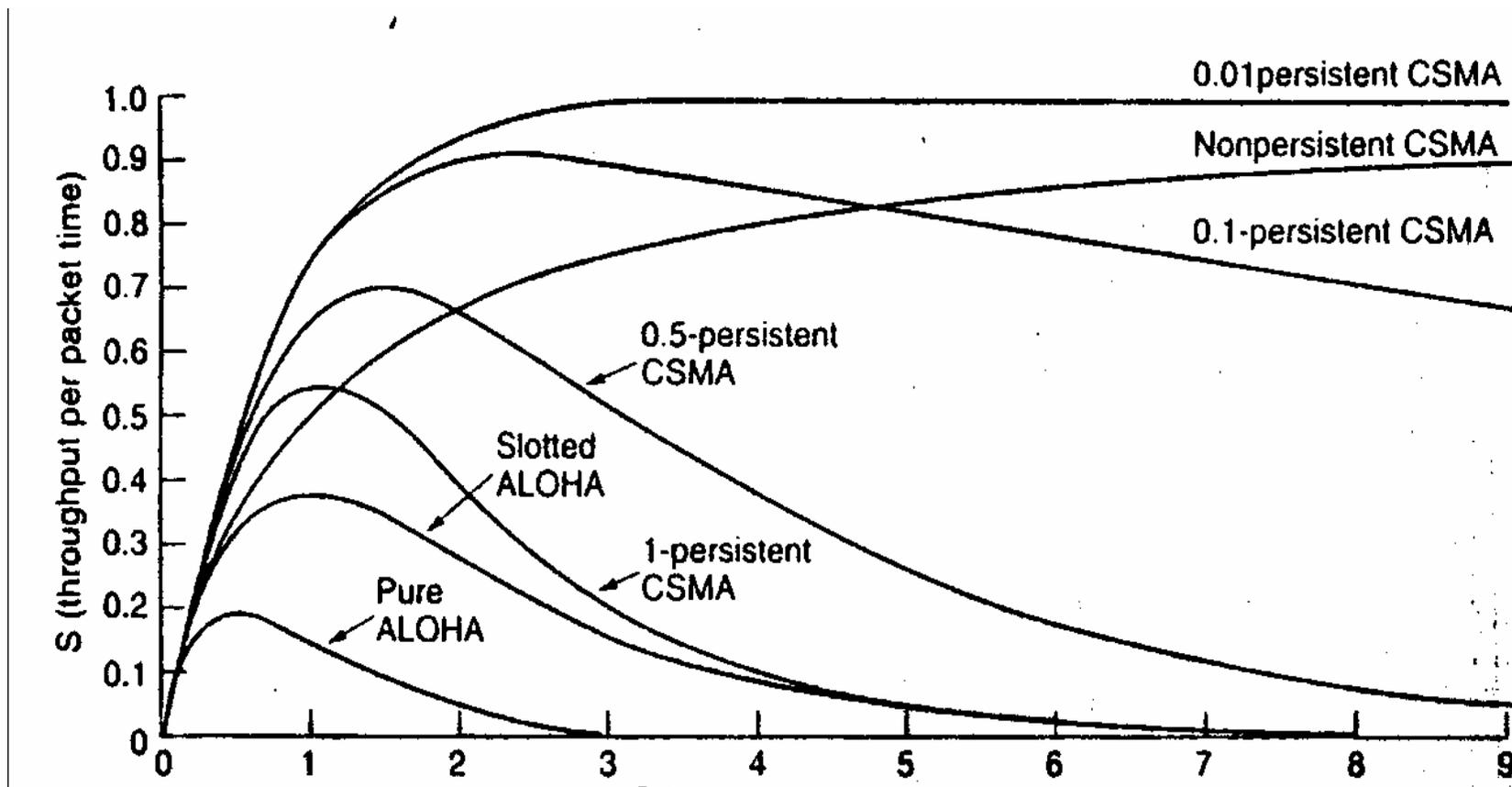
- Il tempo viene suddiviso in intervalli
 - la lunghezza degli intervalli è uguale al periodo di vulnerabilità
 - *round trip propagation delay* 2τ
- Algoritmo
 1. ascolta il canale
 - se il canale è libero
 - si trasmette con probabilità p ;
 - se si è deciso di trasmettere, si passa al punto 2
 - se non si è deciso di trasmettere, si attende un intervallo di tempo e si torna al punto 1
 - se è occupato, si attende un intervallo di tempo e si torna al punto 1
 2. se c'è collisione
 - si attende un tempo casuale e poi si torna al punto 1

Periodo di vulnerabilità

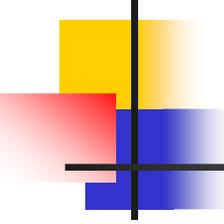
- In questo caso il periodo di vulnerabilità è legato al ritardo di propagazione del segnale (τ)
 - se una stazione ha iniziato a trasmettere, ma il suo segnale non è ancora arrivato a tutte le stazioni, qualcun altro potrebbe iniziare la trasmissione
 - periodo di vulnerabilità $\rightarrow 2\tau$
- A seconda del ritardo di propagazione, se questi risulta paragonabile al tempo di trama o meno, si hanno prestazioni differenti
- In generale, il CSMA viene usato in reti in cui il ritardo di propagazione τ è \ll di T (tempo di trama)



Confronto efficienza algoritmi



(fonte: A. Tanenbaum, Computer Networks)



CSMA con Collision Detection (CSMA-CD)

- Miglioramento
 - se la stazione che sta trasmettendo rileva la collisione, interrompe immediatamente
- In questo modo, una volta rilevata collisione, non si spreca tempo a trasmettere trame già corrotte
- Inoltre, per far sentire a tutte le stazioni che vi è stata collisione, si trasmette una particolare sequenza, detta di jamming