



- **Politecnico di Milano**
- **Dipartimento di Elettronica e**
- **Informazione**

**- 4 -**

## **La moltiplicazione statistica nelle reti a pacchetto**

**Laboratorio di Reti di Telecomunicazione**

# Premessa

- Useremo NS e *nscrip*t per studiare e quantificare alcuni aspetti delle reti di telecomunicazione.

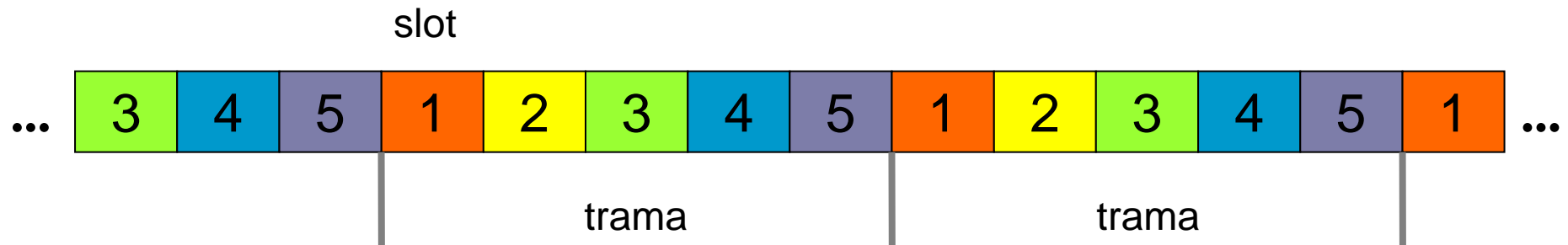
# La moltiplicazione

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



# La moltiplicazione nelle reti a circuito

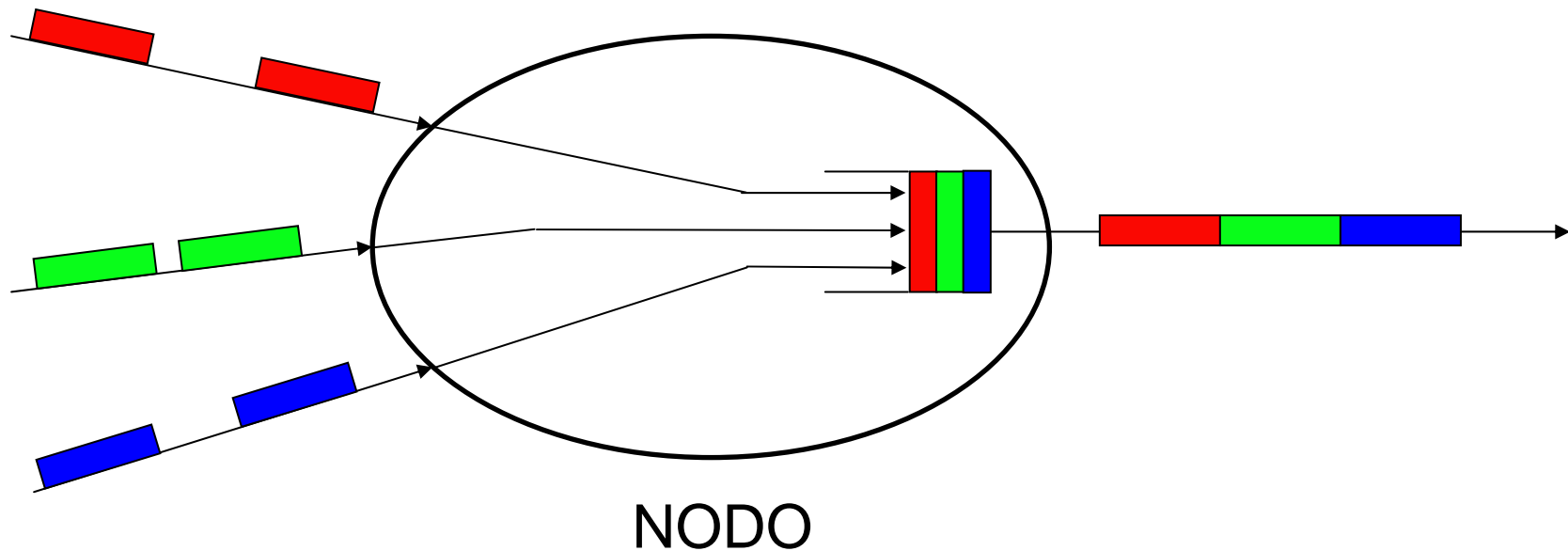
- Nelle reti a circuito (es: rete telefonica) la moltiplicazione tra flussi avviene mediante moltiplicazione statica come la tecnica TDM (Time Division Multiplexing)



- 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;

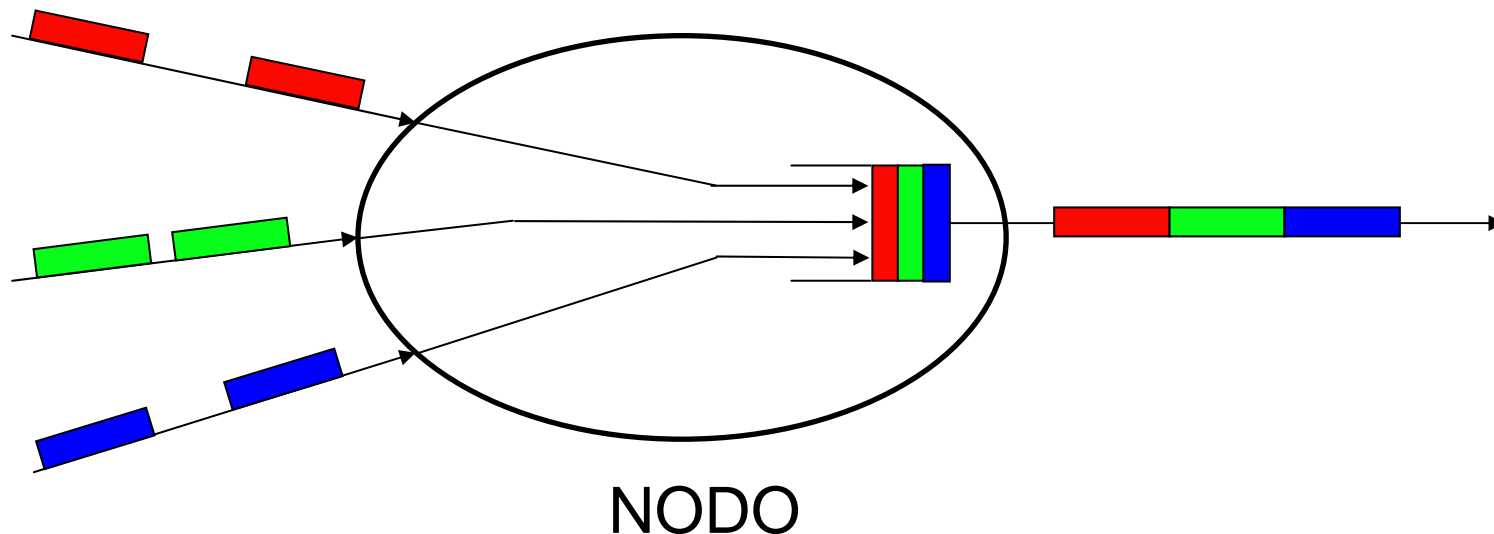
# La multiplazione nelle reti a pacchetto

- 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);
- un pacchetto che arriva in un nodo viene memorizzato, analizzato e trasferito verso il link d'uscita.



# 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;
- le trasmissioni dei diversi flussi sul canale si alternano in base all'ordine di uscita dalla coda in modo casuale;
- 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;
- non riserva risorse per ciascun flusso come nel caso del circuito;
- si ha un miglior utilizzo delle risorse con sorgenti di traffico discontinue (es: servizi dati);
- lo svantaggio 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.

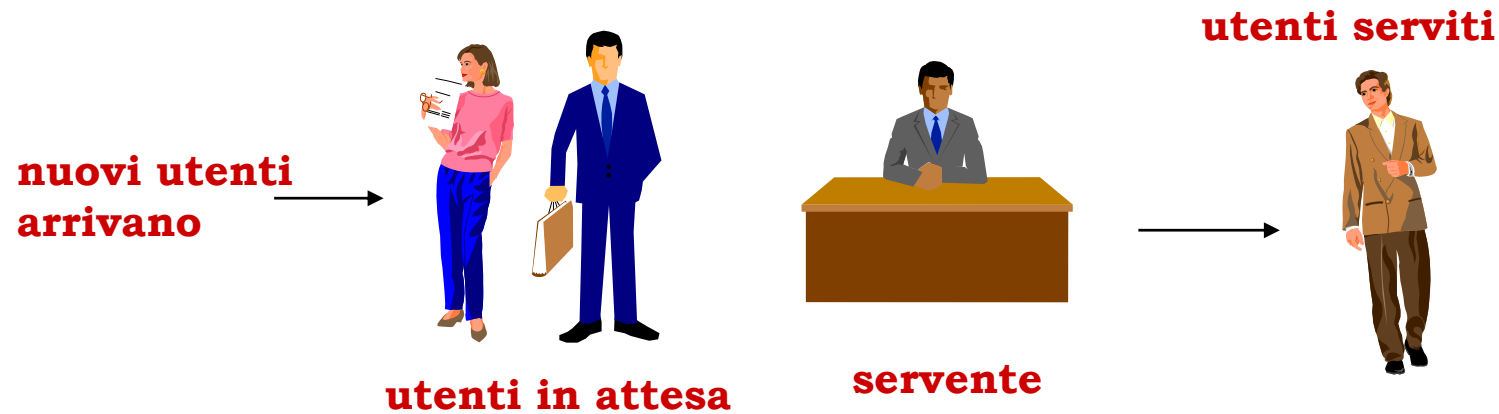


# La multiplazione nelle reti a pacchetto

- Per studiare in modo quantitativo il ritardo di trasferimento (in particolare di accodamento) ci sono due mezzi:
  - *teoria delle code* (metodi analitici - corso di “Modelli di Reti di Telecomunicazione” - 3 anno laurea spec.);
  - simulazione (è il nostro !!!);
- i modelli di sistemi su cui si basano la teoria delle code e la simulazione sono i:

***Sistemi d'attesa***

# Sistemi d'attesa

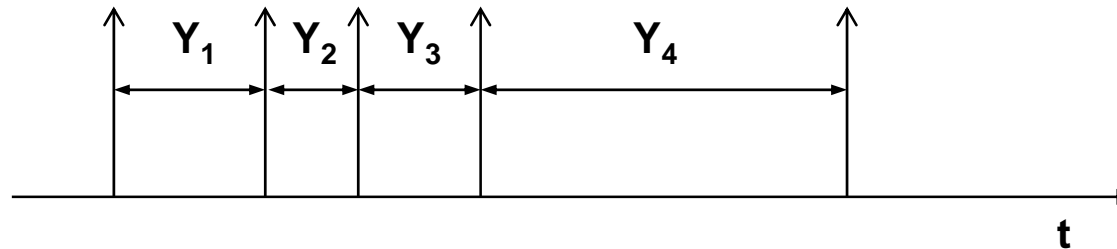


# Sistemi d'attesa

- 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



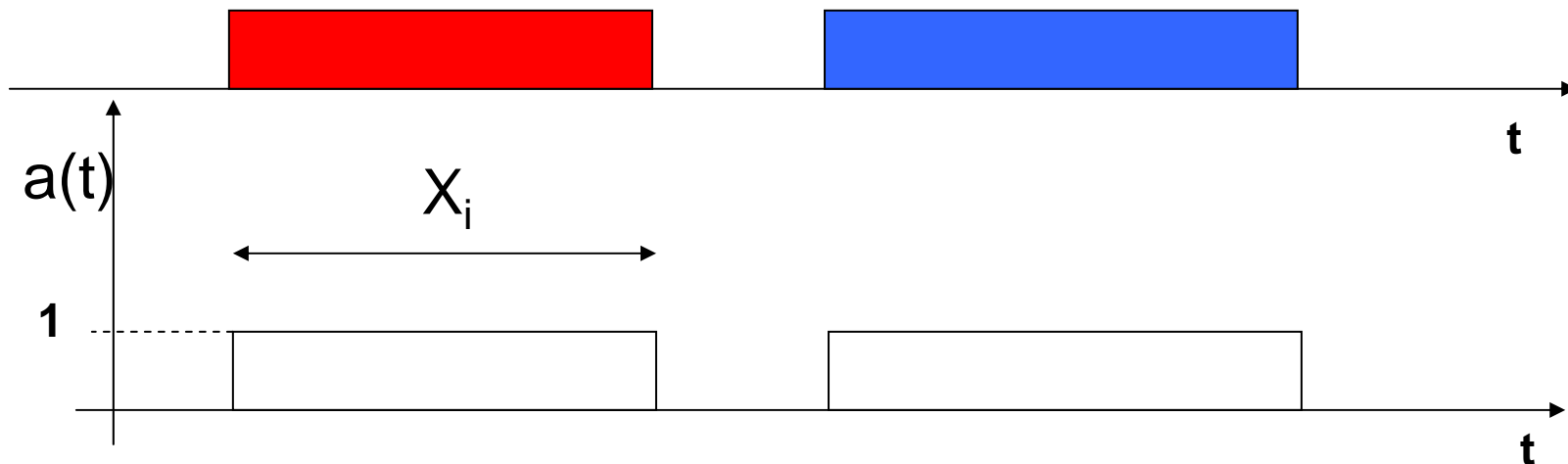
- E' il processo che descrive gli istanti d'arrivo
  - si può usare:
    - $N(0,t)$  numero di arrivi in  $(0,t)$ ;
    - $\{Y_k\}$  sequenza dei tempi di interarrivo;
  - parametri caratteristici:
    - numero medio di arrivi nell'unità di tempo  $\lambda$  (pacchetti/secondo).

# Sistemi d'attesa: processo dei servizi

- È 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;
    - $\mu = 1/E[X]$  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  $a(t)$  trasmessi 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:**

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

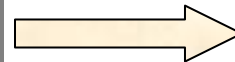
$$\frac{1}{T} \int_T a(t) dt = \frac{n}{T} \frac{\sum_i X_i}{n}$$



$n$  = numero di arrivi in T



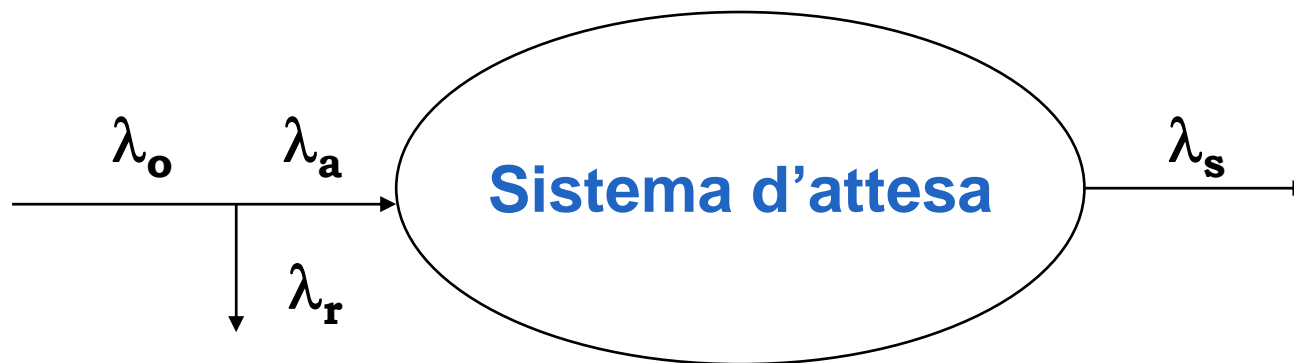
$$\lambda = \frac{n}{T} \quad \frac{\sum_i X_i}{n} \rightarrow E[X] = \frac{1}{\mu}$$



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

# Sistemi d'attesa: traffico

- Se un pacchetto arriva e non trova posto viene scartato (overflow del buffer);
- possiamo definire:
  - $\lambda_o$  frequenza di pacchetti offerti;
  - $\lambda_a$  frequenza di pacchetti accettati;
  - $\lambda_r$  frequenza di pacchetti persi;
  - $\lambda_s$  frequenza di pacchetti smaltiti ( $\lambda_a = \lambda_s$ ).
- Definizioni analoghe valgono per il traffico





# Sorgenti di traffico

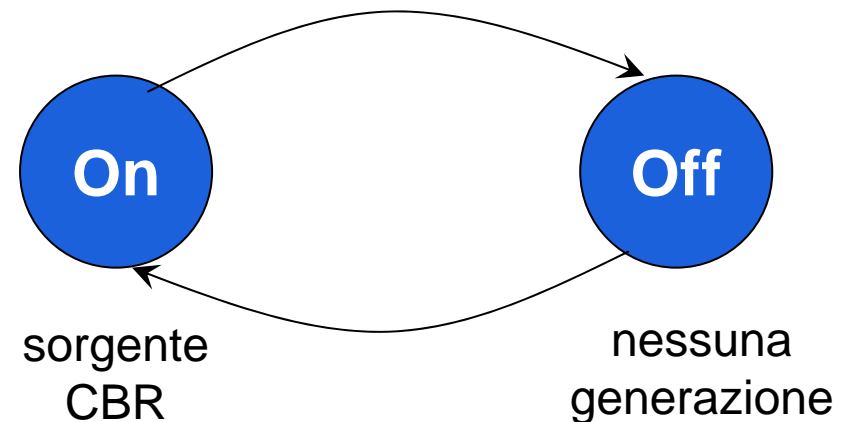
Abbiamo visto alcune semplici sorgenti di traffico in NS

## ■ 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

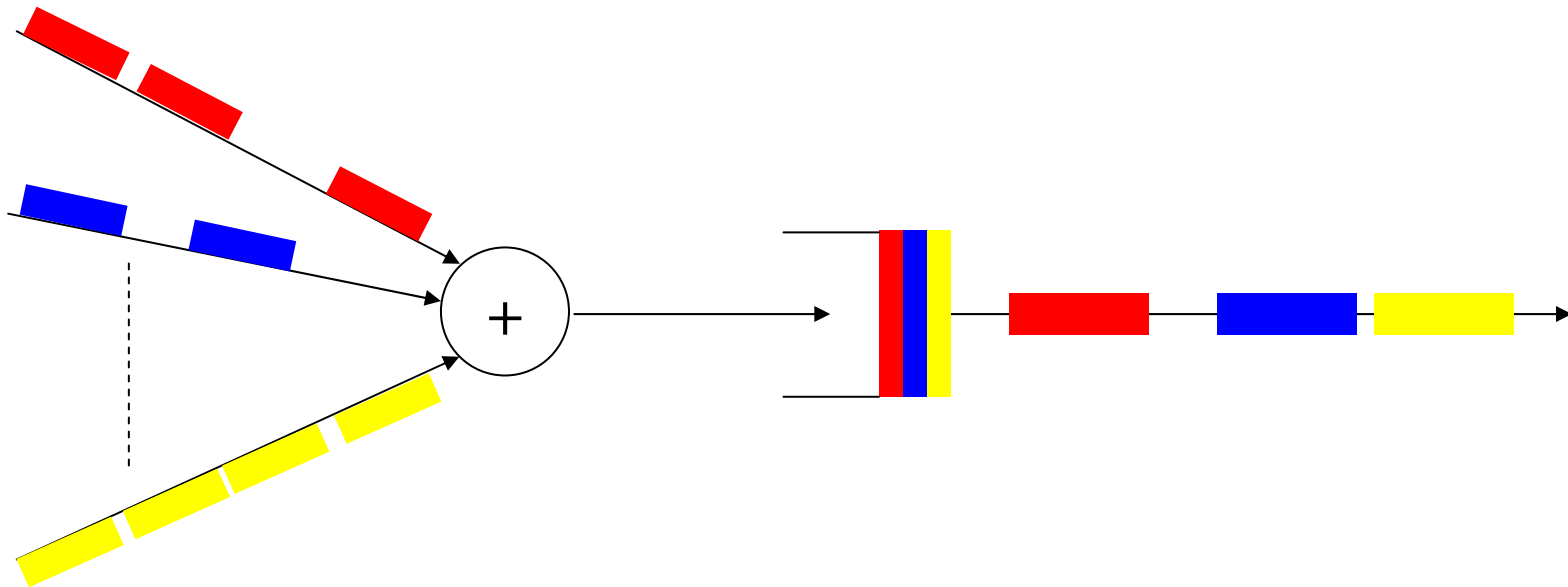
## ■ Sorgente ExpOnOff



**Se più sorgenti confluiscono in un link le frequenze d'arrivo si sommano**

# Sorgenti di traffico

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



# Eventi di Poisson

- Il processo degli eventi di Poisson è descritto (probabilisticamente) da 4 assiomi:
  - 1) La probabilità che ci sia *esattamente* un evento di Poisson in un intervallo  $\Delta t$  è pari a:

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

$\lambda$  = frequenza del processo (in eventi per unità di tempo).

- 2) La probabilità che ci sia più di un evento di Poisson in un intervallo di tempo  $\Delta t$  è pari a  $o(\Delta t)$
- 3) Il numero di eventi di Poisson che si verificano in intervalli di tempo disgiunti sono tra loro indipendenti
- 4) Il numero di eventi di Poisson che si verificano in un certo intervallo di tempo dipende solo dall'ampiezza dell'intervallo.

# Arrivi di Poisson

- Si può far vedere che:
  - La probabilità che il numero di arrivi (punti) di Poisson  $N(t, t+\tau)$  in un intervallo temporale fra  $t$  e  $t+\tau$  sia pari a  $k$  è:

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

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

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

- sotto le ipotesi 3) e 4), si dimostra che un processo con interarrivi esponenziali indipendenti è di Poisson

# Sorgenti di Poisson in NS

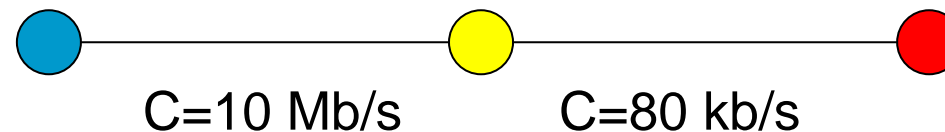
- È possibile avere sorgenti di Poisson in NS con un espediente:
  - sorgenti ExpOnOff con tempo medio di On nullo;
  - tempo medio di Off =  $E[Y]$ ;
  - rate di picco elevatissimo

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

- in *nscrip*t è stato implementato l'espediente ed è possibile usare direttamente le sorgenti di Poisson.

## Es. 4: 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  $L = 1000$  byte;*
  - *$\lambda = 5$  (pacchetti/s).*
- a) Si calcoli il ritardo medio dei pacchetti in coda al nodo centrale (lunghezza buffer associato al link di destra = 1000pacchetti); tempo di simulazione 20 s.

# Esercizio 4b

**b)** Mantenendo invariato  $\mu$  si tracci la curva traffico-ritardo usando i seguenti valori di traffico:

- $A = 0.2;$
- $A = 0.4;$
- $A = 0.6;$
- $A = 0.8;$
- $A = 0.9;$
- $A = 0.99.$

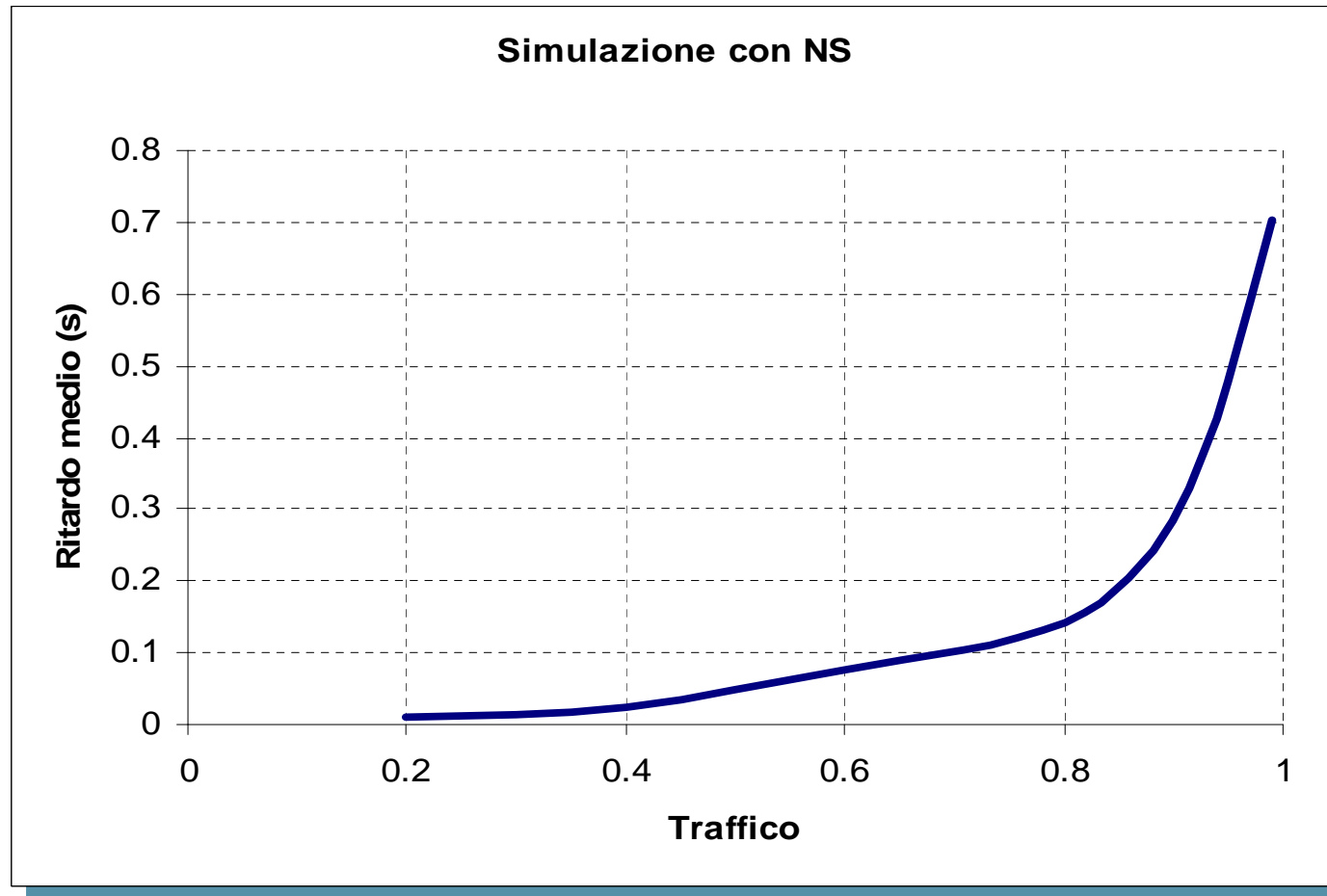
■ **Suggerimenti operativi:**

- *Ottenere i ritardi di accodamento utilizzando i monitor-queue; ad esempio:*
  - `set QM0 [$ns monitor-queue $MiddleNode $DestNode [$ns get-ns-traceall] 0.1]`
  - `set delay0 [new Samples]`
  - `$QM0 set-delay-samples $delay0`
  - `$ns at 21.0s "finish"`
  - `... Nella procedura finish: puts "[$delay0 mean]"`
- *lanciare un ciclo di simulazioni che, dopo aver ricalcolato  $\lambda$ , stampino in un file le coppie traffico-ritardo;*
- *a partire da un file formato da due colonne di numeri, si può ottenere un grafico con il comando:*

`xgraph nomefile`

# Esercizio 4b

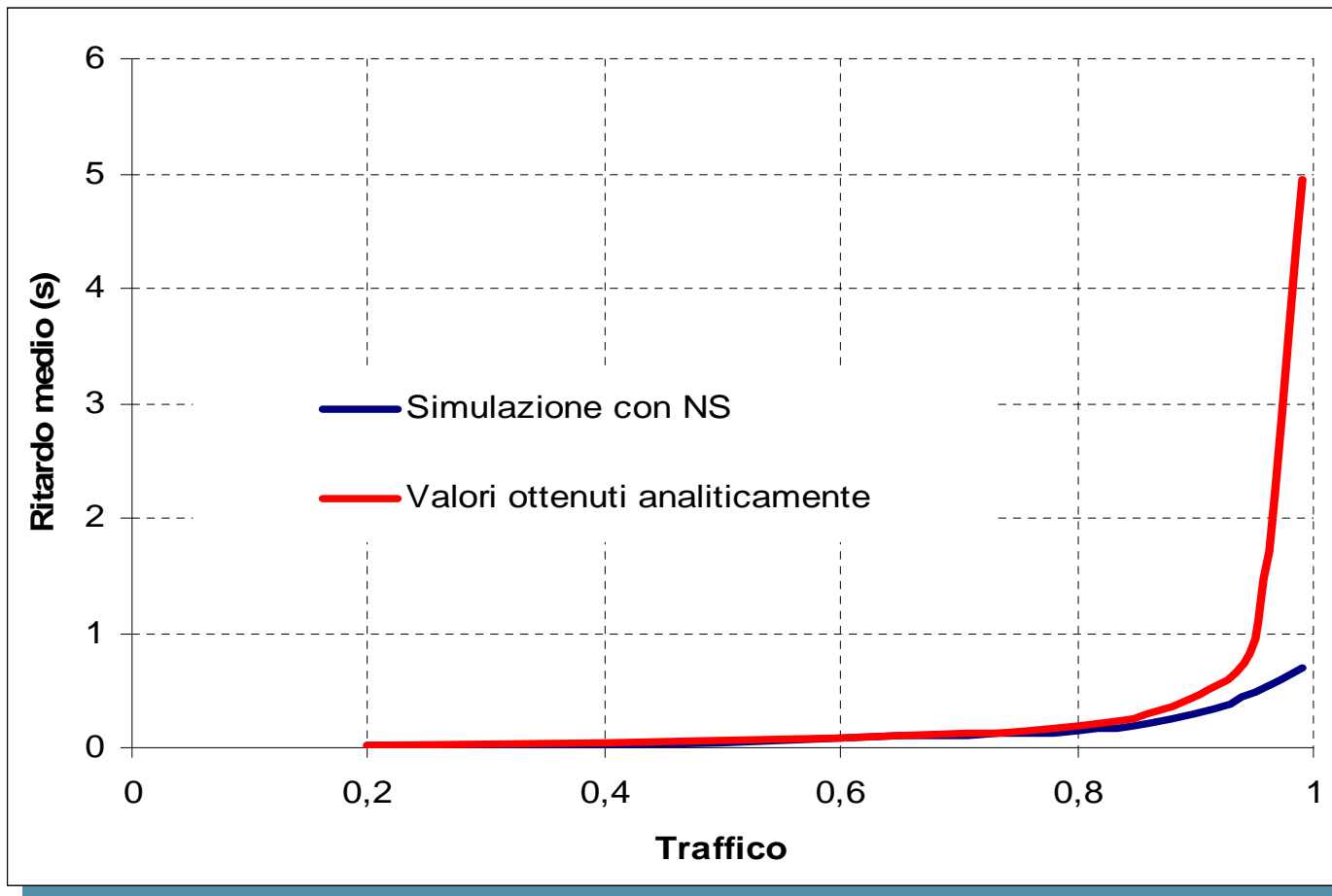
**Risultato:**





# Esercizio 4b

## Confronto con la teoria delle code:

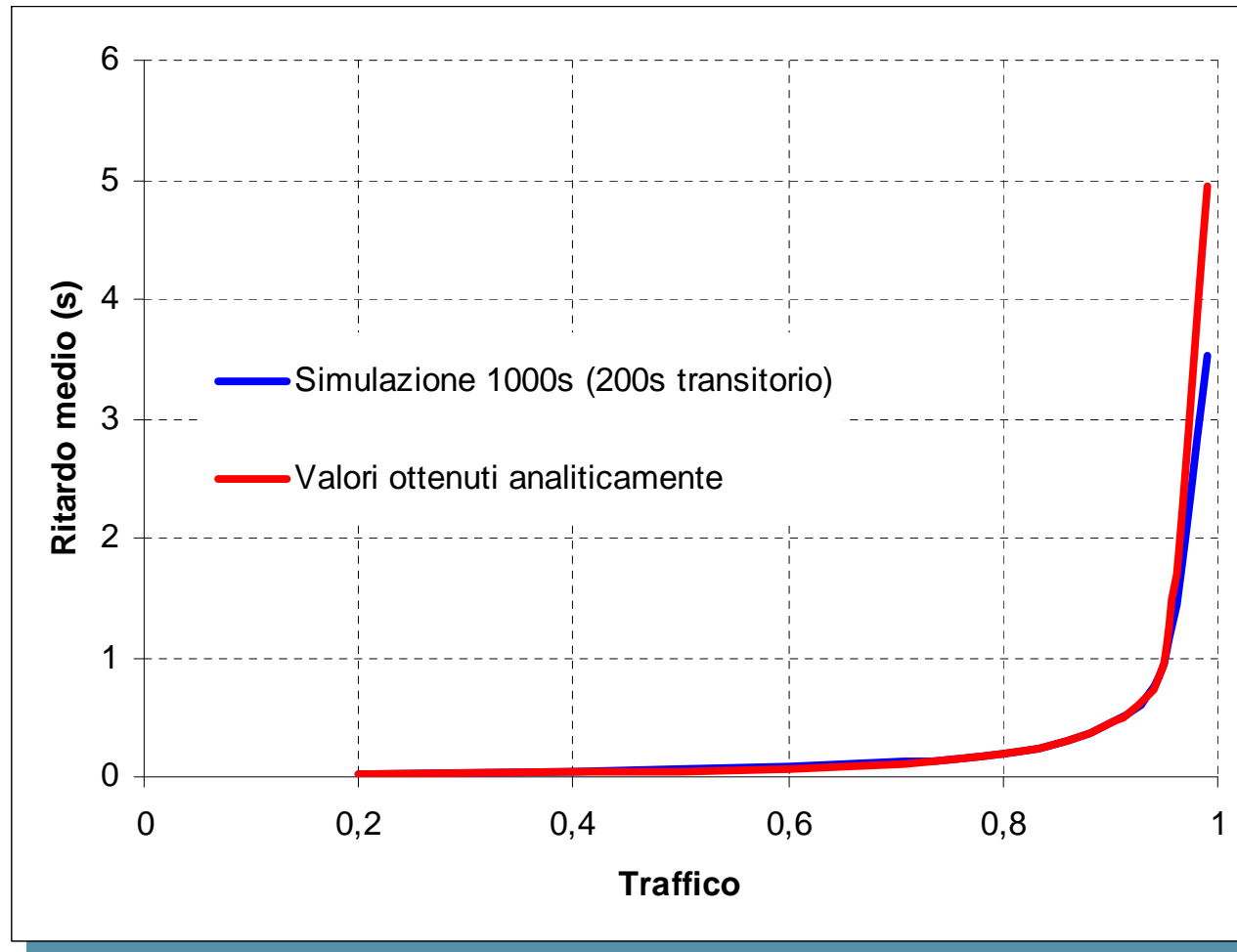


## Esercizio 4c

- 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:

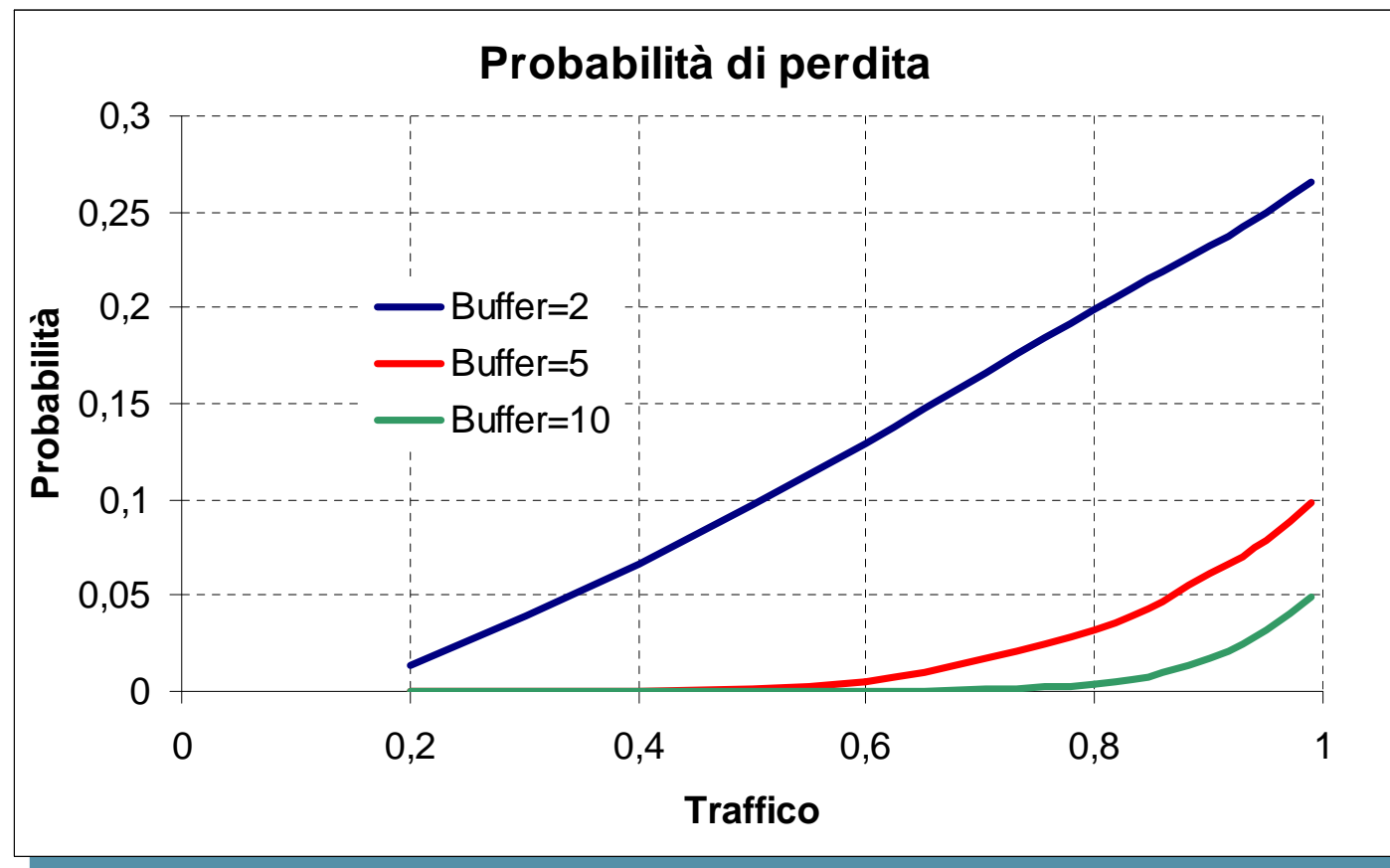


## Esercizio 5: probabilità di trabocco del buffer con traffico di Poisson

- Si consideri lo scenario dell'esercizio precedente;
- si valuti la probabilità di trabocco del buffer per i valori di traffico già usati e con lunghezze del buffer pari a:
  1.  $B=2$
  2.  $B=5$
  3.  $B=10$

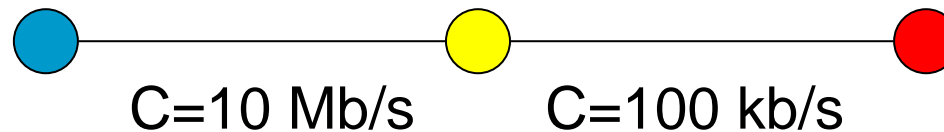
# Esercizio 5: probabilità di trabocco del buffer con traffico di Poisson

**Risultato:**



# Esercizio 6: Sorgenti ON/OFF

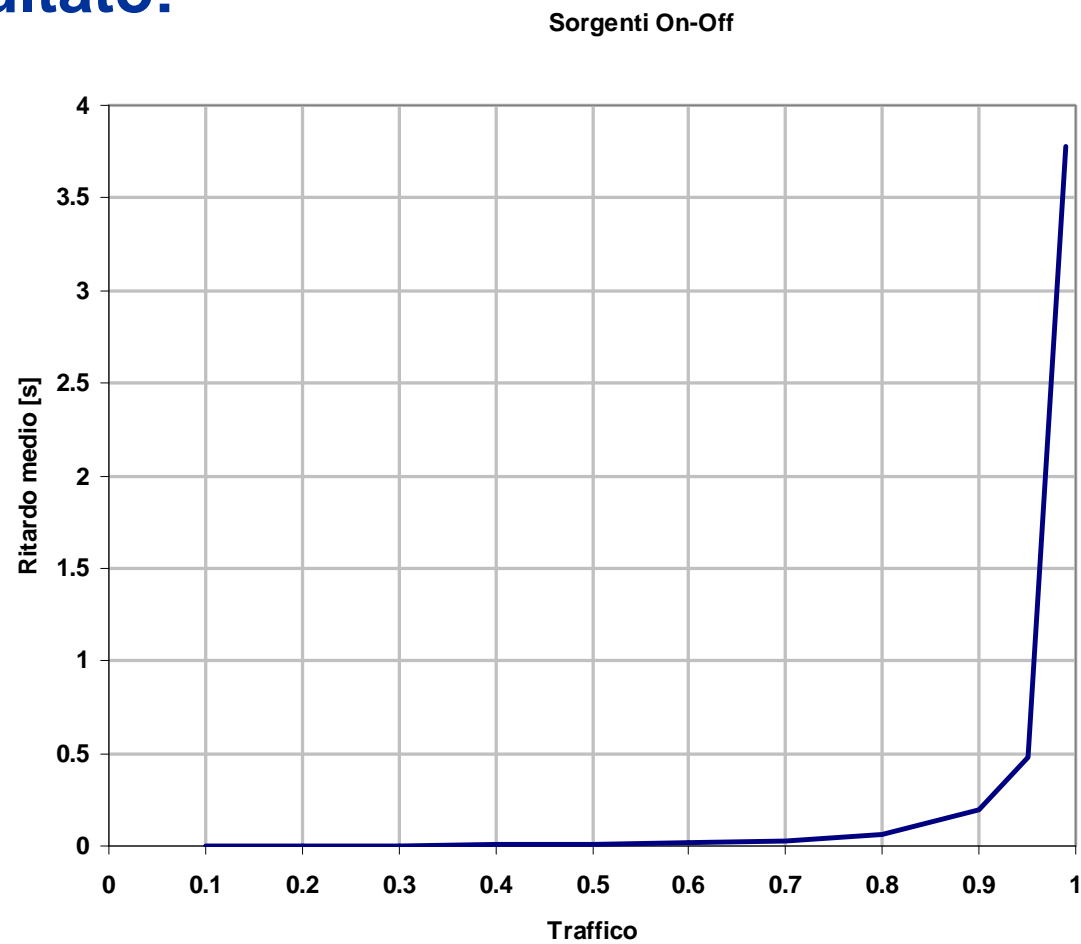
- Si consideri la rete seguente:



- si attacchino al nodo di sinistra 20 sorgenti ExpOnOff;
- si vari il traffico facendo variare il tempo di Off di ciascuna sorgente:
  - packet size 210 byte;
  - On time 1s;
  - rate 10 kb/s.
- a) Si calcoli il ritardo medio in coda (link nodo-giallo nodo rosso) al variare del traffico (stessi valori es. 4).

# 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*

Sorgenti On-Off

