Università degli Studi di Bergamo

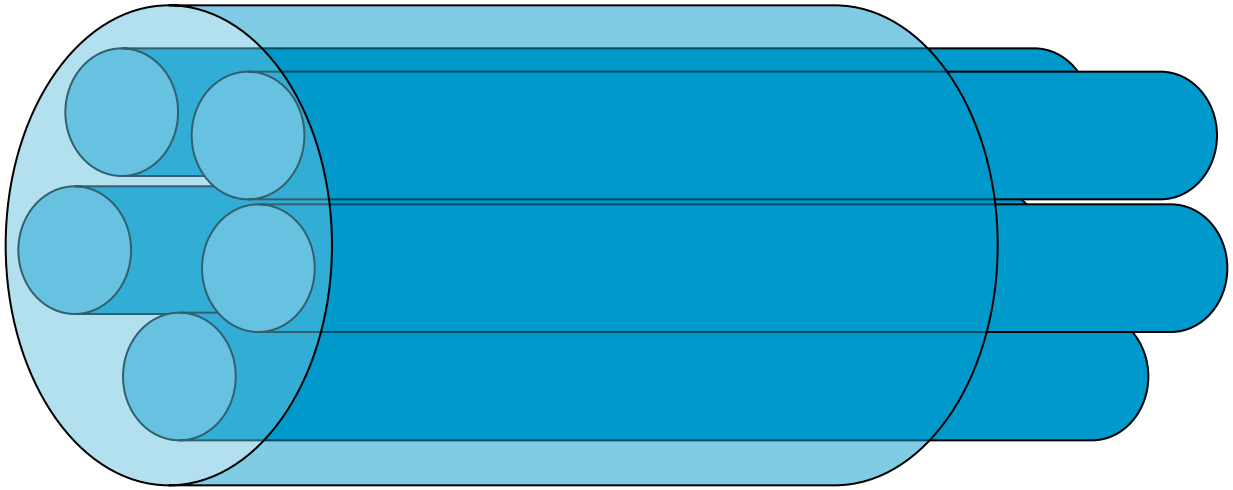
DIP. DI INGEGNERIA DELL'INFORMAZIONE E METODI MATEMATICI

LABORATORIO DI RETI

02 – La Multiplazione Statistica nelle Reti a Paccchetto

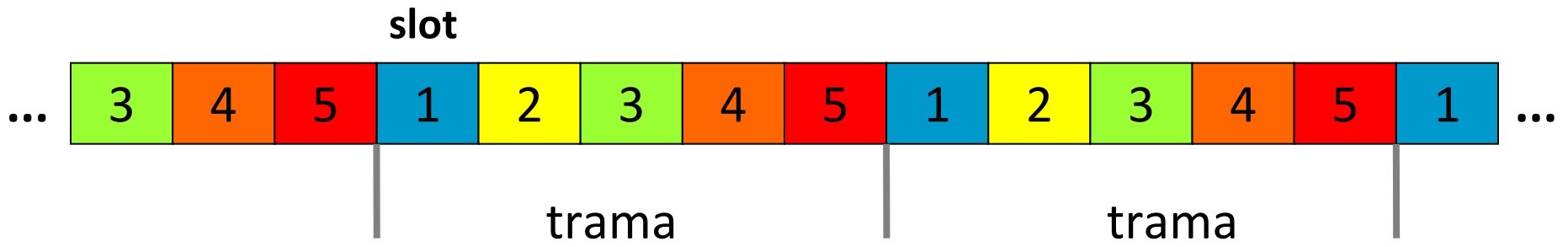
La multiplazione

- La capacità dei mezzi trasmissivi fisici può essere suddivisa per ottenere più canali di velocità più bassa



La multiplazione nelle reti a circuito

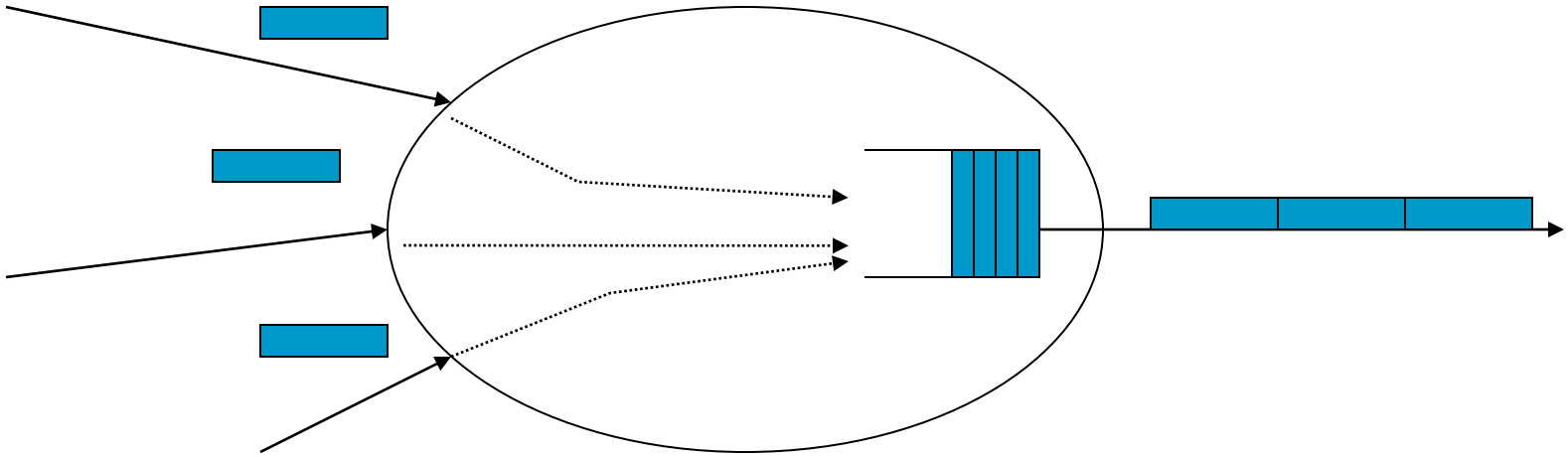
- Nelle reti a circuito come la rete telefonica la multiplazione tra più flussi informativi avviene mediante tecniche di multiplazione statiche (o fisiche) come la tecnica **TDM** (Time Division Multiplexing)



- Un canale può usare uno ***slot*** (intervallo di canale) ogni *N*
- si definisce una struttura a ***trame*** consecutive costituite da N slot consecutivi
- se si numerano ciclicamente gli slot delle trame, un canale è associato a un numero di slot

La multiplazione nelle reti a pacchetto

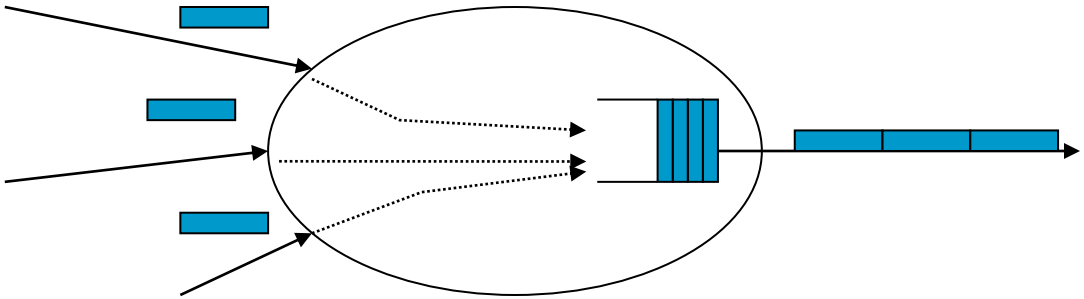
- Nelle reti a pacchetto l'informazione relativa ad un flusso non ha un canale dedicato
- I flussi di pacchetti condividono le risorse trasmissive della rete (link di collegamento tra i nodi)
- all'arrivo in un nodo, ogni pacchetto viene memorizzato, analizzato e trasferito verso il link d'uscita



NODO

La multiplazione nelle reti a pacchetto

- Se il link d'uscita è libero il pacchetto viene immediatamente trasmesso
  - il tempo di trasmissione T_t è pari a L/C, dove L è la lunghezza del pacchetto (in bit) e C è la capacità del link (in bit/s)
- Se il link d'uscita è occupato il pacchetto viene messo in una coda d'attesa (buffer)
  - il tempo di attesa prima della trasmissione dipende da: il numero di pacchetti già in coda, la loro lunghezza, la politica di gestione della coda (es. First In First Out - FIFO).



La multiplazione nelle reti a pacchetto

- In generale gli istanti di arrivo dei pacchetti in un nodo sono casuali e comunque non sincroni come nel caso degli slot TDM
- Per questo, le trasmissioni dei diversi flussi sul canale si alternano sulla base dell'ordine di uscita dalla coda in un modo che può dirsi casuale
- per questo nel caso di reti a pacchetto si parla di ***multiplazione statistica***

La multiplazione nelle reti a pacchetto

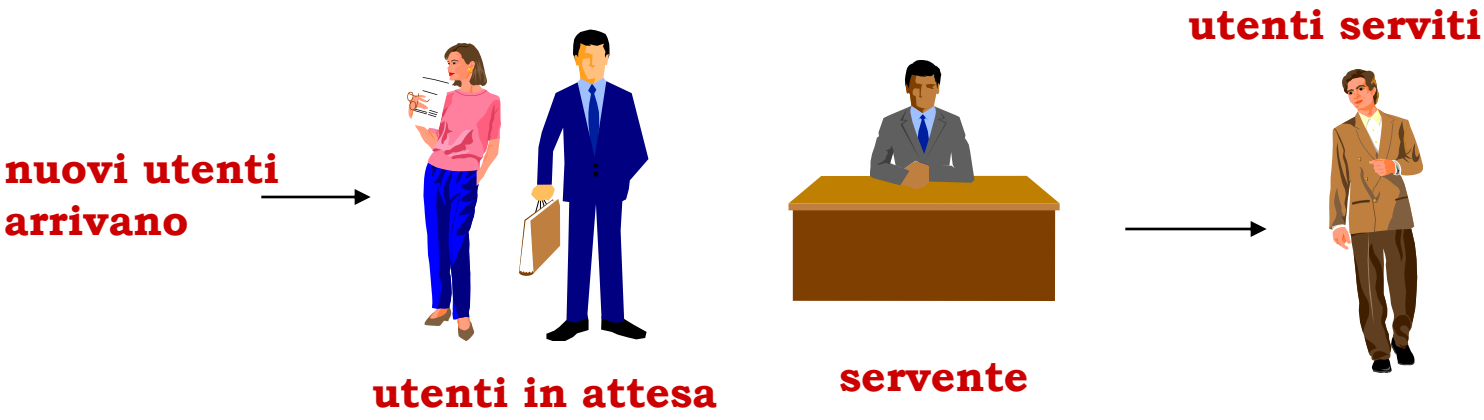
- Il grande **vantaggio** della **multiplazione statistica** è quello di consentire di **usare le risorse** delle rete **quando necessario** senza necessità di riservare risorse per ciascun flusso come nel caso del circuito
- Si può dimostrare che questo porta ad un **miglior utilizzo delle risorse** quando le **sorgenti** di traffico sono **discontinue** (come di solito per i servizi dati)
- Il **prezzo da pagare** per questa maggiore efficienza e flessibilità è un **ritardo** di trasferimento **variabile** che dipende dalle condizioni di traffico
- Il **ritardo di accodamento** nei buffer è la componente **variabile** del ritardo di trasferimento

La multiplazione nelle reti a pacchetto

- Per studiare in modo quantitativo il ritardo di trasferimento (e di accodamento in particolare) ci sono due mezzi:
  - teoria delle code (metodi analitici - corso di “Reti di Telecomunicazione” - Laurea Specialistica)
  - simulazione (è il nostro caso!)
- I modelli di sistemi su cui si basa sia la teoria delle code che la simulazione sono i cosiddetti:

Sistemi d’attesa

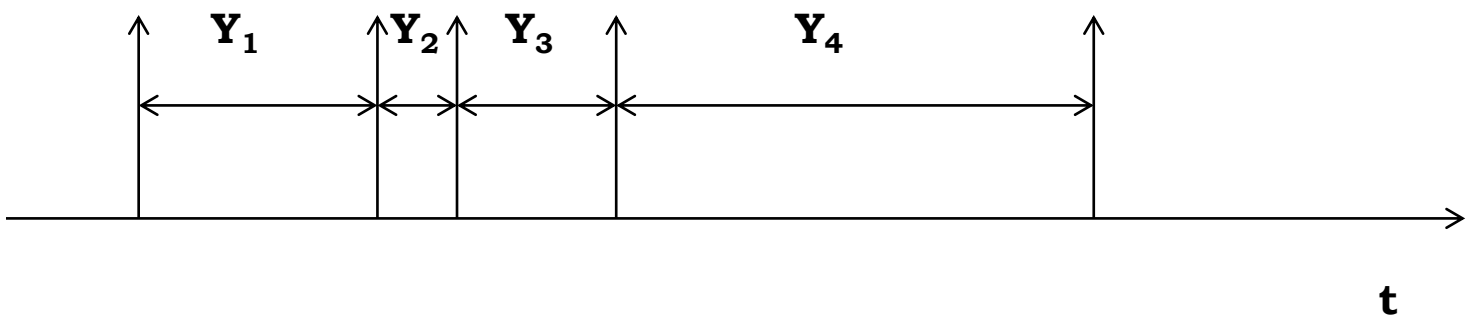
Sistemi d'attesa



Sistemi d'attesa

- I sistemi d'attesa sono caratterizzati da:
  - processo degli arrivi (processo statistico che descrive gli arrivi dei pacchetti)
  - processo dei tempi di servizio (lunghezza dei pacchetti e capacità del link che determinano i tempi di trasmissione)
  - politica di gestione della coda

Sistemi d'attesa: processo degli arrivi



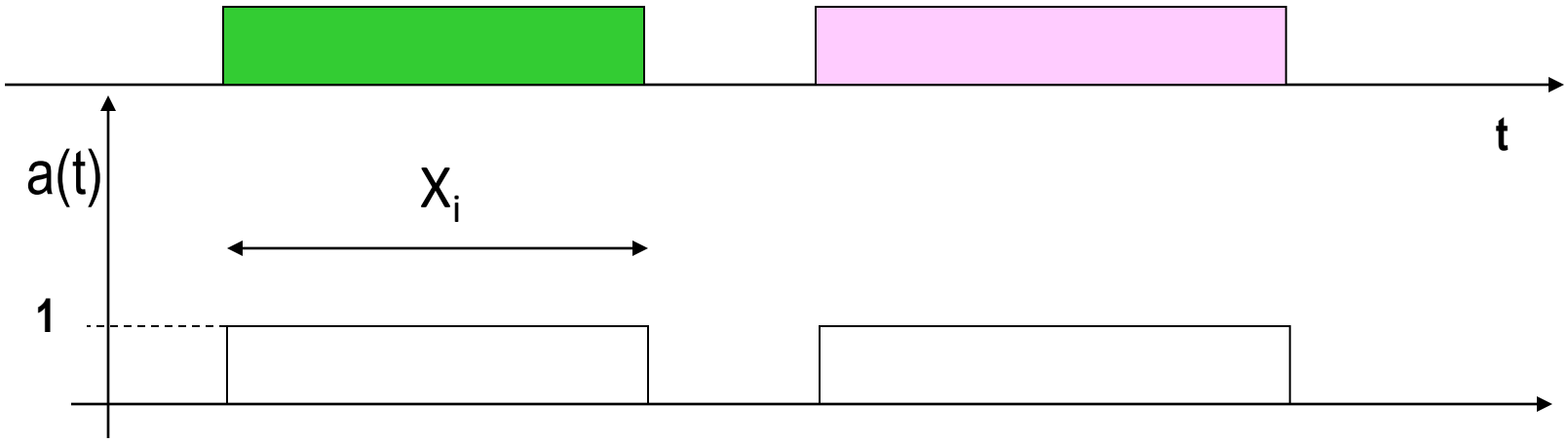
- E' il processo che descrive gli istanti d'arrivo
  - si può usare
    - ***N(0,t)*** numero di arrivi in (0,t)
    - ***{Y_k}*** sequenza dei tempi di interarrivo
  - parametri caratteristici:
    - numero medio di arrivi nell'unità di tempo **λ** (pacchetti/secondo)

Sistemi d'attesa: processo dei tempi di servizio

- E' il processo che descrive la durata del servizio (trasmissione)
  - $\{X_k\}$ sequenza dei tempi di servizio
  - $X_k = L_k / C$, dove L_k è la lunghezza del pacchetto e C la capacità del link
  - parametri caratteristici:
    - **E[X]** ➔ tempo medio di servizio
    - **μ = 1/E[x]** ➔ frequenza media di servizio

Sistemi d'attesa: traffico

- Nelle reti si usa spesso il termine *traffico* per indicare la quantità di informazione gestita da una sistema di trasmissione
- Traffico istantaneo:
  - il numero di pacchetti ***a(t)*** in trasmissione su un link al tempo ***t***
  - (se c'è un solo canale per link il traffico istantaneo può valere solo 0 e 1)



Sistemi d'attesa: traffico

- Traffico medio (in T):

$$A(T) = \frac{1}{T}\int_T a(t)dt$$

- Risultati:

$$\int_T a(t)dt = \sum_i X_i$$

$$\frac{1}{T}\int_T a(t)dt = \frac{n}{T}\frac{\sum_i X_i}{n}$$

$$A = \lambda E[X] = \frac{\lambda}{\mu}$$

n numero di arrivi in T

Sistemi d'attesa: traffico

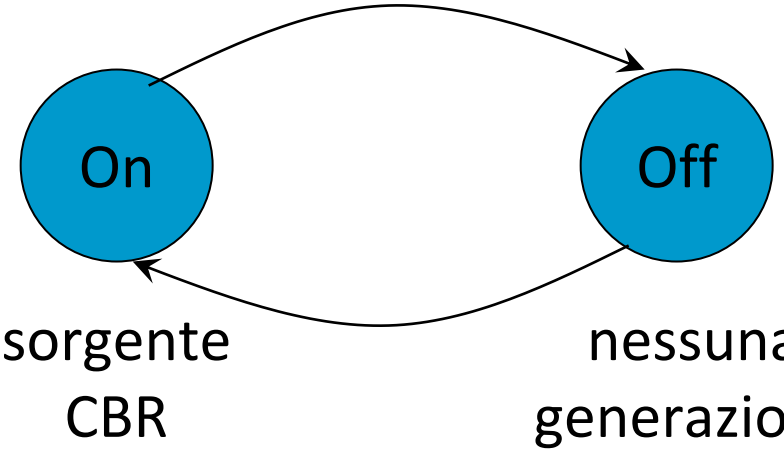
- Se un pacchetto arriva e non trova posto viene scartato (overflow del buffer)
- Possiamo definire:
  - λ_o ➔ frequenza di pacchetti offerti
  - λ_a ➔ frequenza di pacchetti accettati
  - λ_r ➔ frequenza di pacchetti persi
  - λ_s ➔ frequenza di pacchetti smaltiti ($\lambda_a = \lambda_s$)
- Definizioni analoghe valgono per il traffico

Sorgenti di traffico

- Abbiamo visto alcune semplici sorgenti di traffico in NS
- Sorgente CBR
  - $X_k = X = L/C \quad \forall\, k$
  - $Y_k = Y = L/R \quad \forall\, k$
  - $\lambda = 1/Y$
  - $\mu = 1/X$

R: rate di emissione della sorgente

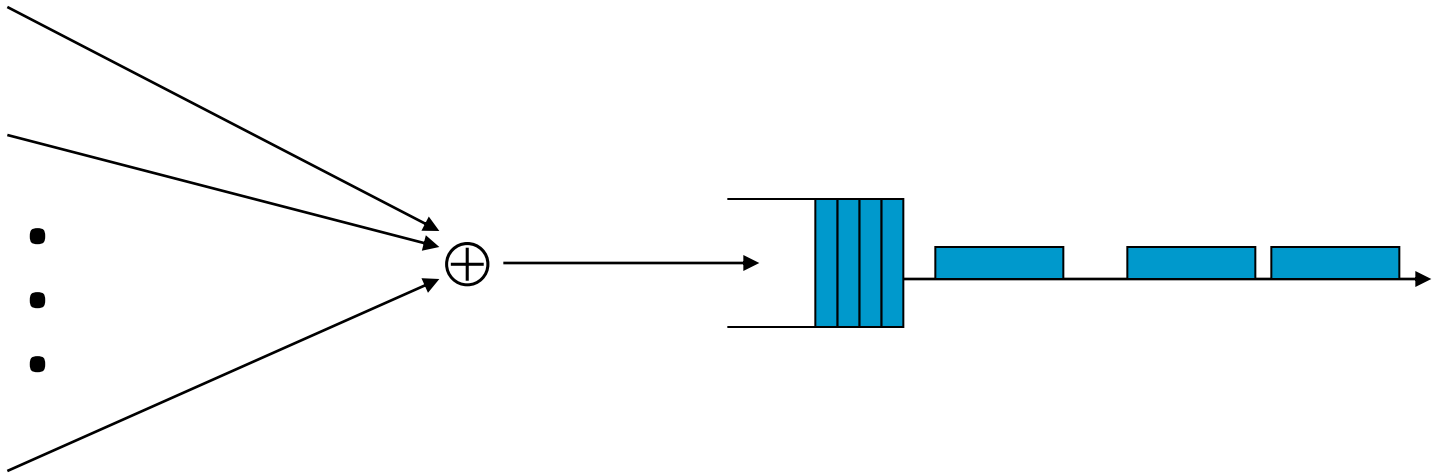
Sorgente ExpOnOff



- se più sorgenti confluiscono in un link le frequenze d'arrivo si sommano

Sorgenti di traffico

- Si può dimostrare che:
  - ***al tendere all'infinito del numero di sorgenti di traffico, il traffico complessivo tende ad essere un traffico di Poisson***



Arrivi di Poisson

- Il processo degli arrivi di Poisson è descritto (probabilisticamente) da 3 assiomi
  - (1) La probabilità che ci sia un punto di Poisson in un intervallo ***Δt*** è pari a:

$$P\left[N(t,t+\Delta t)=1\right]=\lambda\Delta t+o(\Delta t)$$

  - dove il parametro ***λ*** si dimostra essere la ***frequenza del processo*** (in arrivi per unità di tempo).
  - (2) La probabilità che ci siano più punti in un intervallo ***Δt*** è pari a **o(*Δt*)**
  - (3) Il numero di punti presenti in intervalli di tempo disgiunti sono variabili causali indipendenti e dipendono solo dall'ampiezza degli intervalli.

Arrivi di Poisson

- Si può mostrare che:
  - La probabilità che il numero di arrivi (punti) di Poisson ***N(t,t+τ)*** in un intervallo temporale fra ***t*** e ***t+τ*** è pari a:

$$P[N(t,t+\tau)=k]=\frac{(\lambda\tau)^k}{k!}e^{-\lambda\tau}$$

  - gli inter-arrivi Y_k sono variabili casuali indipendenti con densità di probabilità esponenziale negativa

$$f_T(t)=\lambda e^{-\lambda t} \qquad (t>0)$$

Sorgenti di Poisson in NS

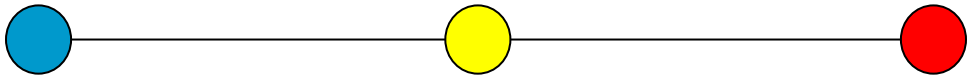
- E' possibile avere le sorgenti di Poisson in NS con un "espediente"
  - sorgenti ExpOnOff con tempo di ***On nullo***
  - tempo di Off (***idle_time_***) = E[Y]

```
set exp0 [new Application/Traffic/Exponential]
$exp0 set packet_size_   1000
$exp0 set burst_time_    0s
$exp0 set idle_time_     0.2s
$exp0 set rate_          100000M
```

  - in *nscript* è stato implementato tale espediente ed è possibile usare direttamente le sorgenti di Poisson

Esercizio 4a: ritardo di attesa in coda con traffico di Poisson

- Si consideri una rete con tre nodi:



- si attacchi una ***sorgente di Poisson*** al nodo di sinistra e un ***LossMonitor*** al nodo di destra e li si connetta
  - packet size L = 1000 (bytes)
  - λ = 5 (pacchetti/s)
- (a) si calcoli il ritardo medio dei pacchetti nella coda del nodo giallo
  - lunghezza buffer 1000
  - tempo di simulazione 20 s

Calcolare il ritardo medio in una coda usando *nscript*

- 1. Inserire un ***QueueMonitor***
  - From Node, To Node: inserire gli estremi del link lungo cui calcolare il ritardo
  - DelayStats: On
  - DelaySamplesName: <nome_var: eg. delay0>
- 2. Inserire un ***ResultPrinter***
  - File: <nome_file>
  - Time: tempo al quale scrivere i risultati (prima della fine della simulazione!!!)
  - Results Header: <stringa: eg. “Traffico di Poisson”>
  - Result 1 Name: <stringa: eg. “Ritardo:”>
  - Result 1 Object: <nome_var: eg. delay0>
  - Result 1 Variable: mean

Esercizio 4b

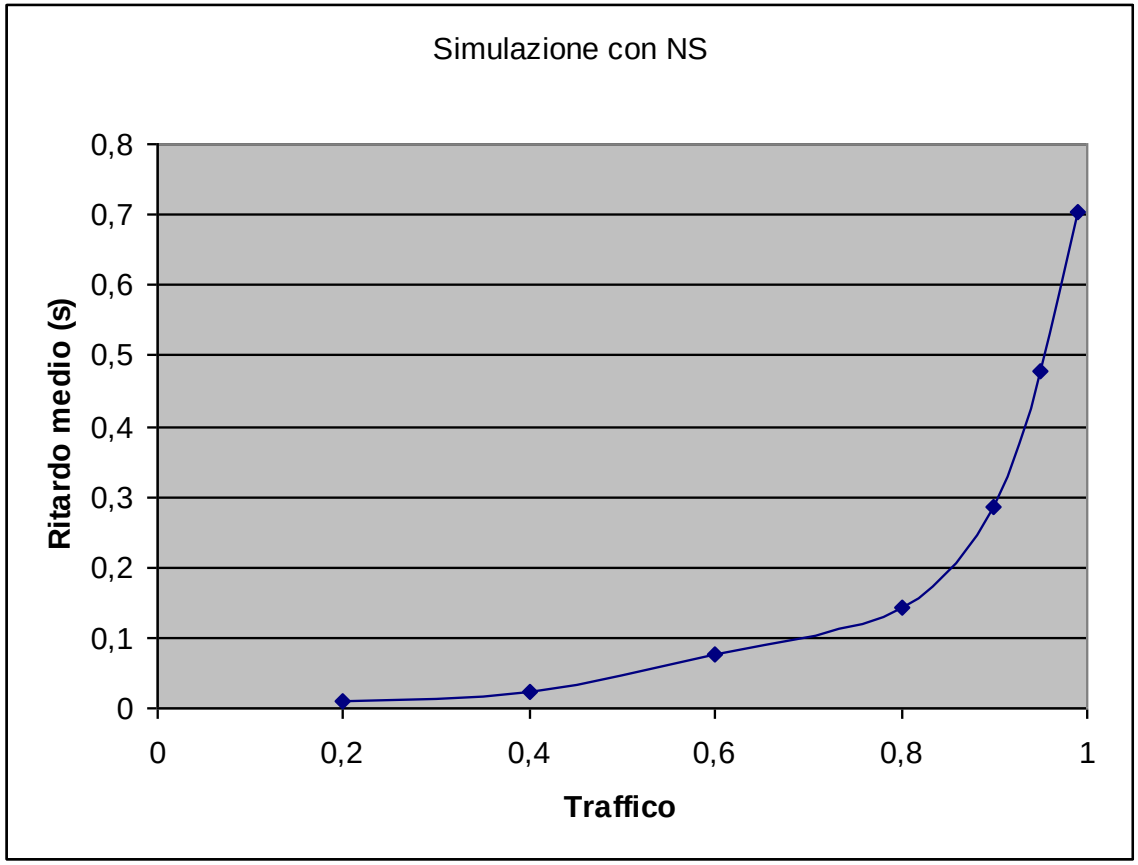
- (b) si tracci la curva traffico-ritardo usando i seguenti valori di traffico:
  - A = 0.2
  - A = 0.4
  - A = 0.6
  - A = 0.8
  - A = 0.9
  - A = 0.95
  - A = 0.99

Esercizio 4b

- Nota: nel nostro esempio μ vale 80000/(8*1000)= 10 pacchetti/secondo
- Quindi, per ottenere
  - A = 0.2 è necessario imporre λ = 2 (ovvero inter-arrivo pari a 1/λ = 0.5)
  - A = 0.4 …
  - A = 0.6 …
  - A = 0.8 …
  - A = 0.9 …
  - A = 0.95 …
  - A = 0.99 …

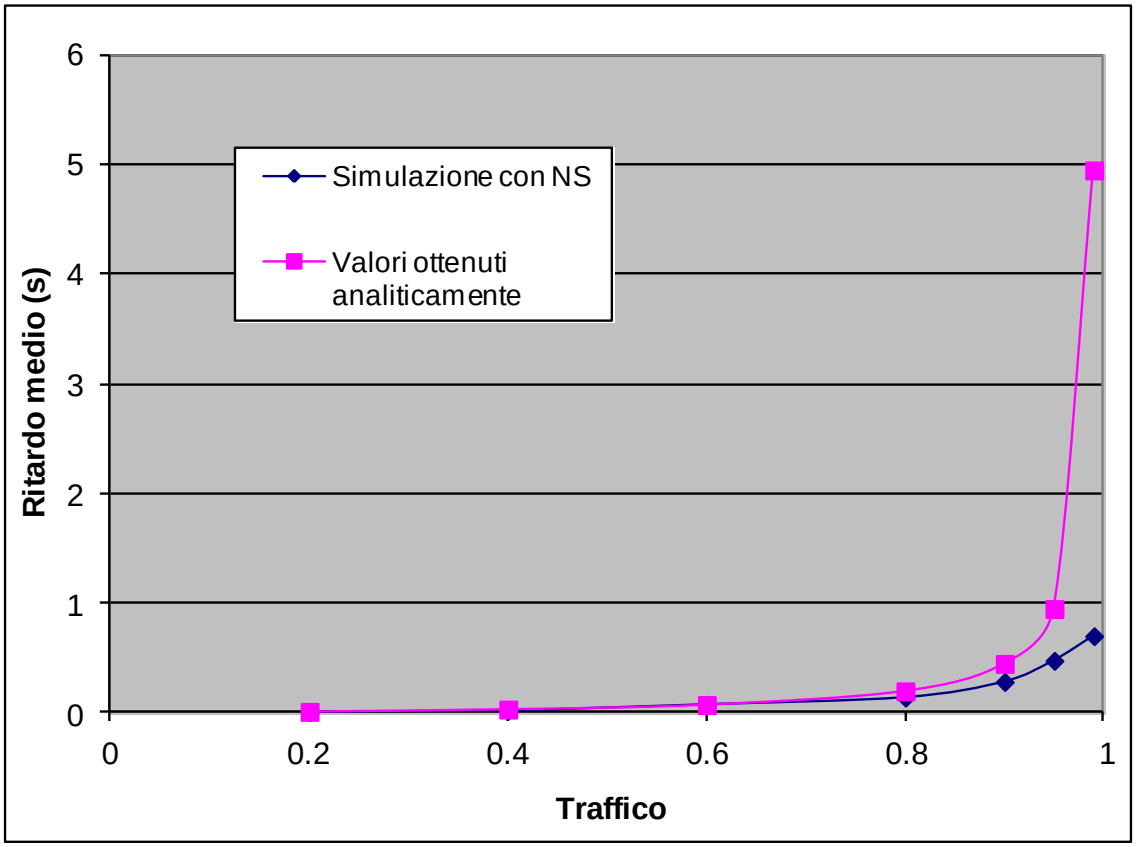
Esercizio 4b

- Risultato:



Esercizio 4b

- Confronto con la teoria delle code:

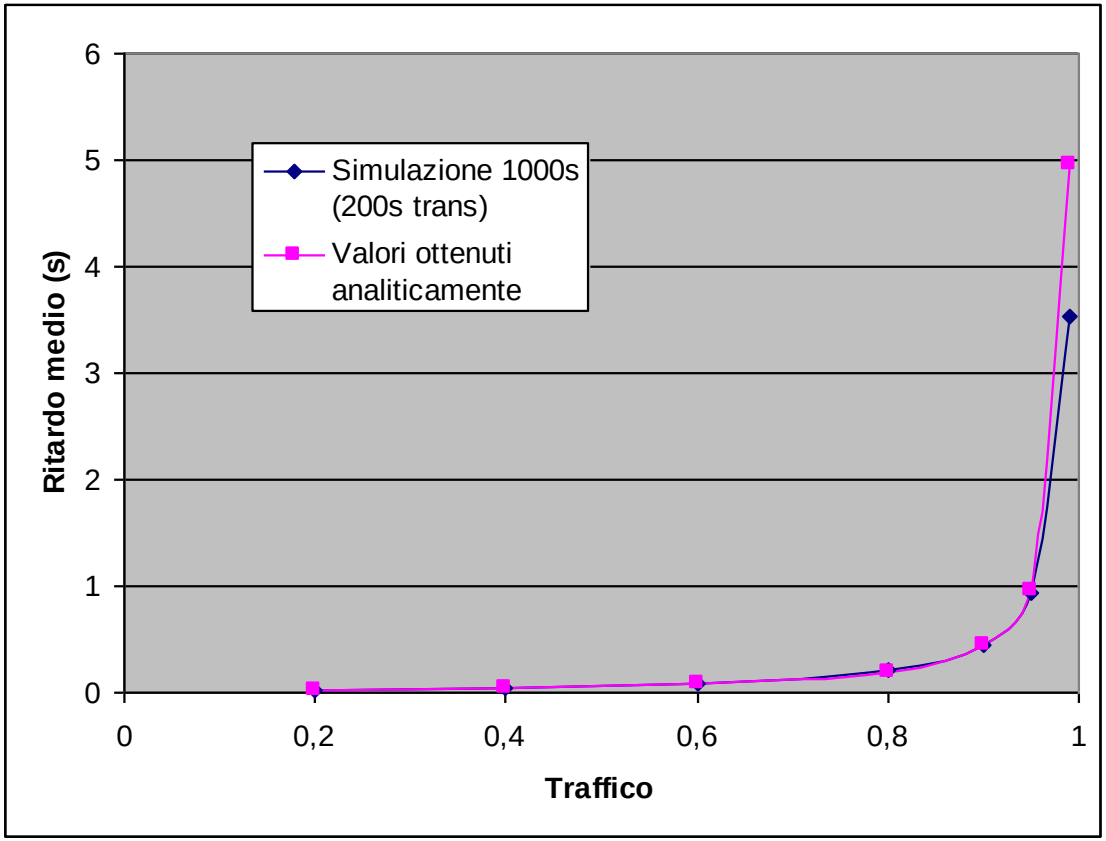


Esercizio 4c

- (c) si tracci di nuovo la curva con una lunghezza di simulazione di 1000 s. Dopo 200 s di simulazione si resetti il contatore statistico dei ritardi

Esercizio 4c

- Risultato:

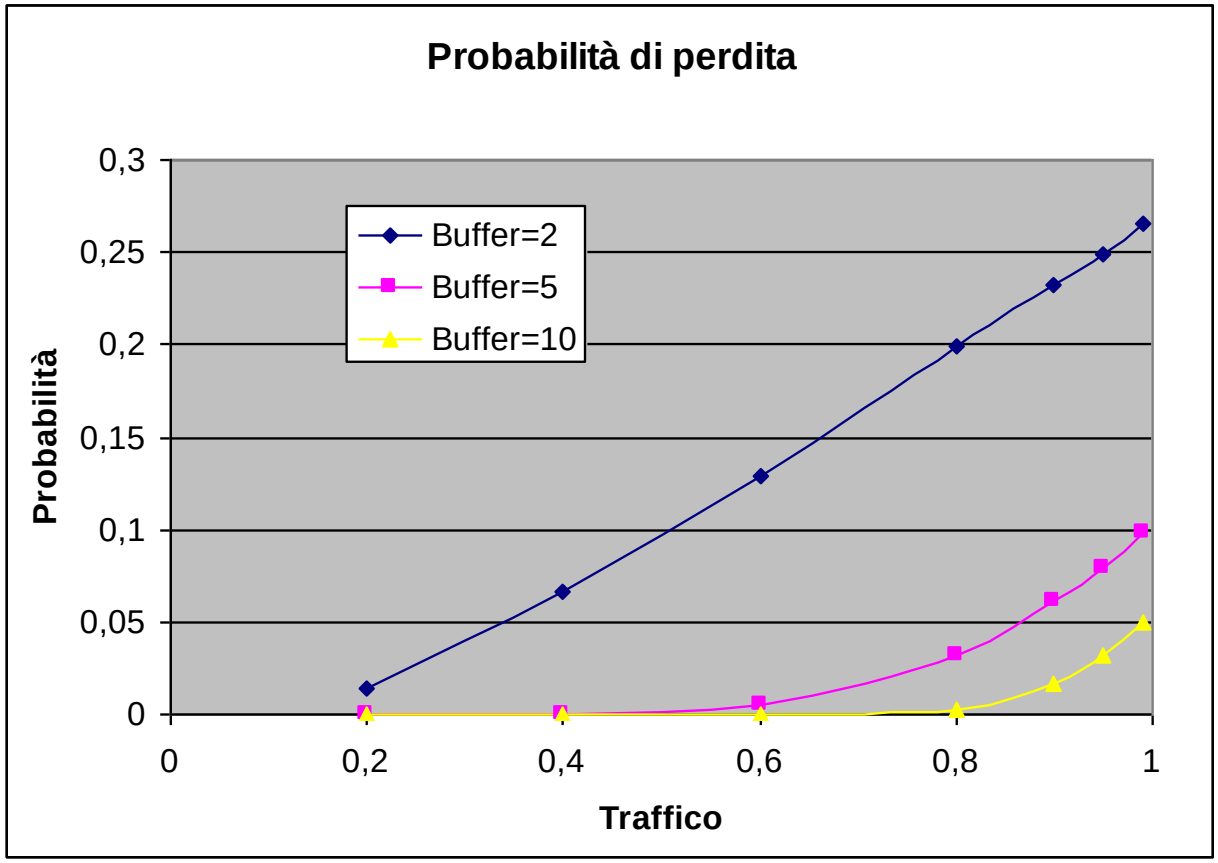


Esercizio 5: probabilità di trabocco del buffer con traffico di Poisson

- Si consideri lo scenario simulativo dell'esercizio precedente
- Si valuti la probabilità di trabocco del buffer per i valori di traffico già usati e per lunghezze del buffer pari a
  - B=2
  - B=5
  - B=10

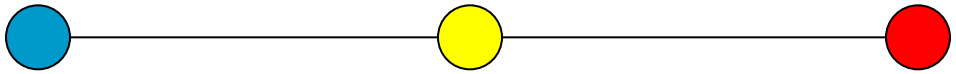
Esercizio 5: probabilità di trabocco del buffer con traffico di Poisson

- Risultato:



Esercizio 6: Sorgenti ON/OFF

- Si consideri la rete degli esercizi precedenti



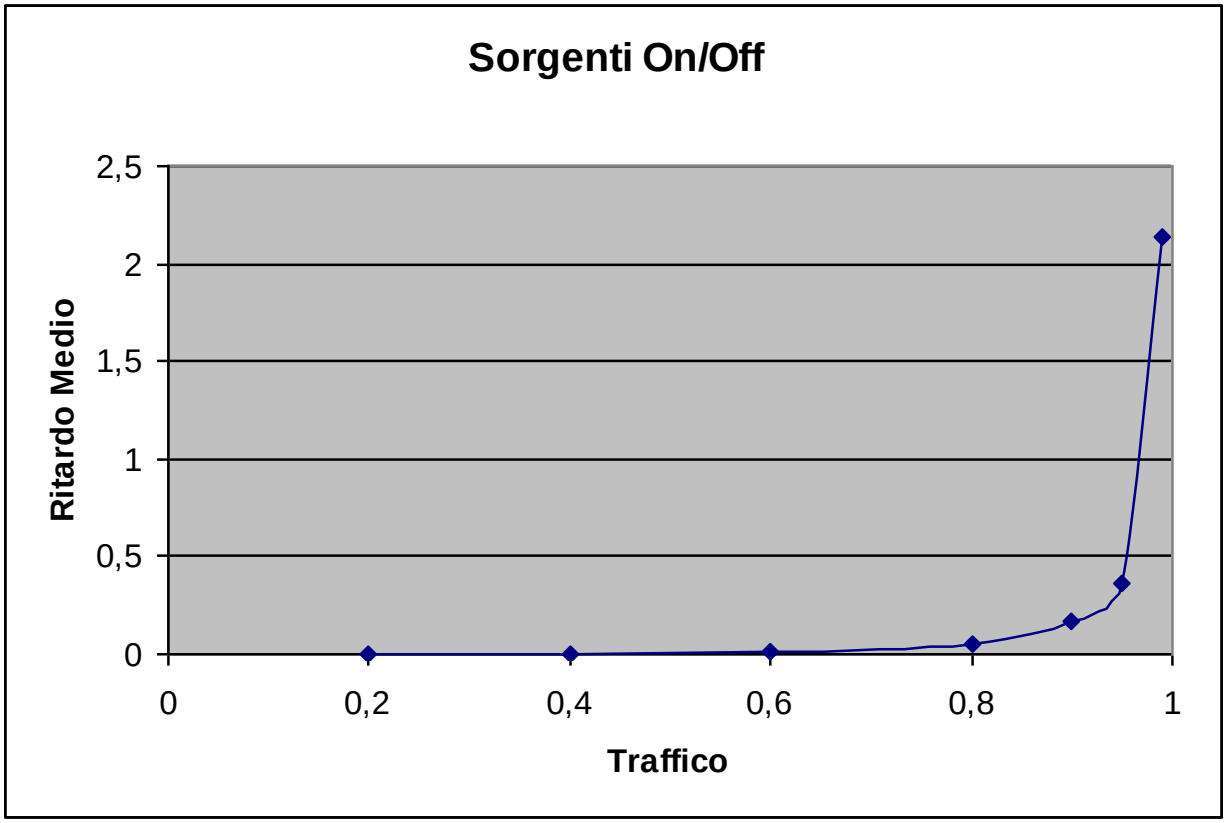
C = 10 Mb/s C = 100 kb/s

τ = 20 ms τ = 20 ms

- si attacchino al nodo di sinistra 20 sorgenti ***ExpOnOff***
- si vari il traffico facendo variare il tempo di Off di ciascuna sorgente
  - Packet Size dell'agente UDP 1000 bytes
  - On time 1s
  - Rate 10 kb/s
- (a) si calcoli il ritardo medio in coda (link nodo-giallo nodo rosso) al variare del traffico A

Esercizio 6a

- Risultato:



Esercizio 6b

- Si ripeta l'esperimento con un tempo di On delle sorgenti pari a 10s
- Come saranno le sorgenti? Più "buone" o più "cattive"?

Esercizio 6b

- Risultato: sono più “cattive”