



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Protocolli: Prestazioni

Franco CALLEGATI

Dipartimento di Informatica: Scienza e Ingegneria



# Affidabilità

- Controllo dell'errore
  - Rivelazione e correzioni
  - CRC e Internet checksum
- Eventuale recupero dell'errore
  - ARQ e ritrasmissione
- Controllo di flusso e sequenza
  - Ack e ARQ
- Questi temi sono stati trattati lo scorso anno



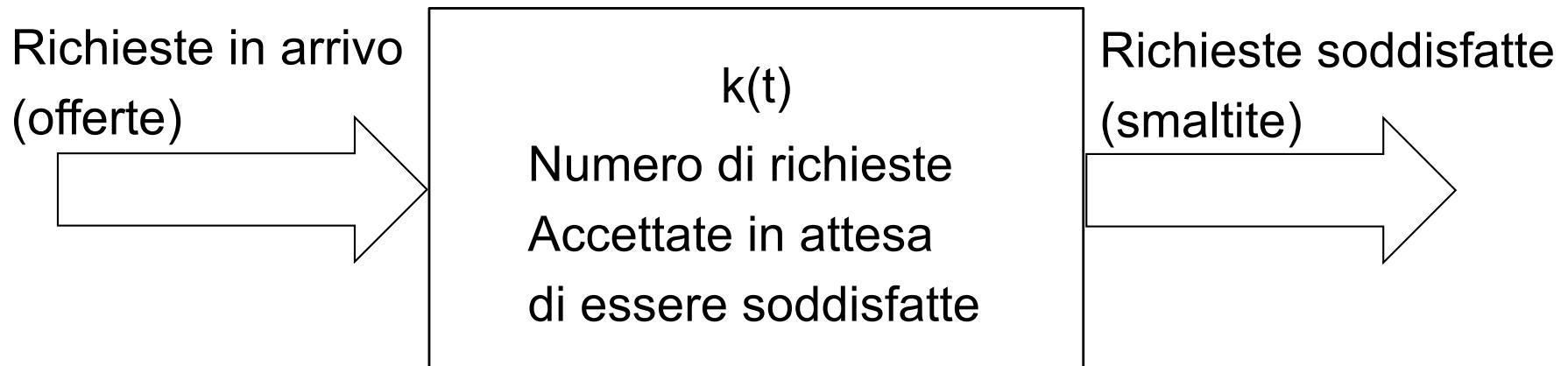
# Funzionalità e prestazioni

- I protocolli sono progettati per garantire
  - Funzionalità
    - La trasmissione dati deve poter avvenire risolvendo i problemi che si riscontrano nell'accesso e nell'uso del canale
  - Prestazioni
    - La trasmissione deve avvenire con successo utilizzando per quanto possibile la capacità messa a disposizione dallo strato fisico



# Le prestazioni in generale

- Un *sistema* deve *smaltire* del *lavoro* che gli viene *offerto* dall'esterno
- Esempio nel caso specifico delle reti di tlc
  - Livello N+1 invia PDU al livello N tramite la relativa interfaccia (e un opportuno SAP)
  - Livello N impiega un certo tempo per soddisfare la richiesta



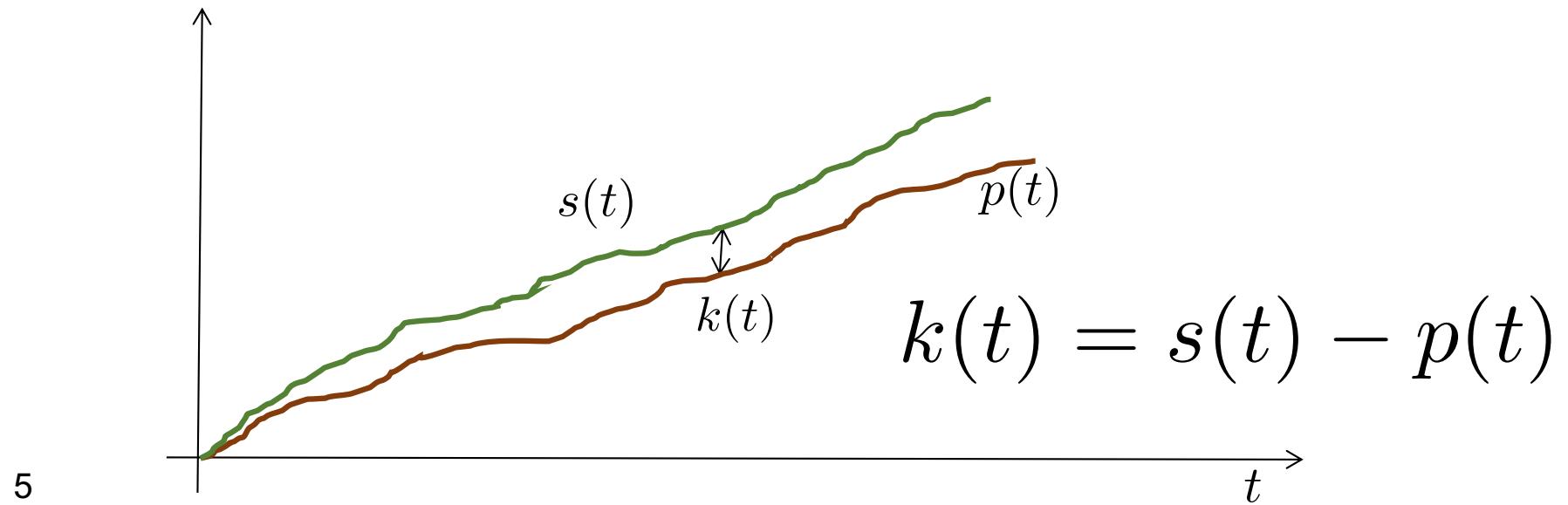


# Arrivi e partenze

$a(t)$  • Numero di richieste di servizio giunte al tempo  $t$

$s(t)$  • Numero di richieste accettate al tempo  $t$

$p(t)$  • Numero di partenze dal sistema al tempo  $t$





# Richieste offerte e smaltite

- Frequenza media delle richieste offerte

$$\lambda = \lim_{t \rightarrow \infty} \frac{a(t)}{t}$$

- Frequenza media delle richieste smaltite

$$\lambda_s = \lim_{t \rightarrow \infty} \frac{p(t)}{t}$$

- Se il sistema in oggetto non produce lavoro ma lo riceve solamente dall'esterno

$$\lambda_s \leq \lambda$$



# Richieste perdue

$$\lambda_s = \lambda \quad \text{Implica} \quad s(t) = a(t)$$

- Tutte le richieste vengono accettate dal sistema e prima o poi soddisfatte

$$\lambda_s < \lambda \quad \text{implica} \quad r(t) = a(t) - s(t)$$

- Dove  $r(t)$  rappresenta le richieste che non vengono accettate e sono *rifiutate o perdute* dal sistema



# Analogamente

- Posso definire

$$\lambda_p = \lim_{t \rightarrow \infty} \frac{r(t)}{t}$$

- Da cui consegue

$$\lambda = \lambda_s + \lambda_p$$



# Utente e servizio

- In una rete a pacchetto considerare il semplice «bit rate» del canale non è del tutto corretto
  - I bit sono raggruppati in pacchetti
  - L'unità di servizio è il pacchetto non il bit
    - In altre parole trasmettere una porzione di bit appartenenti ad un pacchetto non ha senso di per se
- **Il riferimento è il tempo di servizio dell'intero pacchetto che solo se completato produce un risultato «utile» per l'utente**
- $\vartheta$  Tempo richiesto dal servizio di un generico cliente (pacchetto dati, PDU)
  - Servizio aleatorio
    - Si fa riferimento in prima battuta al tempo medio
  - Servizio deterministico
    - Tempo di servizio costante ed uguale al suo valore medio



# La lunghezza del pacchetto

- $L$  lunghezza del pacchetto in bit
- $C$  capacità del canale in bit per secondo
- Ovviamente

$$\bar{\vartheta} = \frac{L}{C}$$



# Frequenza di servizio

- L'inverso del tempo medio di servizio viene detto frequenza media di servizio

$$\mu = \frac{1}{\bar{\vartheta}}$$

- La frequenza media di servizio è legata alla presenza di utenti del sistema
  - Se non vi sono richieste di servizio ovviamente la frequenza di servizio è nulla
  - Se vi sono richieste di servizio il parametro da indicazione di quanto velocemente esse vengono soddisfatte



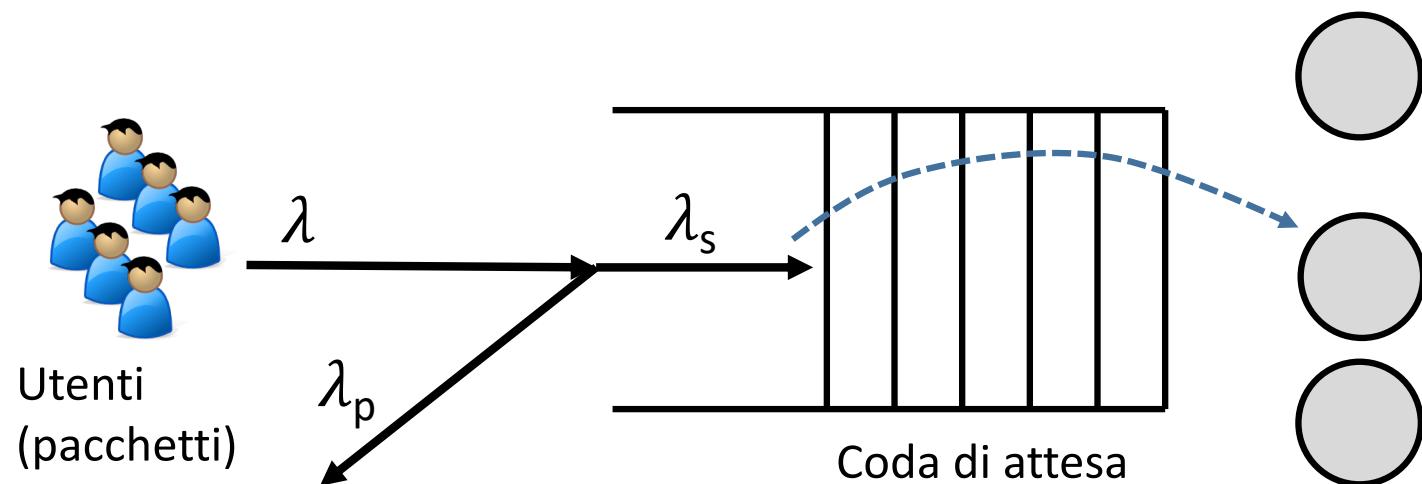
# Il ruolo di $\mu$

- Di fatto  $\mu$  può essere interpretato come una sorta di capacità massima del servitore
- Se  $\bar{\vartheta} = 0.5$  s ne consegue che il servitore al più smaltirà  $\mu = 2$  pacchetti/s
- Possiamo dire che il singolo servitore al massimo può smaltire  $m$  pacchetti

$$\lambda_s^{\max} = \mu$$

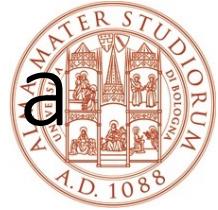
# Di quale sistema parliamo

- Per le reti hanno particolare importanza i sistemi a coda



Uno o più servitori  
(sistemi di trasmissione)

Nella reti a pacchetto 1 solo servitore  
è il caso di riferimento



# Cosa accade all'utente in un sistema a coda?

- L'utente permane nel sistema per un tempo che tiene conto dell'attesa in coda e del tempo di esecuzione del servizio
- $\bar{\delta}$  tempo medio totale speso dal singolo utente nel sistema a coda che è composto da
  - $\bar{\vartheta}$  tempo effettivo di servizio (l'utente fa le cose per le quali è entrato nel sistema)
  - $T_A$  tempo speso in coda, ossia tempo di attesa prima di essere effettivamente servito

$$\bar{\delta} = \bar{\vartheta} + T_A$$



# Il traffico

- Le prestazioni del sistema che fornisce il servizio dipendono:
  - Dalla numerosità degli arrivi, tipicamente frequenza media di arrivo (utenti per secondo)
  - Dalla durata del servizio, tipicamente la frequenza media di servizio (utenti al secondo) o il tempo medio di servizio (secondi)
- Si definisce **traffico** il numero medio di utenti presenti nel sistema
- Si dimostra che il prodotto fra frequenza di arrivo e tempo medio di permanenza nel sistema (teorema di Little) da il traffico

$$A = \lambda \bar{\delta}$$



# Alcune definizioni conseguenti

- Per analogia si definiscono

$$A_0 = \lambda \bar{\vartheta}$$

- Traffico offerto (occupazione media di un sistema ideale che serve subito tutti gli utenti senza attesa)

$$A_s = \lambda_s \bar{\vartheta}$$

- Traffico smaltito (occupazione media dei servitori del sistema)
- Ci ricorda che se  $ls$  utenti mediamente entrano per unità di tempo entrano allora  $ls$  devono uscire

$$A_p = \lambda_p \bar{\vartheta}$$

- Traffico perduto (occupazione media di un sistema che serve gli utenti che invece sono stati rifiutati)



# Il traffico smaltito

- $A_s$  da una valutazione della capacità di servizio del sistema considerato
- Se i servitori sono impegnati al 100% il sistema esprime tutta la sua capacità di servizio
- Se i servitori sono  $m$  allora

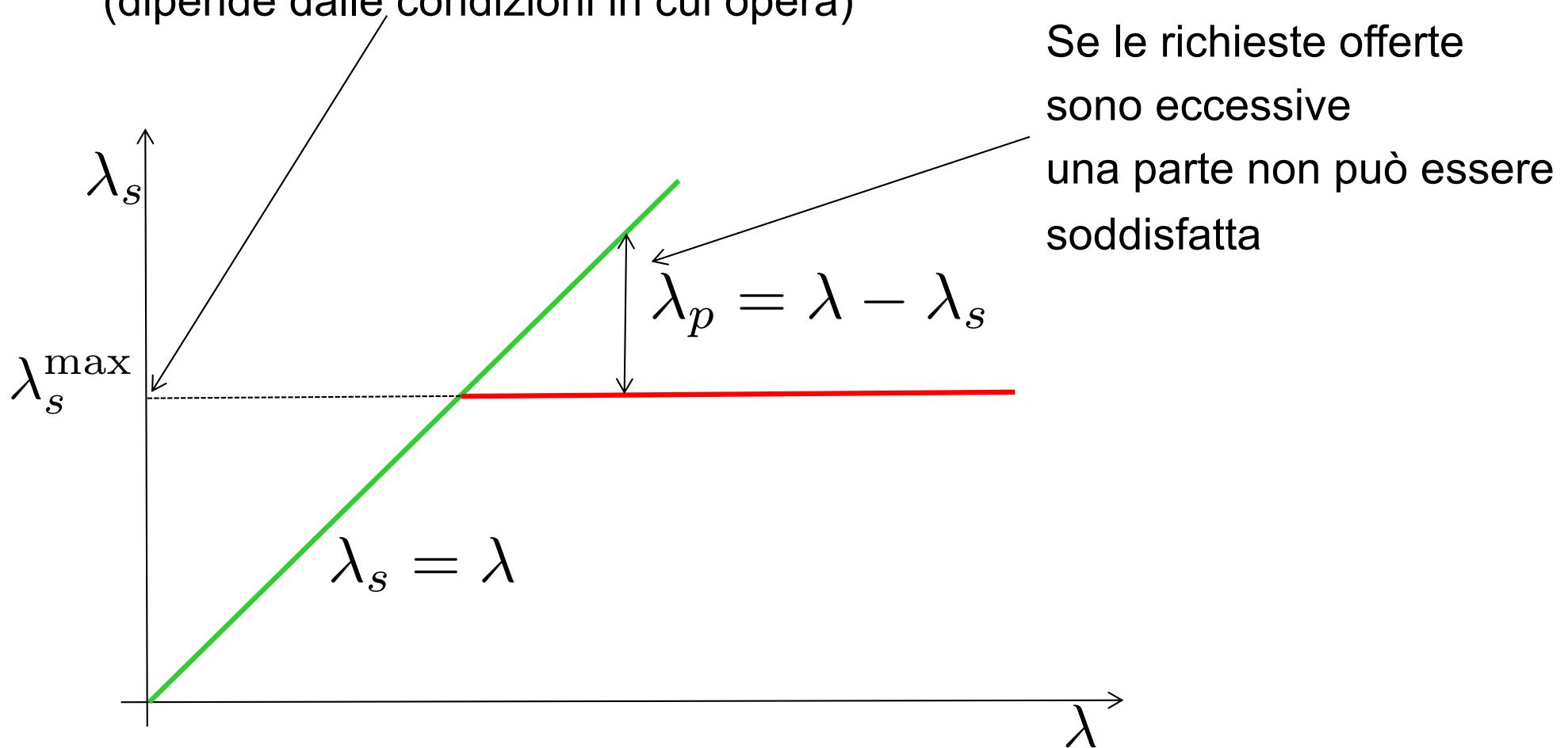
$$0 \leq A_s \leq m$$

- $A_s$  è spesso indicato con il nome di *throughput*



# In un sistema ideale

Il sistema ha una capacità massima finita  
di smaltire richieste  
(dipende dalle condizioni in cui opera)

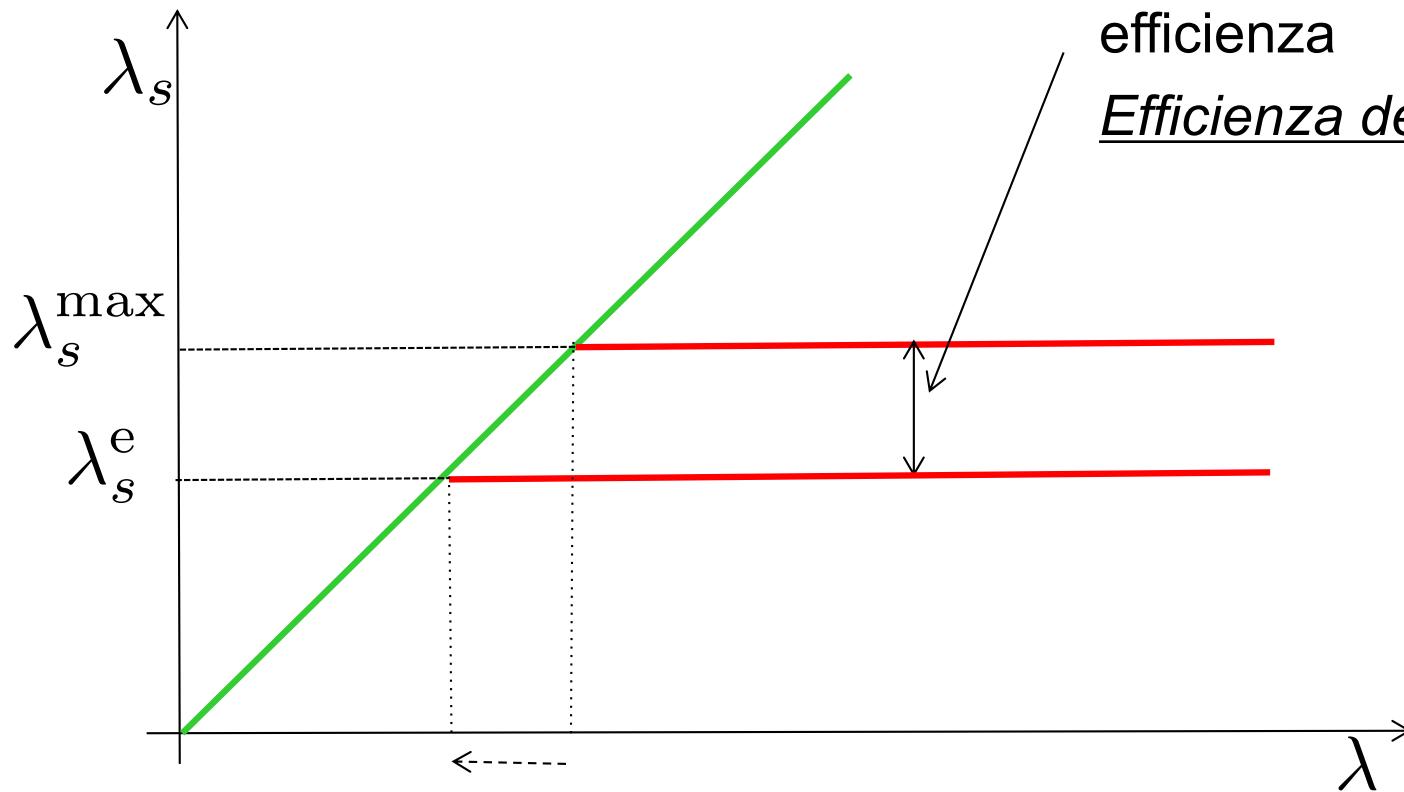


# In un sistema reale

La riduzione di capacità si interpreta come perdita di efficienza

Efficienza del protocollo

$$\eta = \frac{\lambda_s^e}{\lambda_s^{\max}} \leq 1$$



La riduzione di capacità obbliga il livello superiore a limitare la proprie richieste pena la perdita dei dati



# Capacità massima ed efficienza

- Quali sono le prestazioni ideali per un protocollo data link?
  - Poiché il protocollo invia i bit dello strato 3 sul canale la sua capacità massima teorica è la velocità del canale  $C$
- Il tempo medio di servizio minimo possibile sarebbe quindi
$$\bar{\vartheta} = \frac{L}{C} = \frac{1}{\mu}$$
- Se il protocollo richiede maggiore tempo per la completa trasmissione della trama allora

$$\bar{\vartheta}_e = \frac{L}{C_e} > \frac{1}{\mu}$$



$C_e$

- La capacità effettiva dipende dal protocollo
- Se le funzionalità richieste o una situazione non ideale richiedono più tempo per ogni PDU allora parte della capacità risulta inutilizzabile per i dati degli utenti
  - PCI necessarie per la segnalazione
  - Errori di trasmissione
  - Ritrasmissioni
  - Tempi morti legati alle dinamiche del protocollo
  - Tempi morti in attesa di accedere al canale



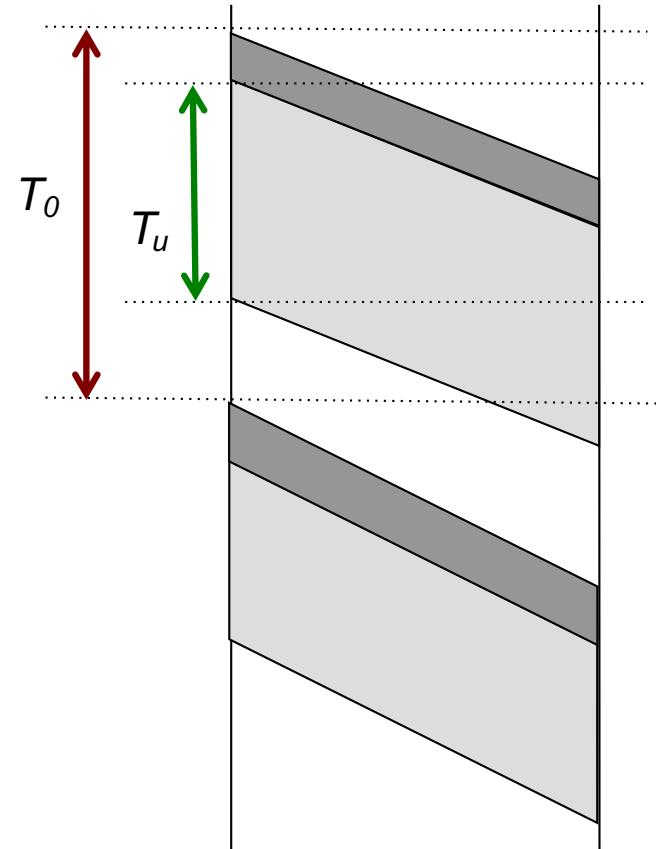
# Unità di misura

- Il traffico è una grandezza adimensionale
  - Moltiplico richieste di servizio per unità di tempo (secondo) per la durata del servizio in unità di tempo (secondi)
- Formalmente si misura con una unità fittizia detta Erlang

$$A_0 = \lambda \bar{\vartheta}$$

# Valutazione efficienza

- Per valutare l'efficienza di solito si fa riferimento alla PDU
- Si confronta:
  - La quantità di tempo strettamente utilizzato per inviare i soli dati d'utente (SDU)
  - La quantità di tempo utilizzato complessivamente per completare correttamente l'invio della PDU
    - In funzione delle regole del protocollo
- L'efficienza è data dal rapporto fra queste due quantità
- Questo problema l'abbiamo studiato lo scorso anno studiando i protocolli ARQ



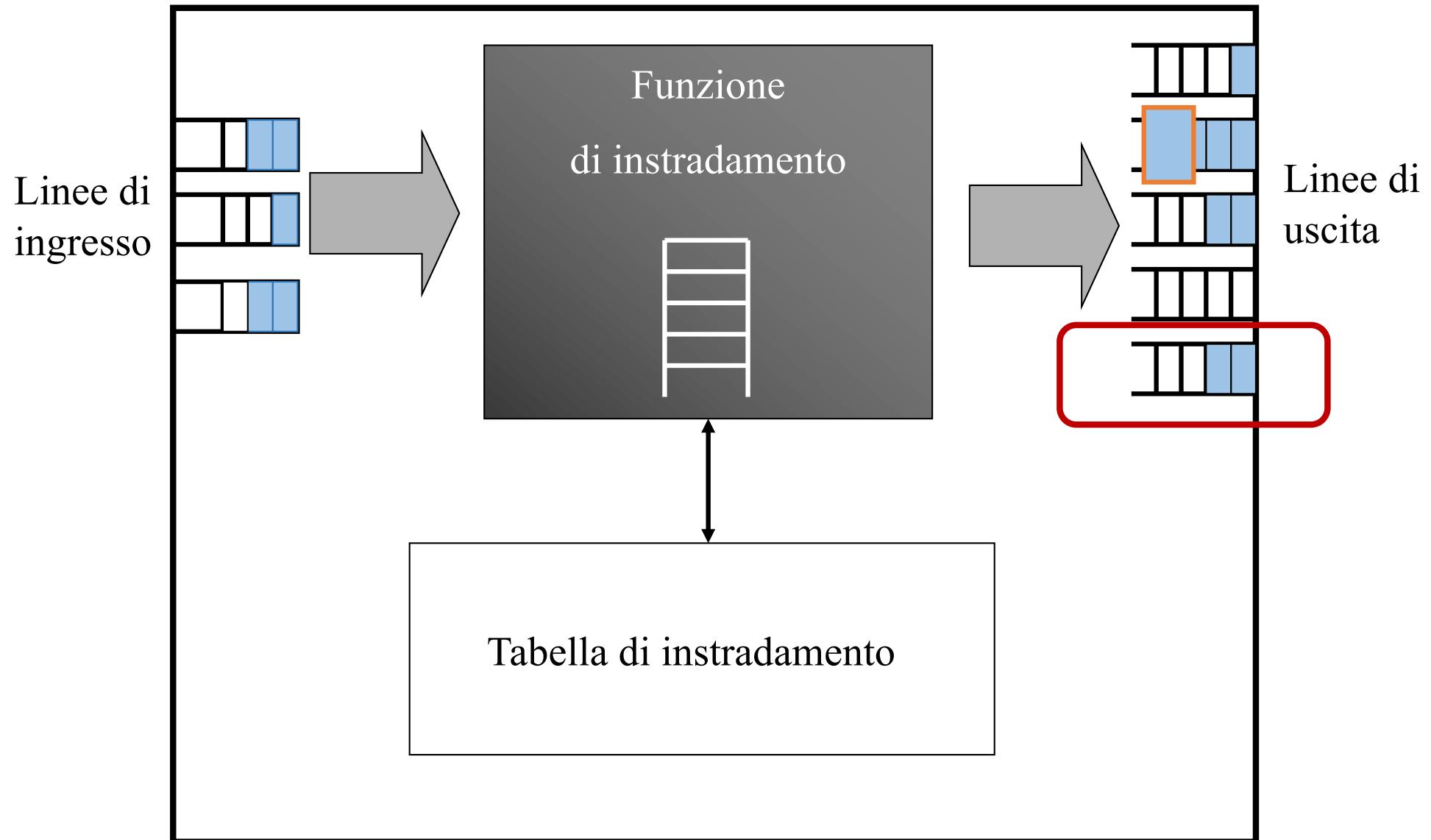
$$\eta = \frac{T_u}{T_0} = \frac{\bar{\vartheta}}{\bar{\vartheta}_e}$$



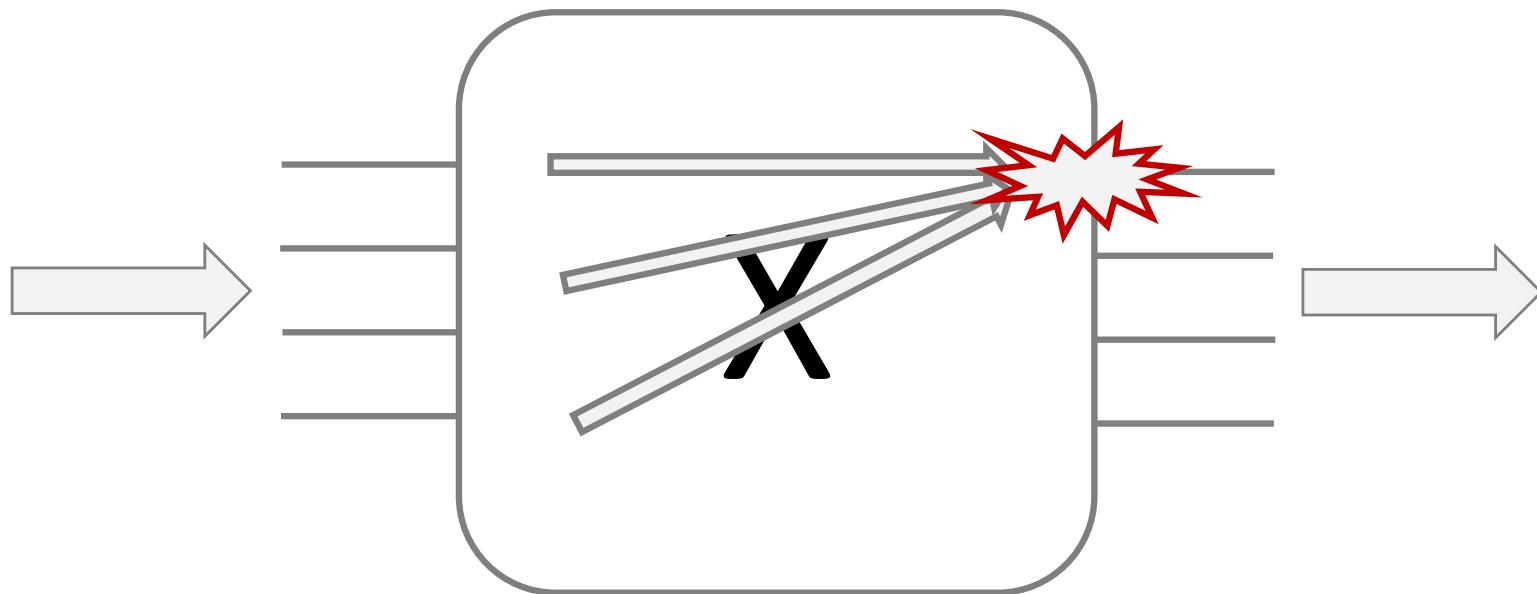
ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Reti commutate: il sistema a coda con singolo servitore

# Il nodo di commutazione a pacchetto



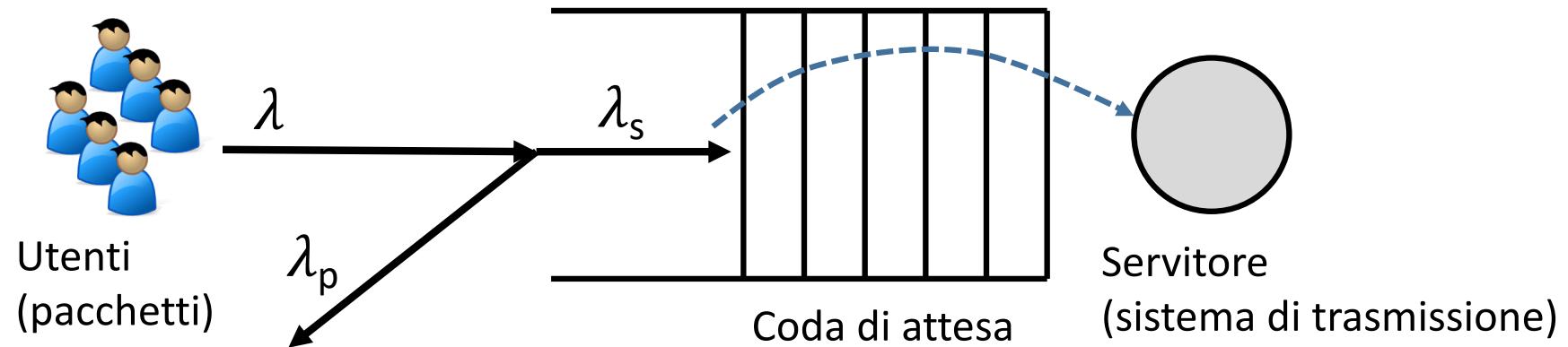
# La congestione





# Coda singolo servitore

- Ogni collegamento in uscita viene schematizzato come sistema coda singolo servitore





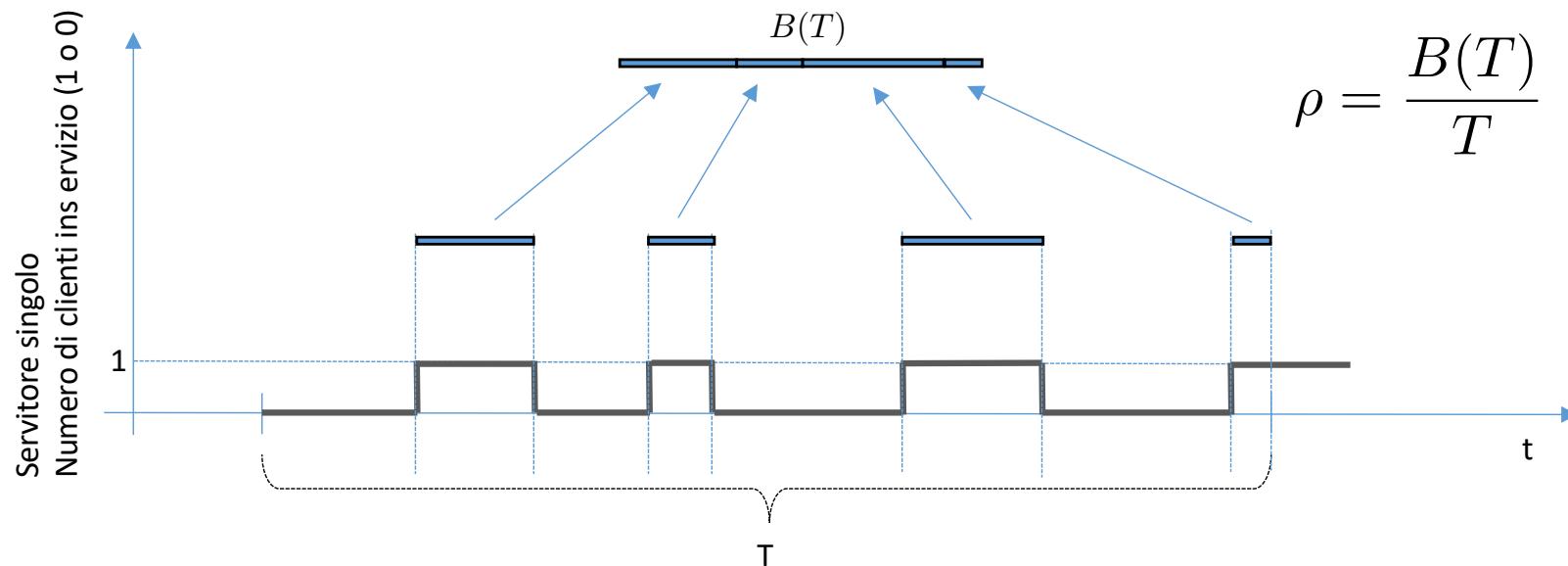
# Alcune ipotesi semplificative

- Le perdite di pacchetti in prima approssimazione sono trascurabili (coda infinita)  $\lambda_p = 0$   
$$\lambda_s = \lambda$$
- I pacchetti arrivano causalmente con distribuzione di Poisson  
$$\Pr \{k \text{ arrivals during } t\} = P(k, t) = \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$
- La dimensione dei pacchetti è casuale con distribuzione esponenziale uguale per tutti i pacchetti

$$\Pr\{\vartheta \leq t\} = F_\vartheta(t) = 1 - e^{-\frac{t}{\vartheta}}$$

# Utilizzazione

- Il servitore alterna fasi di lavoro a fasi di pausa
  - Idealmente vorremmo il servitore sempre attivo
    - Il servitore in pausa è uno spreco di risorse
  - Il servitore può essere attivo solo se ci sono clienti da servire
    - I clienti arrivano in modo casuale e può capitare che ci sia un periodo di tempo senza arrivi





# Servitore occupato

- In un sistema a servitore singolo l'utilizzazione è la percentuale di tempo per cui il servitore è impegnato
- In un sistema ergodico questa è anche la probabilità di trovare il servitore occupato in un istante qualunque
- Quindi

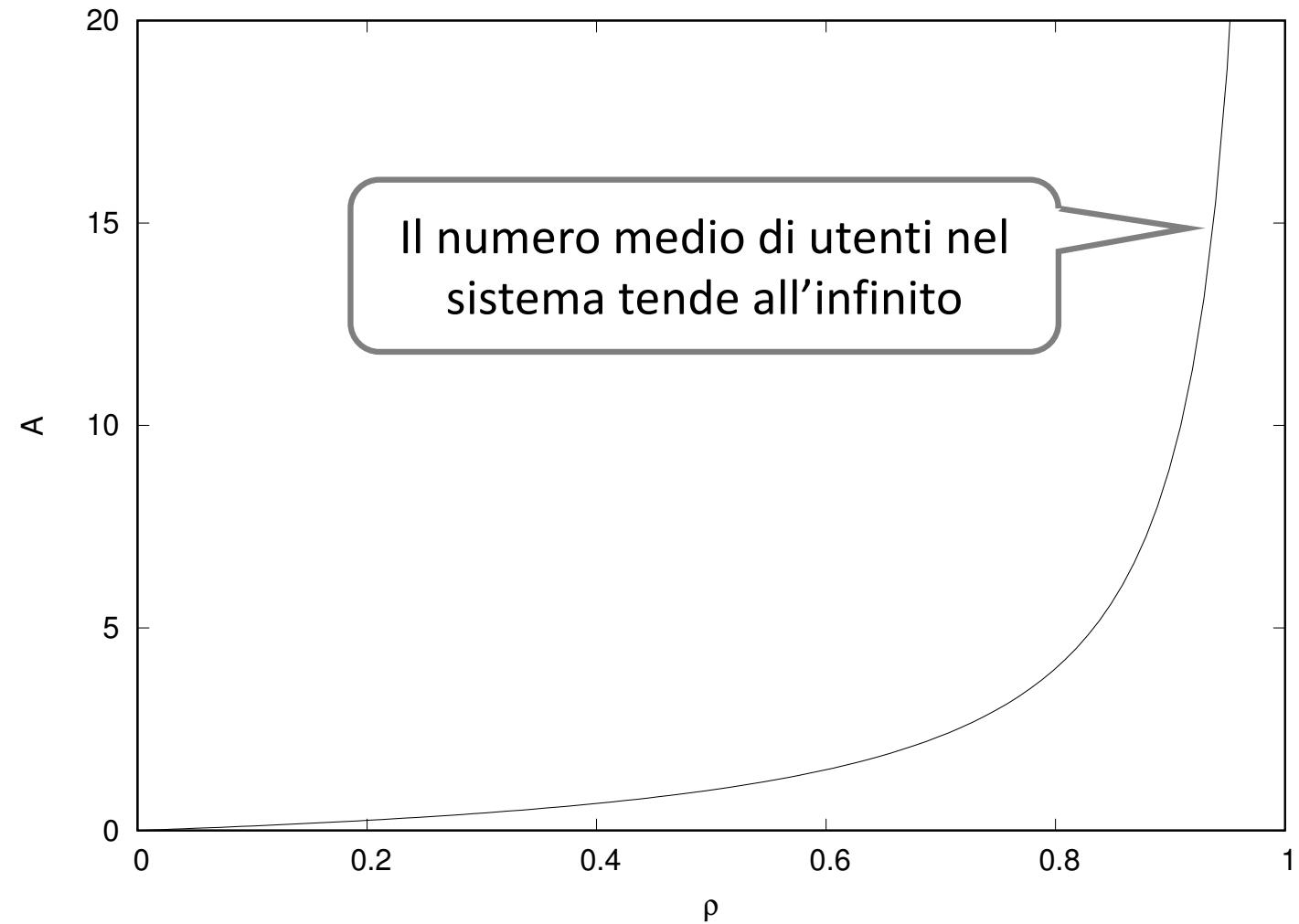
$$\rho = \Pr\{\text{Servitore occupato}\} = \Pr\{\text{Pacchetto accodato}\}$$



# Traffico e utilizzazione

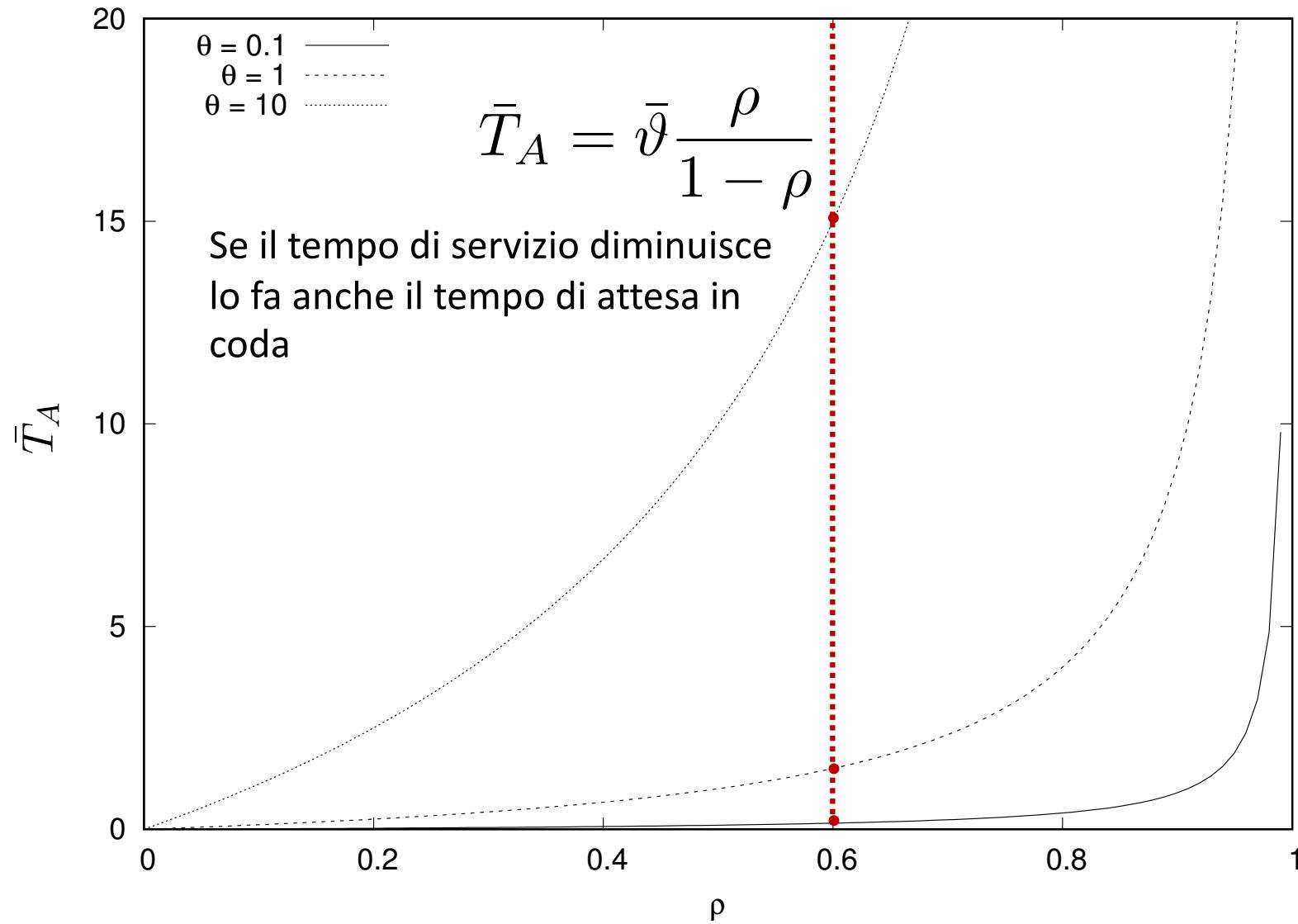
$$\rho = \frac{\lambda}{\mu}$$

Questa grandezza è significativa perché confronta il ritmo di arrivo con quello di servizio





# Tempo medio di attesa in coda



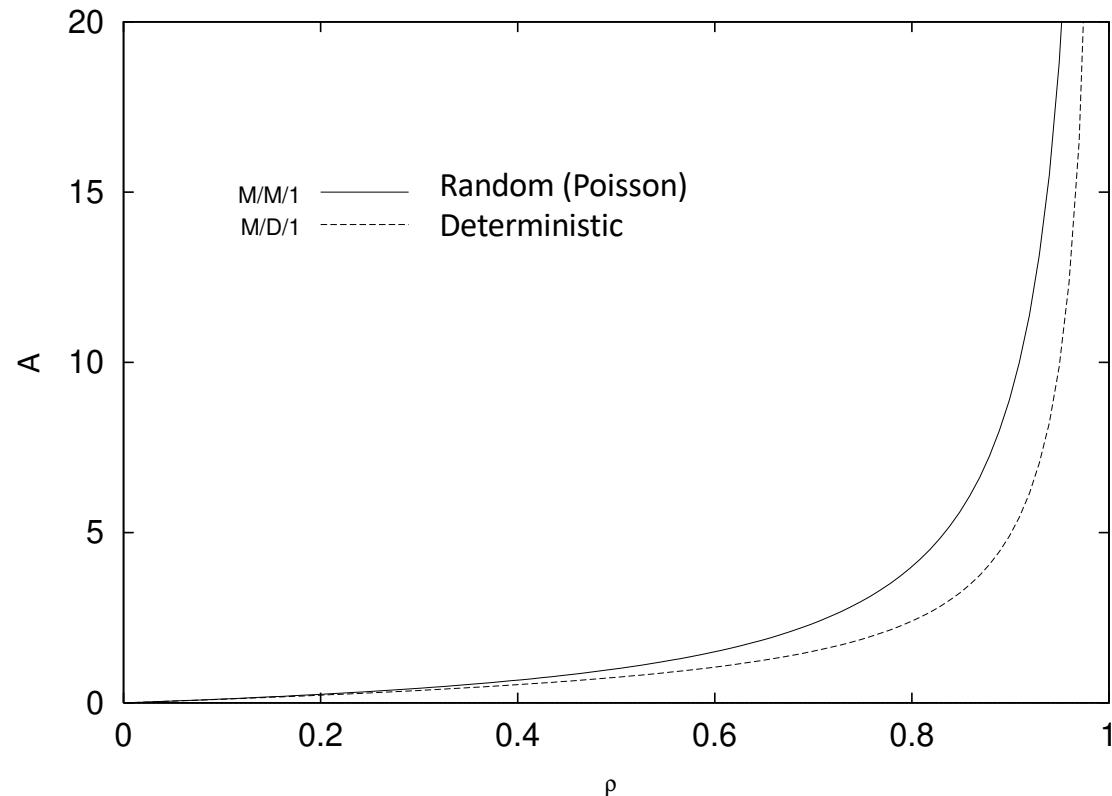


# La statistica del tempo di servizio

- Il traffico nel sistema cambia se cambia la statistica del tempo di servizio
  - Servizio casuale con distribuzione esponenziale (M)
  - Servizio deterministico ossia sempre uguale (D)

$$\bar{T}_A = \bar{\vartheta} \frac{\rho}{1 - \rho}$$

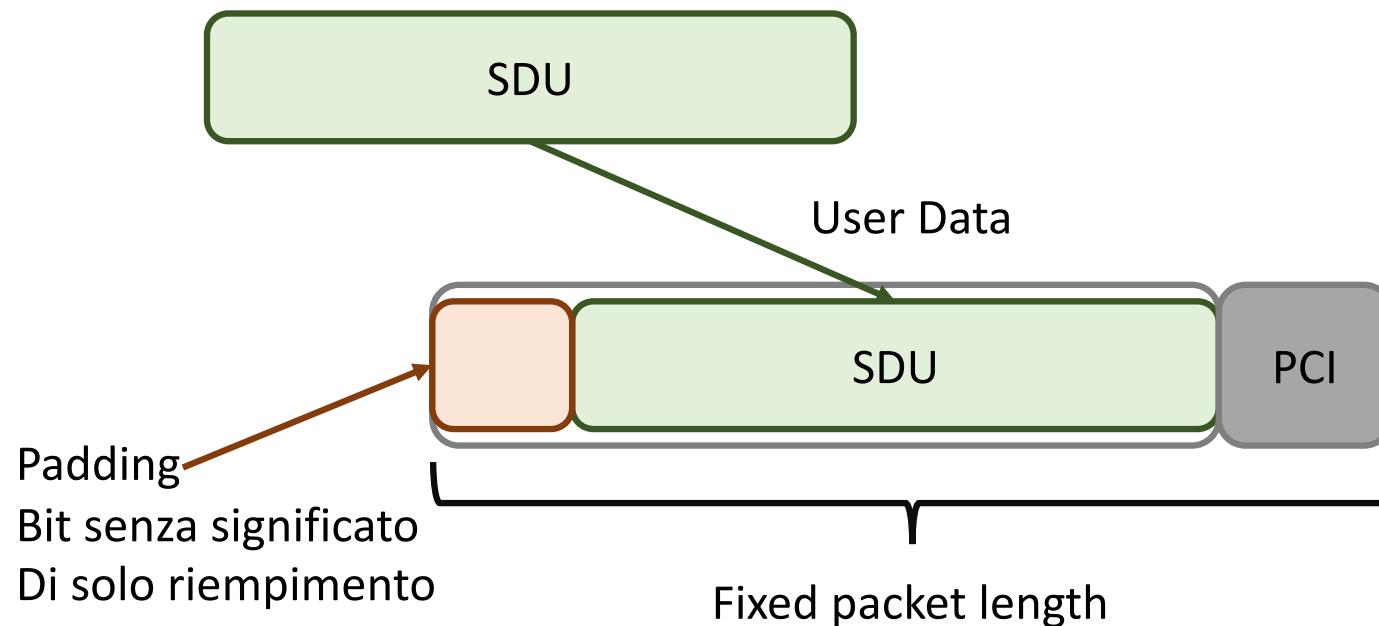
$$\bar{T}_A = \bar{\vartheta} \frac{\rho}{2(1 - \rho)}$$





# Un classico problema nella progettazione dei protocolli

- Pacchetti di lunghezza predeterminata e tutti uguali migliorano le prestazioni in caso di accodamento
- I dati di utente arrivano in quantità casuale quindi è necessario il **padding**





# Il compromesso

- I pacchetti di lunghezza fissa migliorano il tempo di attesa in coda
- Il padding aumenta la lunghezza dei pacchetti, quindi aumenta il tempo di servizio e quindi peggiora il tempo di attesa in coda
- Cosa conviene?

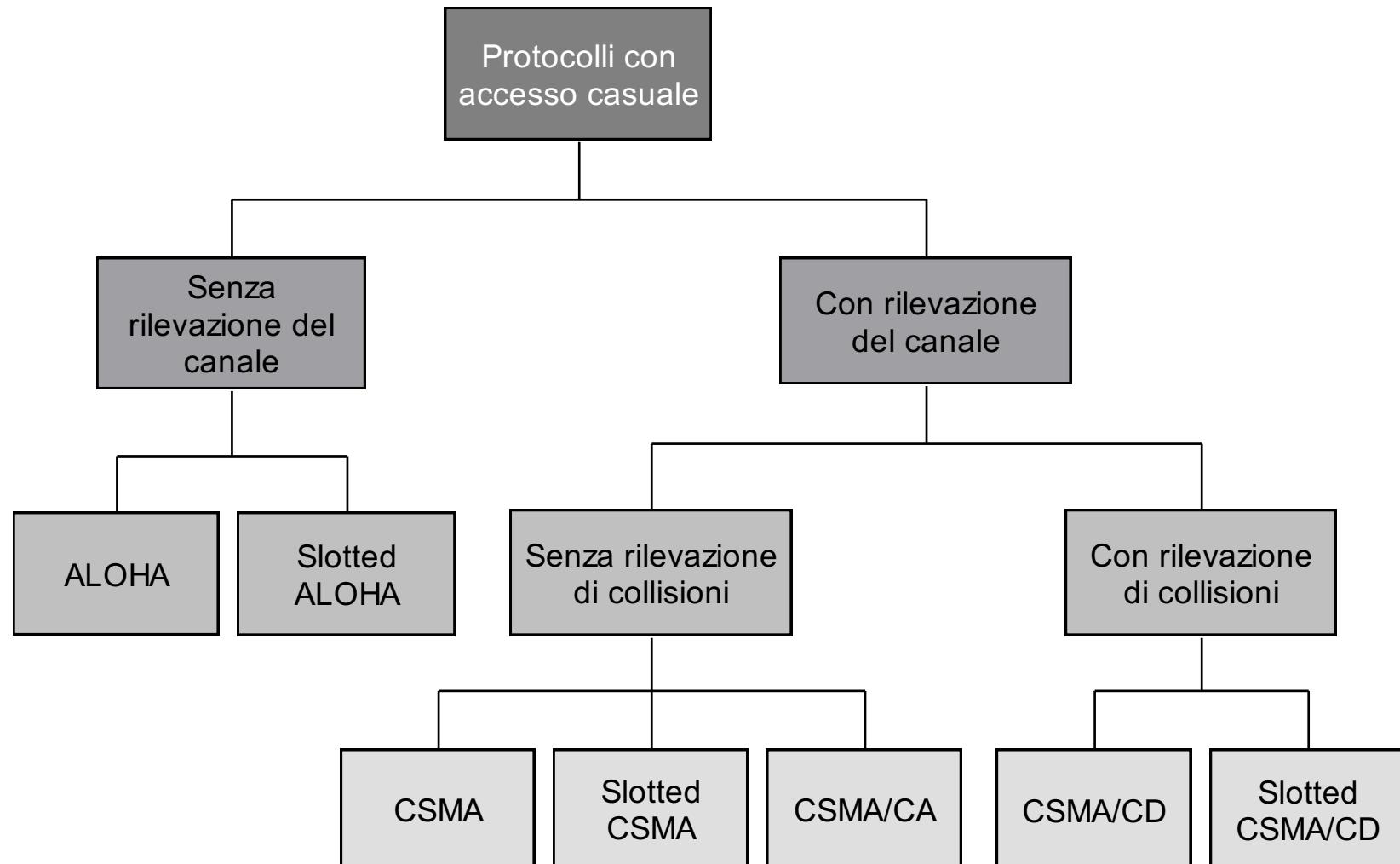


ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Protocolli ad accesso casuale: funzionalità e prestazioni



# Protocolli ad accesso casuale





# Protocolli ad accesso casuale

- CAP - Channel Access Procedure
  - E' l'insieme delle procedure che la stazione effettua per realizzare l'accesso al canale
    - Capacità di capire se il canale è già utilizzato
    - Impossibilità di monitorare il canale per capire se qualcuno lo sta utilizzando
- CRA - Collision Resolution Algorithm
  - E' l'insieme delle procedure che la stazione effettua per rivelare ed eventualmente recuperare situazioni di collisione
    - Capacità o meno di capire se una collisione sta avvenendo od è avvenuta a valle di una trasmissione



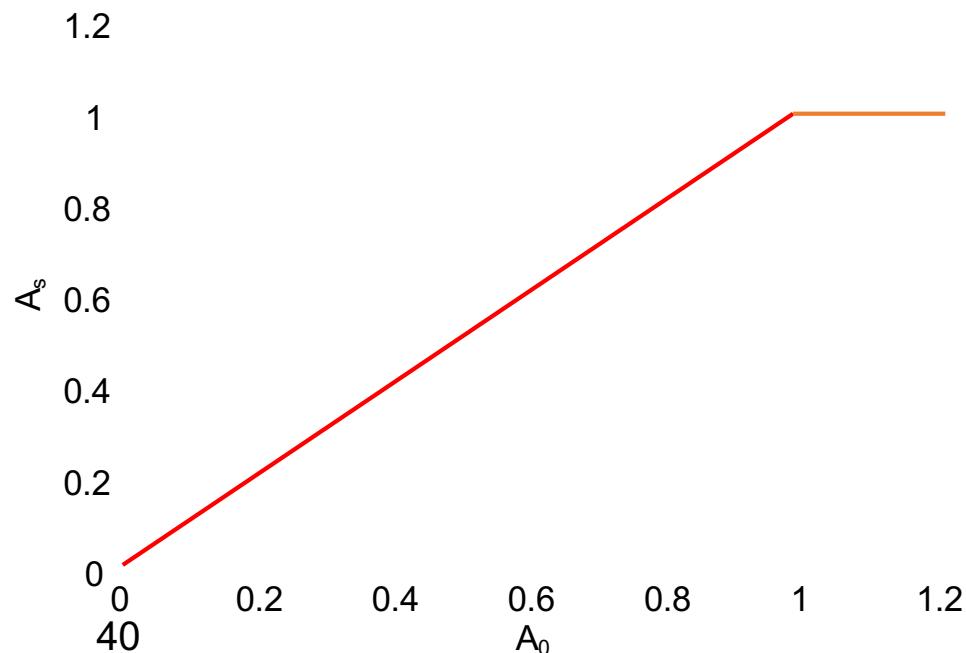
# Parametri caratterizzanti la LAN

- $L$  : lunghezza massima della trama
    - Tutte le trame sono della dimensione massima
  - $C$  : velocità di trasmissione sul mezzo
  - $d$  : massima distanza fra due stazioni della LAN
  - $v$  : velocità di propagazione del segnale
- 
- $\theta = L/C$  : tempo di trasmissione di una trama
  - $d/v$  : tempo di propagazione di un singolo bit sulla LAN
  - $Cd/v$  : massimo numero di bit che possono essere presenti contemporaneamente sulla LAN



# LAN ideale

- Utilizza una CAP ideale
  - Coordina le stazioni per evitare accessi contemporanei al canale di trasmissione
  - Tutte le trame in arrivo vengono trasmesse con successo, quindi  $A_s = A_0$
- Il tempo di propagazione della trama è nullo
- È possibile trasmettere le trame una di seguito all'altra
  - Il canale di trasmissione della LAN può essere utilizzato al 100%



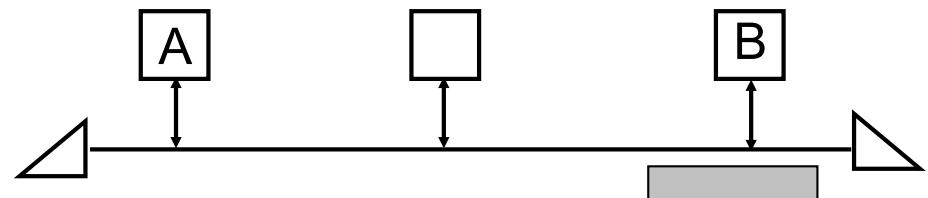
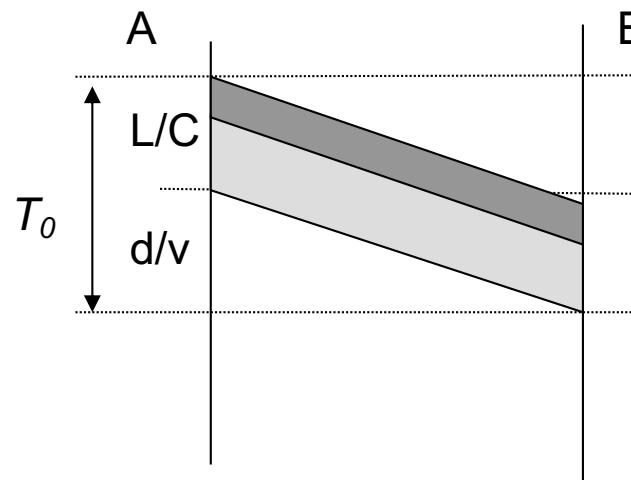
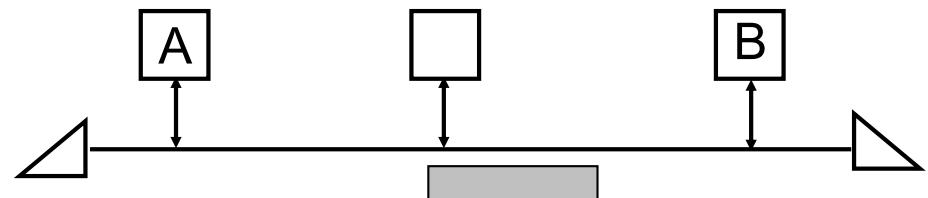
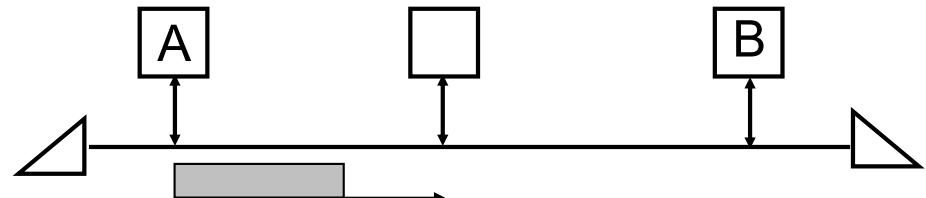
Se  $A_0 < 1$  allora  $\lambda_s = \lambda$

Se  $A_0 \geq 1$  allora  $A_s = 1$

La LAN ideale permette di smaltire tutto il traffico offerto, fino alla saturazione del canale

# Propagazione reale (topologia bus)

- La trama impiega un tempo non nullo per attraversare la LAN
  - $t$  : A inizia la trasmissione
  - $t + L/C$  : A termina la trasmissione
  - $t + d/v$  : B riceve il primo bit
  - $t + L/C + d/v$  : B riceve l'ultimo bit



# Efficienza con MAC ideale

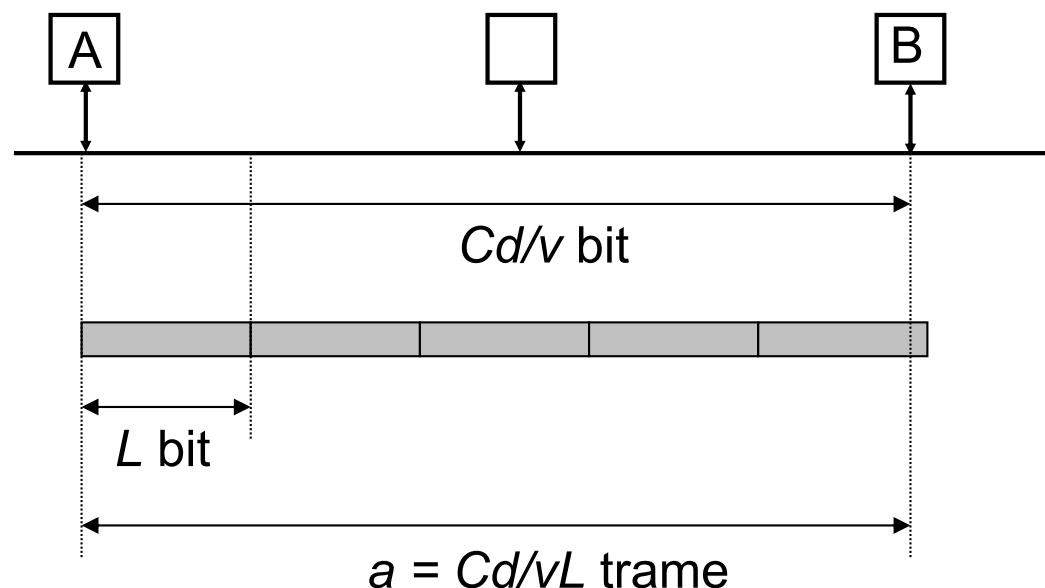
- Una trama tiene impegnata la LAN per  $T_0$
- Il canale di trasmissione non può più essere usato al 100%
- Al massimo viene utilizzato per  $T$  secondi ogni  $T_0$
- **Efficienza** del MAC

$$\eta = T/T_0 = (L/C)/(L/C + d/v) = 1/(1+a)$$

- L'efficienza pone un limite superiore al massimo traffico smaltito  $A_S$

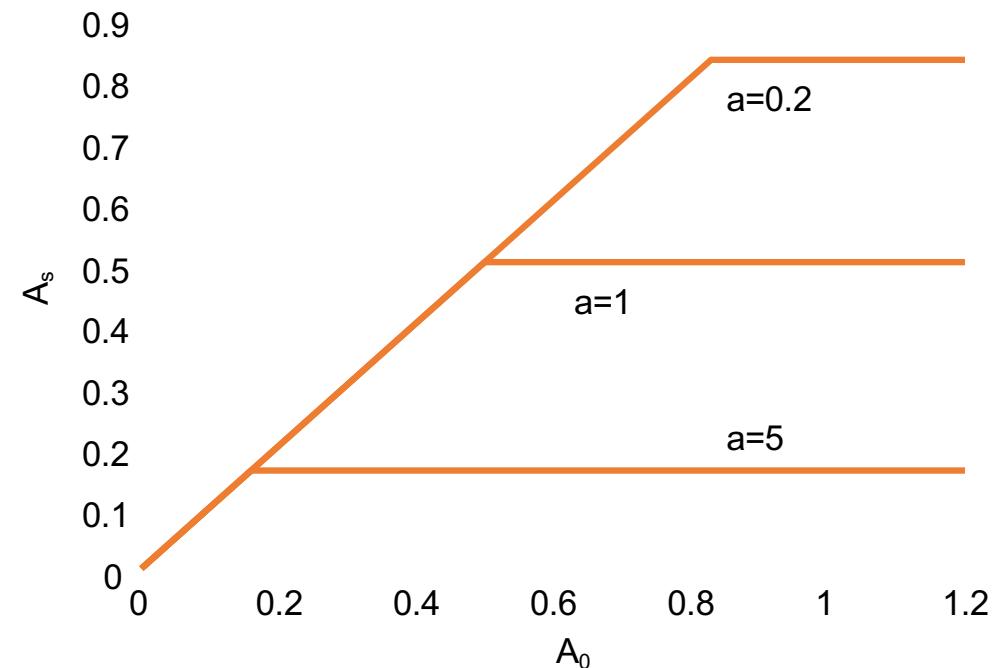
$$a = Cd/vL$$

è interpretabile come la lunghezza della LAN misurata in PDU



# Traffico smaltito dalla LAN

- $A_0 < 1/(1+a)$ 
  - Tutte le trame in arrivo vengono trasmesse
  - $A_s = A_0$
- $A_0 \geq 1/(1+a)$ 
  - Il MAC non permette la trasmissione di tutte le trame
  - Parte delle trame viene accodata
  - $A_s = 1/(1+a) < A_0$





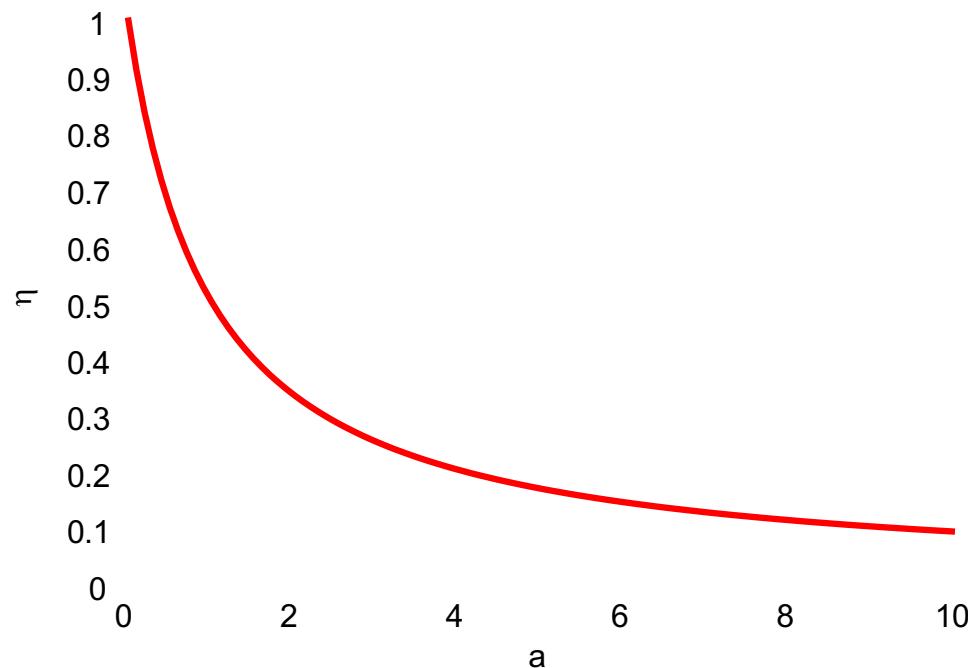
# Esempio numerico

- $d=1 \text{ km}$        $v=200000 \text{ km/s}$        $d/v = 100$
- $C=10 \text{ Mbit/s}$
- $d/v=5 \cdot 10^{-6}$
- $C \cdot d/v = 10 \cdot 10^6 \cdot 5 \cdot 10^{-6} = 50$
- $a = 50 / 100 = 0,5$     $\eta = 1/1,5 = 0,67$
- $C=100 \text{ Mbit/s}$
- $C \cdot d/v = 500$
- $a = 500/100=5$     $\eta = 1/6$
- $C=1 \text{ Gbit/s}$
- $C \cdot d/v = 5000$
- $a = 5000/100 = 50$     $\eta = 1/51$



# Quale efficienza per le LAN

- $a$  determina le prestazioni della LAN
- Maggiore è la lunghezza del canale in trame, minore risulta il traffico massimo smaltibile (massimo throughput)
  - I protocolli ad accesso multiplo sono efficienti quando le distanze e le velocità di trasmissione sono abbastanza limitati





# Protocollo a contesa: ALOHA

- È nato nel 1970 per collegare tra loro le università delle isole Hawaii.
- Prevede stazioni a terra ed un satellite geostazionario
  - Le stazioni trasmettono tutte sul medesimo canale radio (uplink)
  - Il satellite ritrasmette a terra amplificati i dati su un canale diverso (downlink)
- **CAP**
  - Quando un trasmettitore ha una trama da trasmettere la trasmette senza alcun verifica preventiva
  - La trama viene ritrasmessa dal satellite verso tutte le stazioni
  - La stazione trasmittente riceve la propria trama ed ha quindi conferma della corretta trasmissione
- **CRA**
  - Quando due stazioni trasmettono contemporaneamente i segnali collidono e si interferiscono sull'uplink
  - Il satellite scarta le trame non correttamente ricevute
  - La stazione non riceve la propria trama sul downlink e quindi identifica una collisione
  - Non ritrasmette subito ma fa partire l'algoritmo di **back-off**
    - Sceglie l'istante per la ritrasmissione in modo aleatorio all'interno di un intervallo di lunghezza prefissata  $T_b$  (tempo di back-off)



# Aloha: prestazioni

- Assumiamo che i pacchetti generati dalle sorgenti di traffico (applicazioni) determinino gli arrivi di trame alle stazioni secondo un **processo di Poisson** con frequenza media di arrivo  $\lambda$ 
  - Tenendo conto delle ritrasmissioni, il numero medio di pacchetti trasmessi in effetti al satellite nell'unità di tempo è  $\lambda_r > \lambda$
  - Le collisioni con successive ritrasmissioni generano delle **correlazioni** fra gli arrivi, ma le trascuriamo in prima approssimazione (il traffico in arrivo al MAC è ancora di Poisson)



# Traffico offerto e smaltito

- Ipotesi:
  - Trame tutte uguali di lunghezza pari a  $L$  e quindi richiedono un tempo di trasmissione pari a  $T$
- Traffico offerto dalle applicazioni

$$A_0 = \lambda T$$

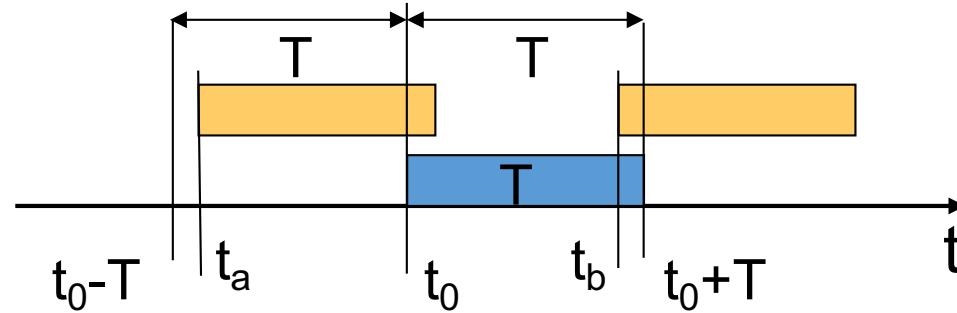
- Traffico offerto al MAC

$$G = \lambda_r T$$

- A causa delle collisioni  $\lambda_r \geq \lambda$
- Il traffico smaltito è pari al traffico offerto che viene trasmesso senza collidere
  - Una trama viene trasmessa senza collidere con probabilità  $P_0$

$$A_s = G P_0$$

# Intervallo di vulnerabilità



- Si definisce **intervallo di vulnerabilità**  $T_v$  l'intervallo all'interno del quale una trasmissione può dar luogo a collisione
- Nel caso di ALOHA vale  $T_v = 2 T$ 
  - La trama considerata inizia in  $t_0$  e finisce in  $t_0 + T$
  - Si ha collisione se
    - il primo bit della trama considerata si sovrapponga all'ultimo bit di una trama precedente
    - Il primo bit di una nuova trama si sovrapponga all'ultimo bit della trama considerata
  - Nessuna trama deve essere trasmessa per un tempo  $T$  prima di  $t_0$  e per un tempo  $T$  successivo a  $t_0$



# Calcolo del Throughput

- La probabilità di non avere una trasmissione in 2T (probabilità di non collisione) è

$$P_0 = e^{-2\lambda_r T} = e^{-2G}$$

- Quindi il numero medio di trasmissioni a venti successo (traffico smaltito  $A_s$ ) è pari a

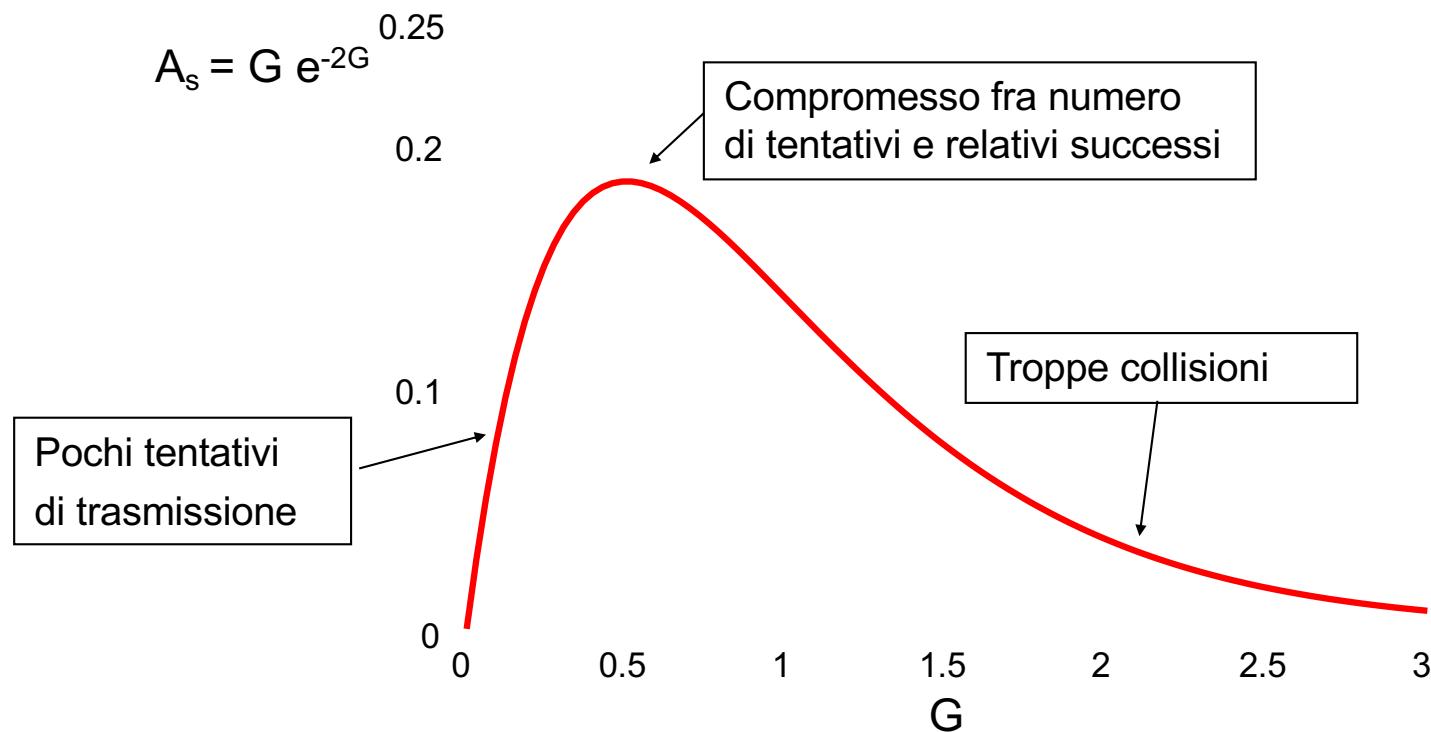
$$A_s = G e^{-2G}$$

- Valore massimo di  $A_s$

$$A_s^{\max} = 1/(2e) \approx 0.18 \text{ per } G = 0.5$$

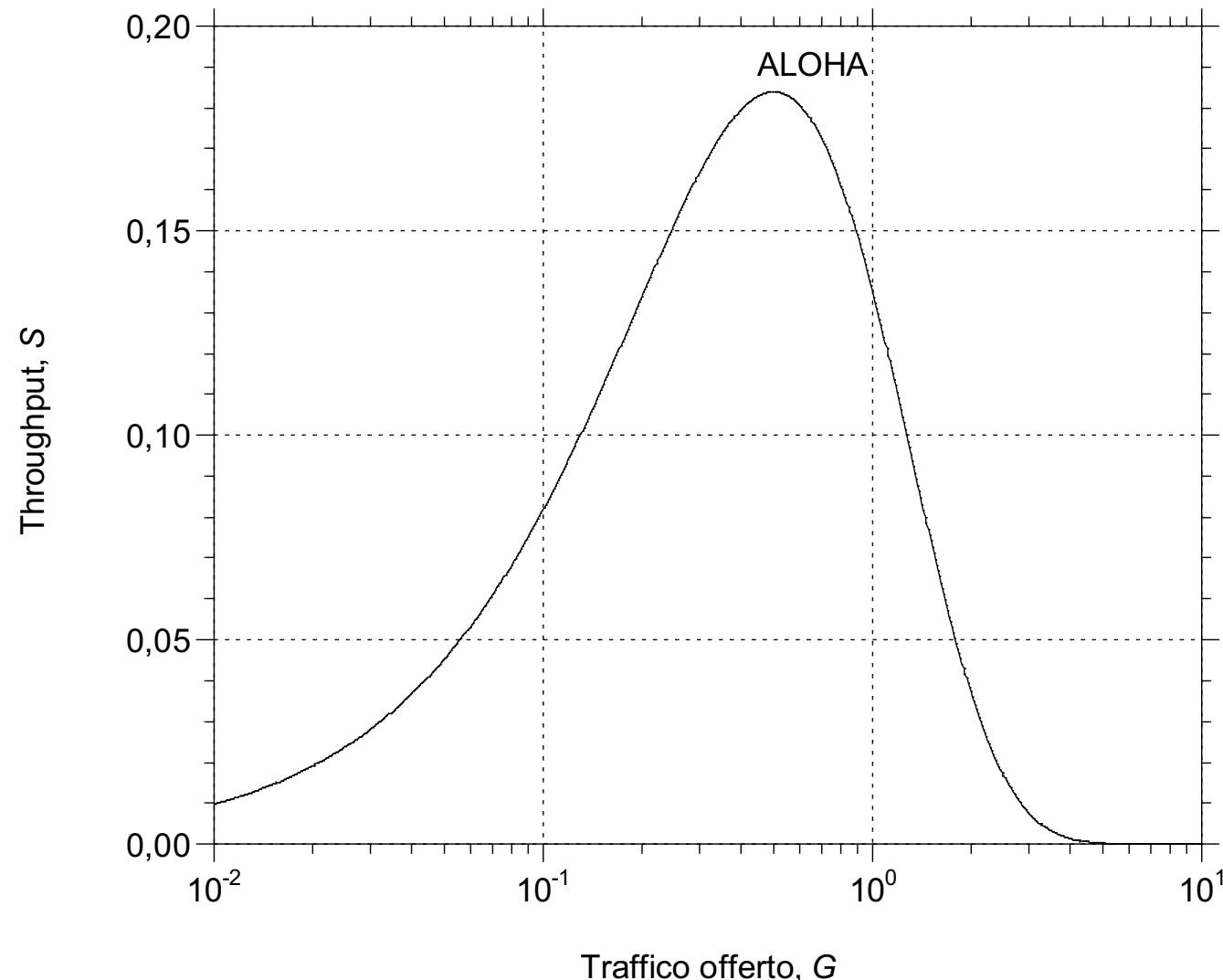
# Aloha: throughput

- $A_s \approx G$  per piccoli valori di  $G$
- $A_s \rightarrow 0$  per grandi valori di  $G$



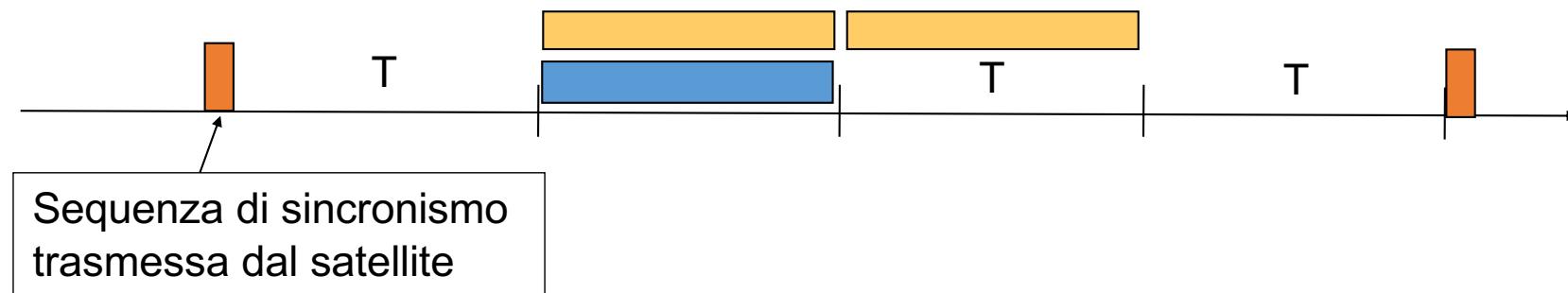


# Throughput di Aloha



# Slotted Aloha

- Un possibile miglioramento: **SLOTTED ALOHA**.
  - Il sistema lavora in modo **sincrono**: l'asse dei tempi viene diviso in intervalli (slot) di lunghezza  $T$
  - Le trame vengono trasmesse in corrispondenza di istanti predefiniti



Prima di iniziare le trasmissioni la stazione deve acquisire il sincronismo, inviando trame di tentativo e rivelando come si posizionano rispetto agli slot

Due trame o si sovrappongono completamente o non si sovrappongono per nulla



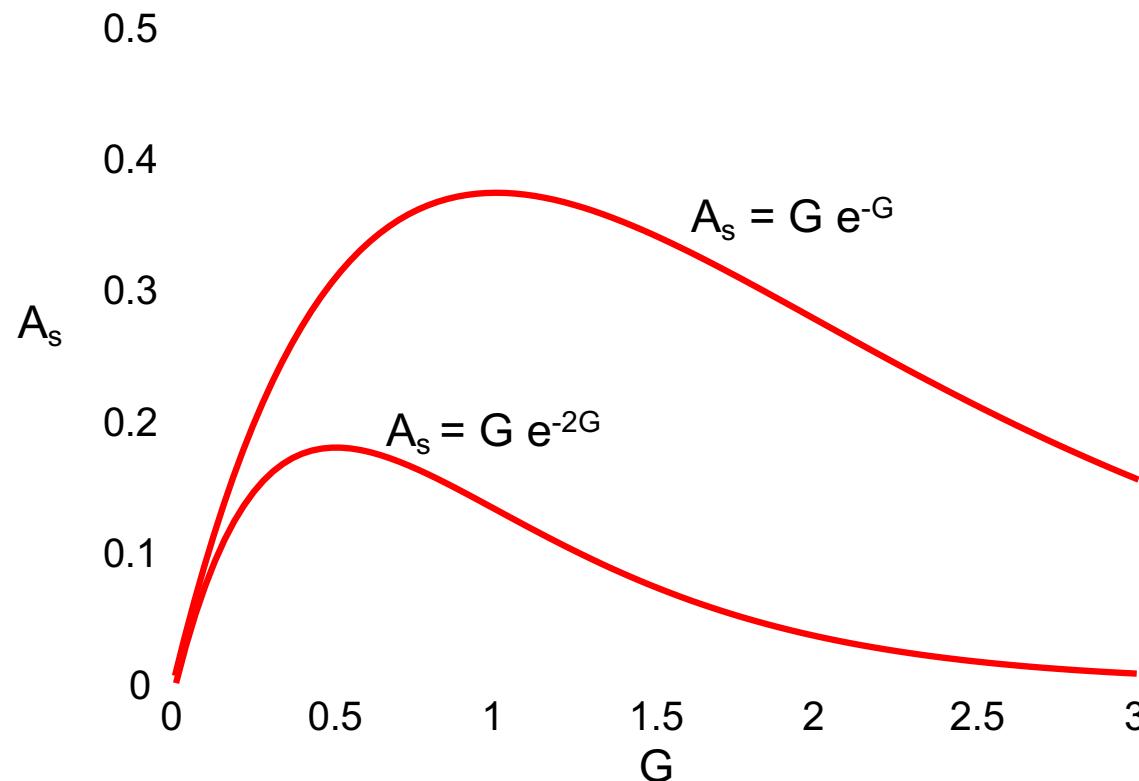
# Slotted Aloha

- L'intervallo di vulnerabilità si riduce a  $T$

- $P_0 = e^{-G}$

- $A_s = Ge^{-G}$

- il massimo di  $A_s$  vale  $A_s^{\max} = 1/e \cong 0.36$  per  $G_{\max} = 1$





# Algoritmi di back-off

- Aloha classico
  - Sceglie a caso (con probabilità uniforme) il nuovo istante di trasmissione nell'intervallo 0 e  $T_b$ 
    - Deve essere  $T_b \gg T$  per rendere piccola la probabilità di una nuova collisione
- Aloha slotted
  - Si ritrasmette negli istanti di sincronismo, ci sono due alternative:
    - Si prende  $T_b = n_b T$  e si sceglie un numero a caso fra 0 ed  $n_b-1$
    - Si ritrasmette nel primo slot utile con probabilità  $p_b$  e si passa allo slot successivo con probabilità  $(1-p_b)$ ; ripetendo l'algoritmo ad ogni slot fino a che non si trasmette
- A parità di valore medio del tempo di ritrasmissione, queste due alternative danno prestazioni simili



# Derivati del protocollo Aloha

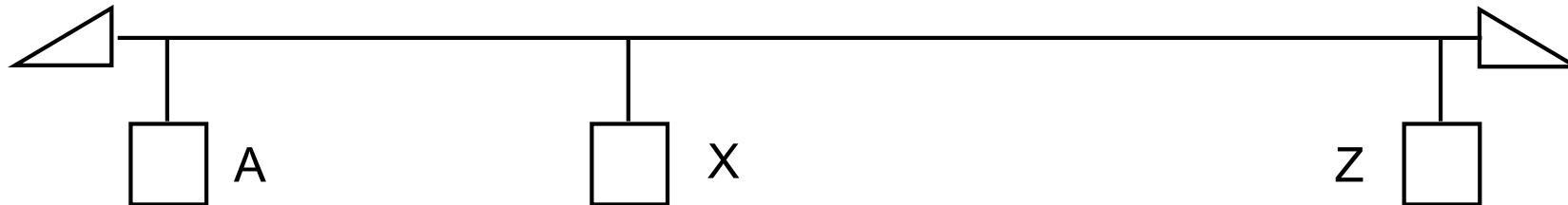
- Il protocollo Aloha può essere implementato su qualunque mezzo trasmissivo e qualunque topologia
- Ha una efficienza piuttosto bassa ma è circa quanto di meglio si può fare quando i ritardi di propagazione sono grandi come nel caso del satellite
- Se lo si vuole applicare ad una rete locale conviene sfruttare la conoscenza che ogni stazione può acquisire sull'attività delle altre
- Nasce così il protocollo **CSMA Carrier Sensing Multiple Access**
  - Viene proposto su una topologia a Bus bidirezionale
  - È ancora un protocollo ad accesso casuale a contesa



# CSMA: Carrier Sensing Multiple Access

- **Carrier sensing**
  - Ogni stazione che debba trasmettere **rivela** presenza di segnale sul bus e trasmette solo se è libero
  - Se il bus è occupato si aspetta la fine della trama e poi
    - Si trasmette (caso 1 persistent)
    - Si fa partire l'algoritmo di back off (caso non persistent o 0 persistent)
    - Si trasmette con probabilità  $p$  e si fa partire l'algoritmo di back off con probabilità  $(1-p)$  (caso  $p$  persistent)
- Una volta iniziata la trasmissione, i dati inviati da una stazione **possono collidere** con quelli di un'altra
  - Questo avviene a causa del ritardo di propagazione non nullo
  - Sul bus non c'è un meccanismo immediato di rivelazione delle collisioni: occorre affidarsi a un sistema di Acknowledgement
- L'algoritmo di back-off può essere come quello dell'Aloha con  $T_b \gg 2\tau$

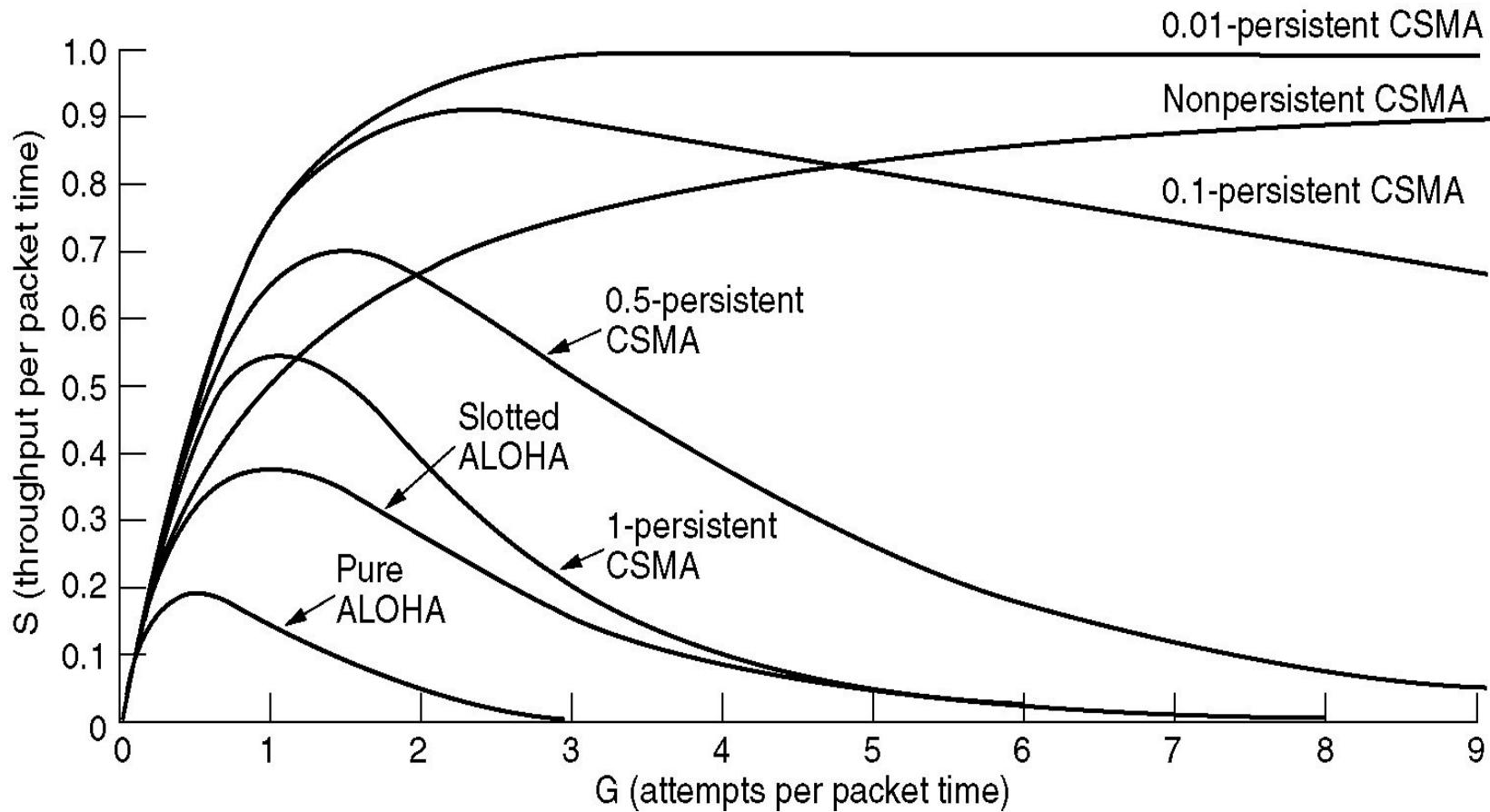
# CSMA: intervallo di vulnerabilità



- Chiamiamo A e Z le due stazioni più distanti sul Bus e  $\tau$  il **tempo di propagazione** fra di loro + il tempo necessario per rivelare il segnale
- A esegue il carrier sensing nell'istante  $t_A$ 
  - Se Z fa carrier sensing fra  $t_A$  e  $t_A + \tau$  non rileva attività e può quindi anch'essa iniziare a trasmettere: *si ha collisione*
  - Analogamente se Z ha trasmesso fra  $t_A$  e  $t_A - \tau$  A non rileva il segnale di Z e trasmette in  $t_A$ : *si ha collisione*
- **L'intervallo di vulnerabilità vale  $2\tau$**
- Le prestazioni sono tanto migliori dell'Aloha quanto più  $\tau/T < 1$ 
  - In generale le prestazioni dipendono anche dal valore di  $p$

# Persistent e Nonpersistent CSMA

Utilizzazione del canale per Aloha e CSMA





# Versione slotted e problemi di stabilità

- Anche per il CSMA esiste la **versione slotted**
- In questo caso la misura più opportuna del **tempo di slot** è  $\tau$
- L'intervallo di vulnerabilità vale  $\tau$  invece che  $2\tau$



# CSMA/CD: CSMA con Collision Detect

- Un miglioramento del CSMA è stato proposto da **Metcalfe** nel 1976
- **Collision Detection:**
  - Una stazione è in grado di rilevare l'avvenuta collisione *rimanendo in ascolto* sul mezzo mentre trasmette
  - E' un **processo analogico** basato sulla **rilevazione di potenza** sul canale (facilitato anche dalla codifica di Manchester adottata)
- In caso di collisione:
  - si ferma subito la trasmissione
  - si invia una particolare sequenza di bits (**jamming**) per informare tutte le altre stazioni dell'avvenuta collisione

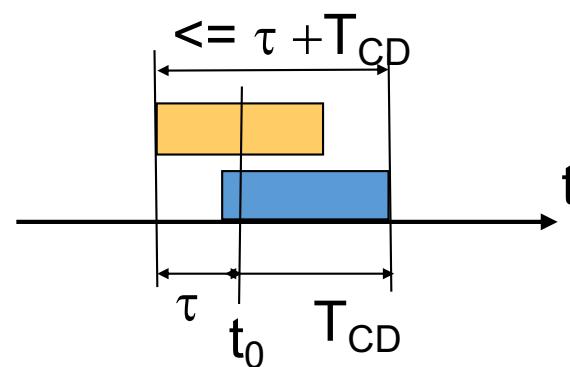
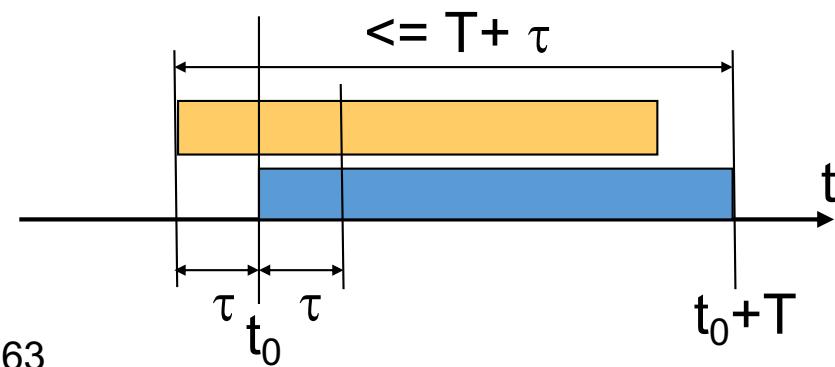


# Codifica di Manchester

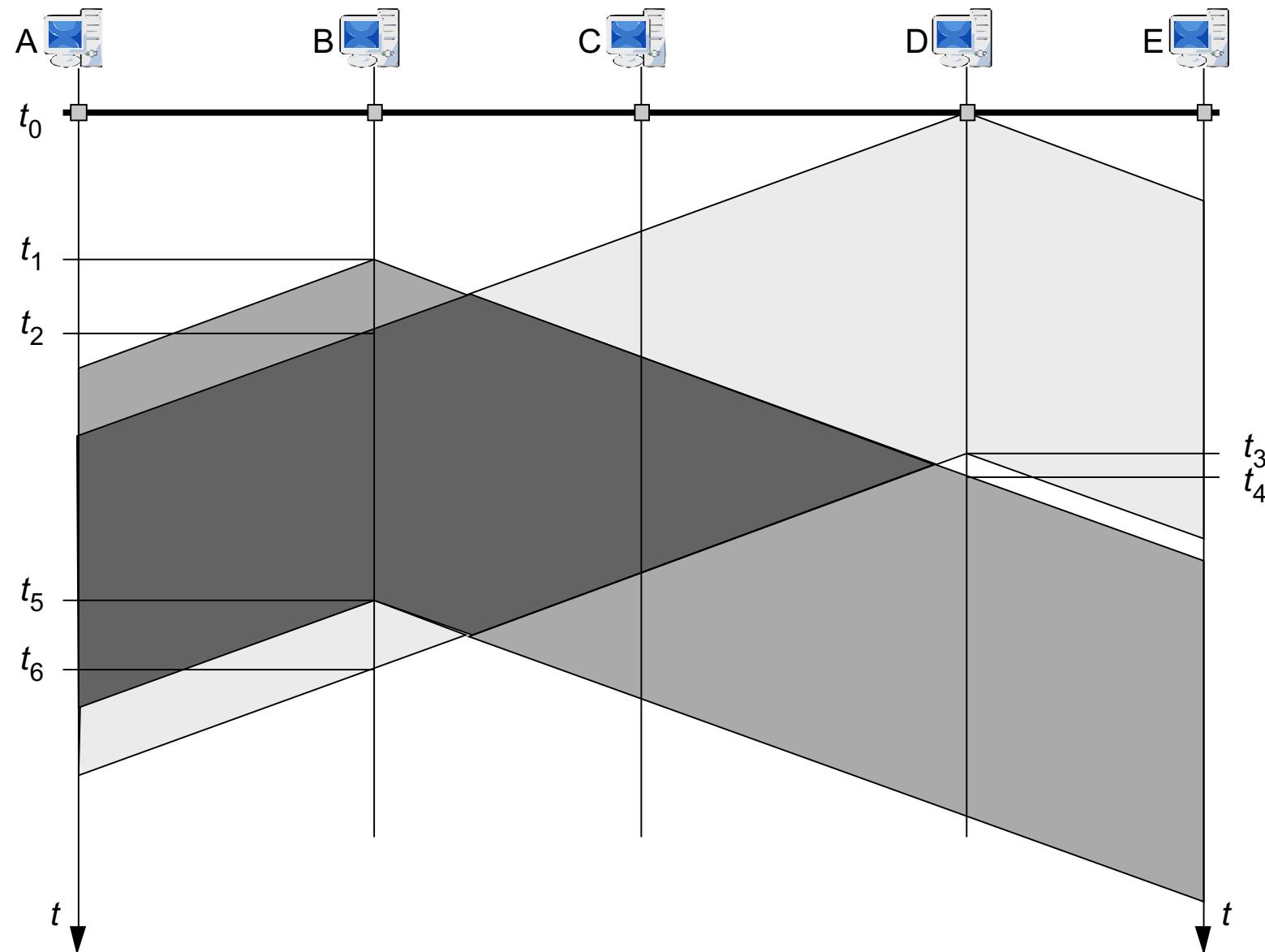
- Rappresentazione dei bit
  - “0” logico: segnale basso (-0.85 Volt) per mezzo tempo di simbolo e segnale alto (+0.85 Volt) per l’altro mezzo
  - “1” logico: segnale alto per mezzo tempo di simbolo e poi segnale basso
- Vantaggi
  - Una transizione al centro di ogni bit, che può essere rivelata mediante un derivatore, facilita
    - L’acquisizione del sincronismo
    - Il carrier sensing
    - Il collision detection
  - Sono disponibili simboli (alto alto e basso basso) per rappresentare *non dati*
- Svantaggi
  - Per trasmettere a 10 Mbit/s occorre un clock a 20 MHz
- Il protocollo CSMA/CD con codifica di Manchester è stato adottato nella rete Ethernet, standard di mercato per le LAN

# Cosa migliora nel CSMA-CD

- Nei casi in cui avviene collisione
  - Nel CSMA le stazioni continuano la trasmissione dell'intera trama
    - Il canale rimane impegnato inutilmente per un intervallo di tempo all'incirca pari a  $T$
  - Nel CSMA/CD
    - Al più il canale rimane impegnato inutilmente al più per la somma di
      - Un intervallo di vulnerabilità ()
      - Il tempo necessario a rilevare la collisione più il tempo della sequenza di Jamming ( $T_{CD}$ )



# CSMA



# CSMA-CD

