Mobile Communications

Performance Evaluation

Manuel P. Ricardo

Faculdade de Engenharia da Universidade do Porto

Techniques Available for Performance Evaluation

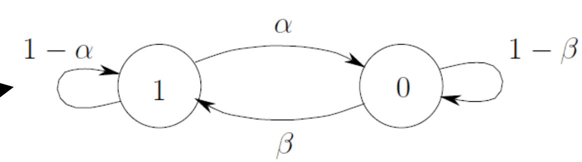
- Analytical Stochastic Models
- Deterministic analysis
- Bottleneck analysis
- Simulation

Techniques Available for Performance Evaluation

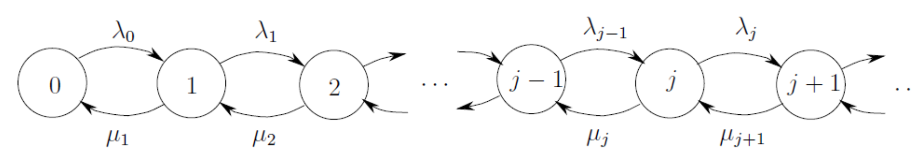
- Analytical Stochastic Models
- Deterministic analysis
- Bottleneck analysis
- Simulation

Analytical Models

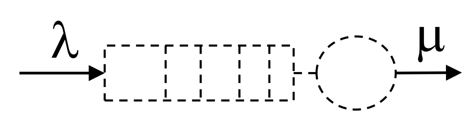
- Stochastic process X(ω,t) , Poisson process $P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \equiv P[N(t) = n]$
- Markov Chains
  - » Discrete-Time Markov Chain



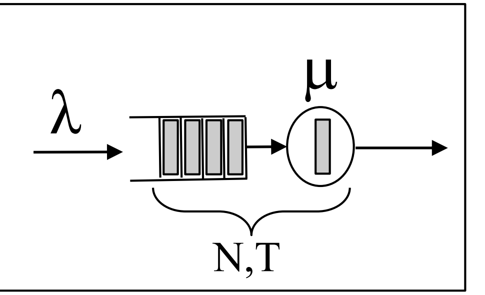
  - » Continuous-Time Markov Chain
  - » Birth-Death Chains



- Queueing Theory
  - » M/G/1, M/M/1, M/M/1/K, M/D/1, M/M/s, M/G/1/PS
  - » Open Queueing, Closed Queueing Networks



M/M/1 Queue



- Average Queue size N

$$N=\sum_{n=0}^{\infty} nP(n)=\sum_{n=0}^{\infty} n\rho^{n}(1-\rho)=\frac{\rho}{1-\rho} \qquad N=\sum_{n=0}^{\infty} nP(n)=\frac{\rho}{1-\rho}=\frac{\lambda/\mu}{1-\lambda/\mu}=\frac{\lambda}{\mu-\lambda}$$

- Average amount of time the client spends in the system, T
  - » Little's formula, T=N/λ ➔ $T=\frac{1}{\mu-\lambda}$

- Average waiting time T_w ➔ $T_w=T-T_s=\frac{1}{\mu-\lambda}-\frac{1}{\mu}=\frac{\rho}{\mu(1-\rho)}$

- Average number of clients waiting in the queue, N_w

$$N_w=T_w\lambda=\frac{\lambda}{\mu-\lambda}-\frac{\lambda}{\mu}=N-\rho$$

M/M/1/B Queue

- M/M/1 queue has limited capacity (B buffers)
  - » Packets can be lost
  - » Probability of packet being lost = P(B) ➔ Queue is full
- Analysis similar to M/M/1

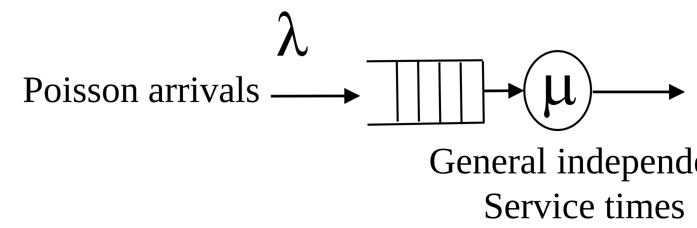
$$\sum_{i=0}^{B} P(i) = 1 \qquad P(n) = \rho^n P(0)$$

$$P(0) = \frac{1-\rho}{1-\rho^{B+1}} \qquad P(B) = \frac{(1-\rho)\rho^B}{1-\rho^{B+1}}$$

- Particular cases

$$\rho = 1, \quad P(B) = \frac{1}{B+1} \qquad \rho >> 1, \quad P(B) \approx \frac{\rho - 1}{\rho} = \frac{\lambda - \mu}{\lambda}$$

M/G/1 Queue



- Poisson arrivals at rate λ
- Service time X has arbitrary distribution with given E[X] and E[X^2]
  - » Service times Independent and Identically Distributed (IID)
  - » Independent of arrival times
  - » E[service time] =E[X]= 1/μ
  - » Single Server queue

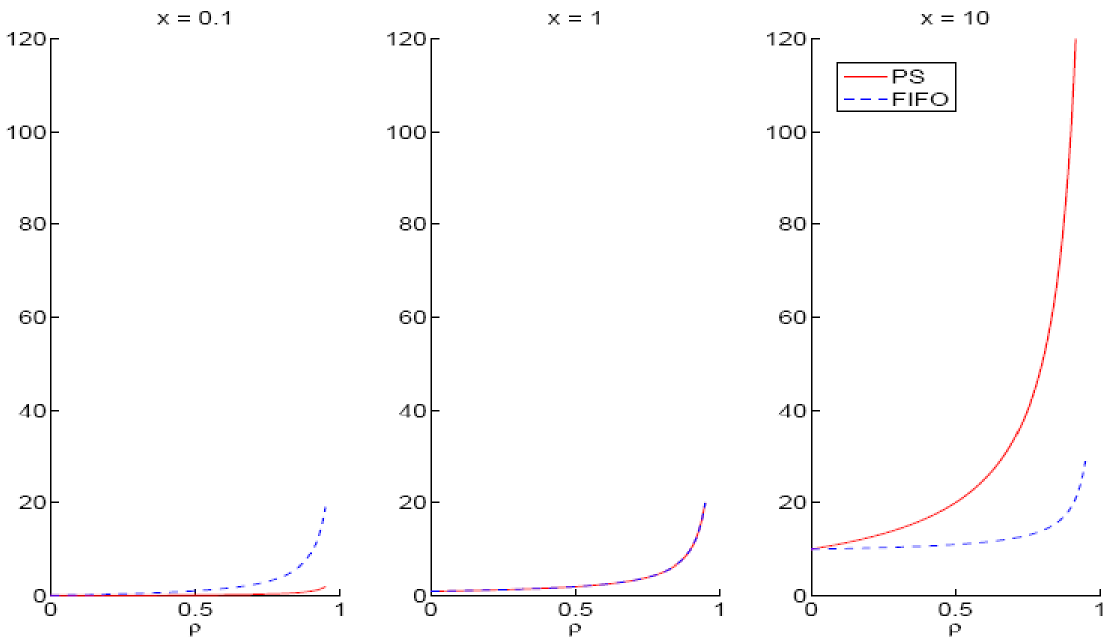
$$T_w = \frac{\lambda E[X^2]}{2(1-\rho)}$$

- where ρ = λ/μ = λE[X] = line utilization
- From Little's Theorem
  - » $N_w = \lambda T_w$
  - » $T = T_w + E[X] = T_w + 1/\mu$
  - » **$N = \lambda T$**$= \lambda(T_w + 1/\mu)$**$= N_w + \rho$**

The Processor Sharing Queue M/G/1/PS

- Non-FIFO queue
- Server is equally shared between all customers present

$$\mathrm{E}(\text{ response time }|\text{service time} = x) = \frac{x}{1-\rho}$$

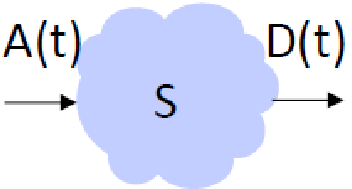


Techniques Available for Performance Evaluation

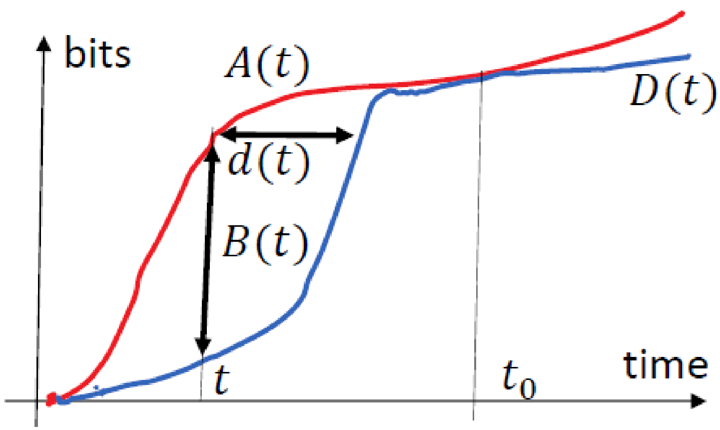
- Analytical Stochastic Models
- Deterministic analysis
- Bottleneck analysis
- Simulation

Deterministic Analysis - Cumulative Functions

- Cumulative functions
  - » *A(t)* : amount of bits input to system S in [0, t]
  - » *D(t)* : amount of bits output from system S in [0,t]



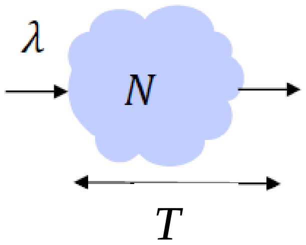
- Assuming no loss
  - » *B(t)* = buffer content at time
  - » *d(t)* = virtual delay



Techniques Available for Performance Evaluation

- Analytical Stochastic Models
- Deterministic analysis
- Bottleneck analysis
- Simulation

Little's Law



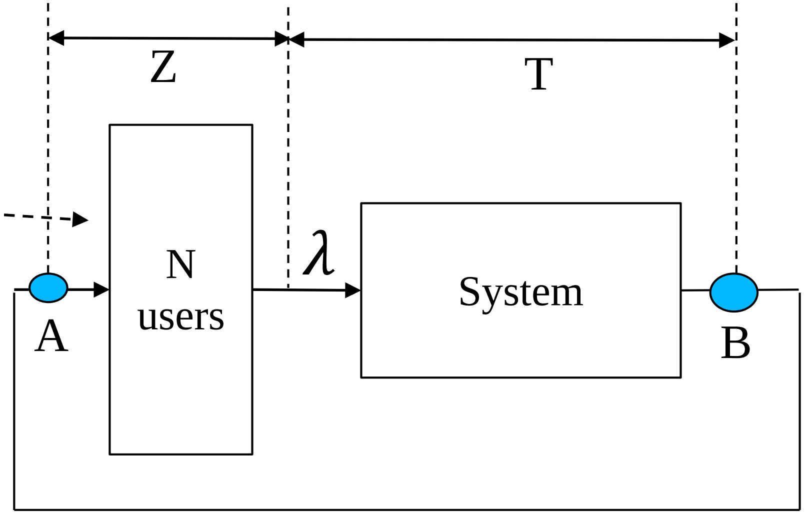
- Consider a discrete system
  - » λ = customer arrival rate [customer/s]
  - » T = average time spent in system by an arbitrary customer [s]
  - » N = average number of customers in system [customers]

- Little's law: $\lambda T = N$
  - » Applies to average values

- Note: customer = packet

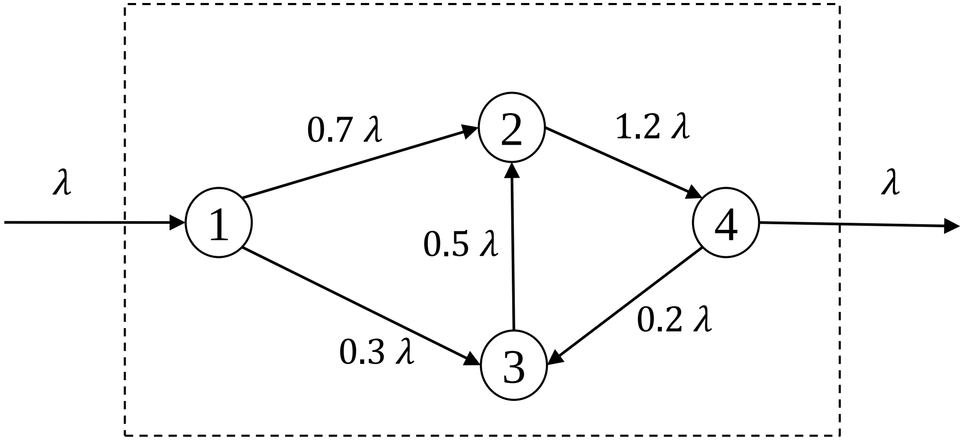
Interactive User Model - General

- N users alternate between
  <u>*think time*</u> and visit to the system



- Little's law applied to A → B
  - » $\lambda(\bar{Z} + \bar{T}) = N$
  - » Suppose we have N packets to continuously transmit through the system
  - » *Think time* forces a packet to wait for Z sec before reintroduced in the system
  - » Z is a random delay and $\bar{Z} = E[Z]$
  - » N is kept constant

Network Laws



- $\lambda_k = \lambda\ V_k$
  - » λ_k : expected number of customer arriving per second to node k
  - » V_k : expected number of visits to node k by an arbitrary customer
  - » $\lambda_1 = \lambda \rightarrow V_1 = 1$ | $\lambda_2 = 1.2\,\lambda \rightarrow V_2 = 1.2$
  - » $\lambda_3 = 0.5\,\lambda \rightarrow V_3 = 0.5$ | $\lambda_4 = 1.2\,\lambda \rightarrow V_4 = 1.2$

- $\bar{T} = \sum \overline{T_k} V_k$
  - » $\bar{T}$: expected total response time seen by a customer
  - » $\overline{T_k}$: expected response time seen by a customer which visits node k
  - » $\bar{T} = \overline{T_1}V_1 + \overline{T_2}V_2 + \overline{T_3}V_3 + \overline{T_3}V_4 = \overline{T_1} + 1.2\,\overline{T_2} + 0.5\,\overline{T_3} + 1.2\,\overline{T_4}$

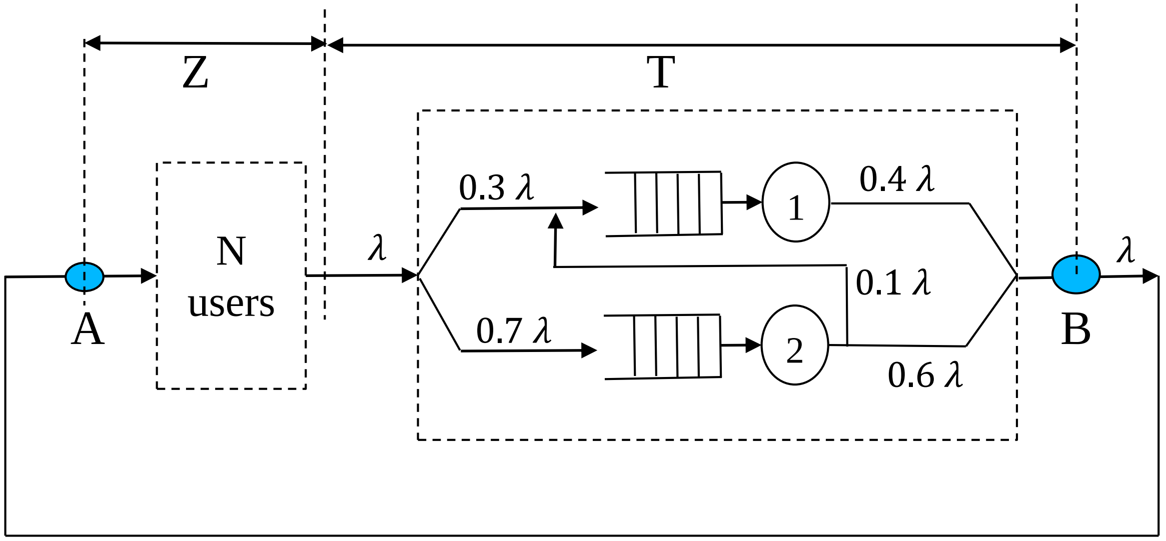
Bottleneck Analysis

- Powerful approach to analyze complex queuing systems
- Based on Little's Formula
- Considers two (extreme) scenarios:
  - » Very low utilization: no waiting time,

    $T_W=0 \rightarrow T = T_W+T_S = T_S$

  - » Very high utilization: $\rho = \frac{\lambda}{\mu} = \lambda . T_s \sim 1$ *(for a single server)*

Bottleneck Analysis



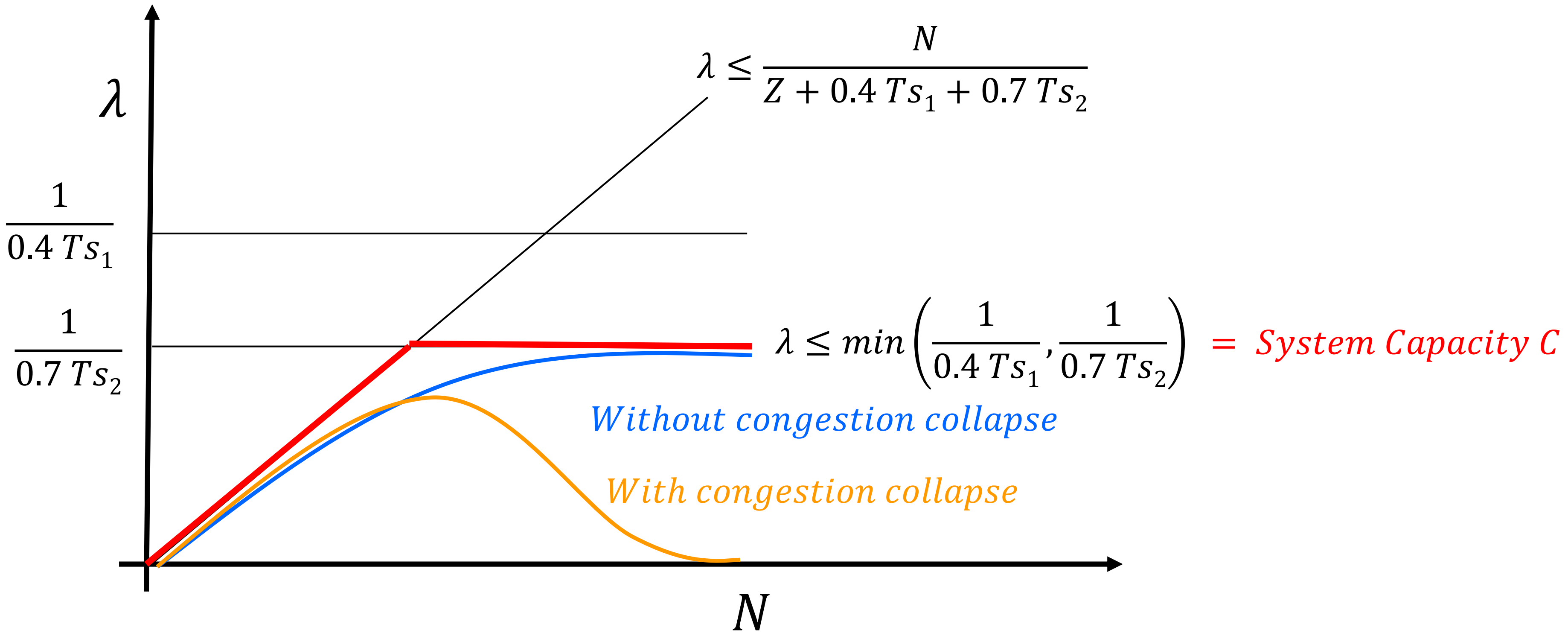
- $\lambda(Z + T) = \lambda(Z + 0.4\,T_1 + 0.7\,T_2) = N$

- Very low utilization, $Tw_i = 0 \quad \rightarrow \quad \lambda \leq \frac{N}{Z+0.4\,Ts_1+0.7\,Ts_2}$

- Very high utilization, $\lambda\,T_s \leq 1$,

  $0.4\,\lambda\,Ts_1 \leq 1$ and $0.7\,\lambda\,Ts_2 \leq 1 \quad \rightarrow \quad \lambda \leq min\left(\frac{1}{0.4.Ts_1}, \frac{1}{0.7.Ts_2}\right)$

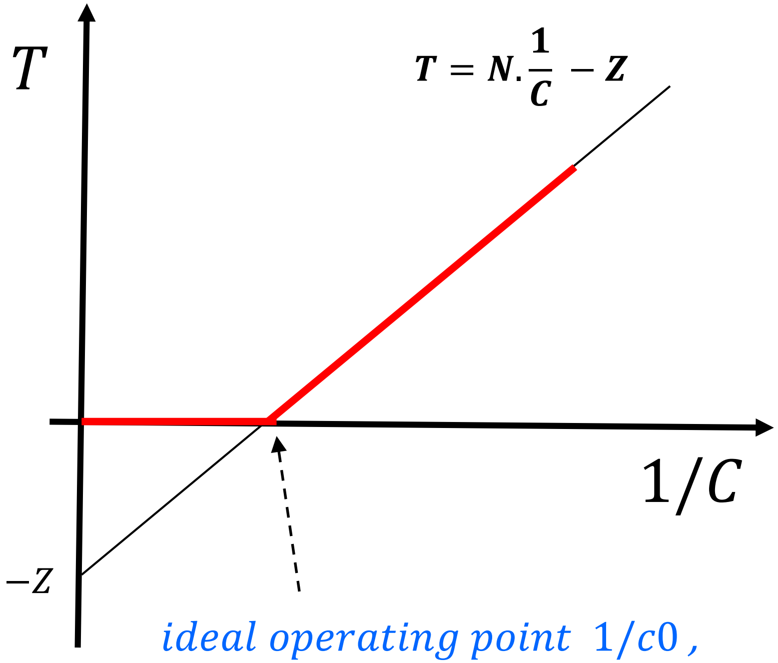
Bottleneck Analysis



- Real curve is either blue or orange

Delay vs Capacity

- Let us assume the system is operating near the system capacity
  - » $\lambda = C$
  - » Then, $\lambda(Z + T) = N \quad \rightarrow \quad C(Z + T) = N \quad \rightarrow \quad T = N.\frac{1}{C} - Z$



$\textit{ideal operating point } 1/c0\,,$
$\textit{the minimum capacity at which we experiment no relevant delay}$

Techniques Available for Performance Evaluation

- Analytical Stochastic Models
- Deterministic analysis
- Bottleneck analysis
- Simulation

Discrete Event Simulation

Introduction

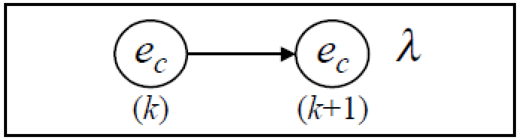
- Precise analytical solutions for complex systems: difficult to obtain
- Simulation
  - process which evaluates a system model numerically
  - estimates quantities of interest
- Discrete Event Simulation
  - Change of system state is caused by events
  - An execution path of the system is obtained

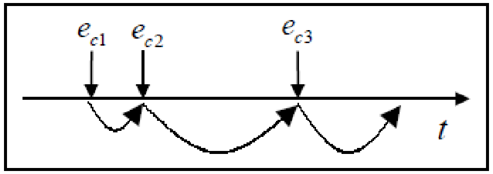
Simulation Process

- Initialization
  - » Time, initial system state and other counters are set to initial values
  - » Future events and their time of occurrence are calculated
- Time advance
  - » Time advances for the time of the next event (the one with smallest time)
  - » Next event is processed and the system state is updated
  - » Counters and other variables are updated
  - » Future events are calculated and stored in Events List
- Events List
  - » List of events   E(OccurenceTime, EventType, ...)
  - » Events ordered by their increasing occurrence time
- End of simulation
  - » Counters and other variables are kept for analysis
  - » Simulations may be repeated with different sequence of random numbers
  - » New independent samples increase the confidence in the results obtained

Simulation of the Poisson Process

- Poisson Process
  - » λ: client arrival rate (customer/s)
  - » Event e_c : client arrives
  - » Processing event k → determines occurrence of event k+1



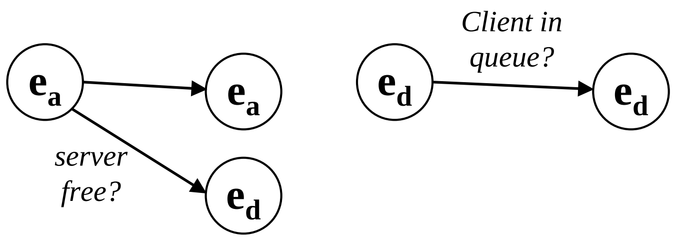


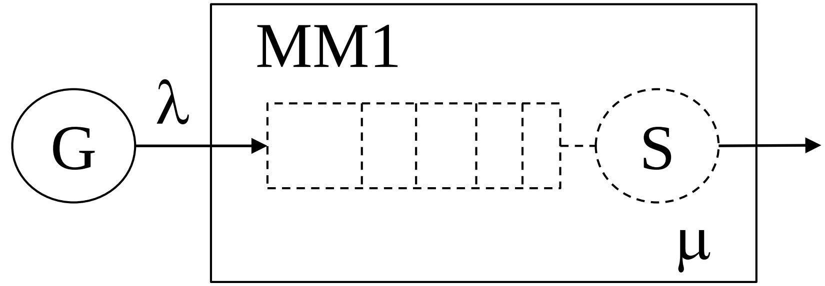
  - » Inter client arrival interval **c** has an exponential distribution F(c)
    - $F(c) = 1 - e^{-\lambda c}$ E[c]= 1/λ
  - » If **u** is randomly generated in [0,1] (uniform distribution, random function) then a sample of c may be obtained as

$$u = F(c) \iff c = F^{-1}(u)$$

$$c = -\frac{1}{\lambda}\ln(1-u) \iff c = -\frac{1}{\lambda}\ln(u)$$

A Simple Queue Model






```
main()
{
  < create sch, g, m >
  sch.schedule(0.4, g.send() );
  sch.schedule( 10, sch.stop() );
  sch.loop();
  <clean up>
}
```

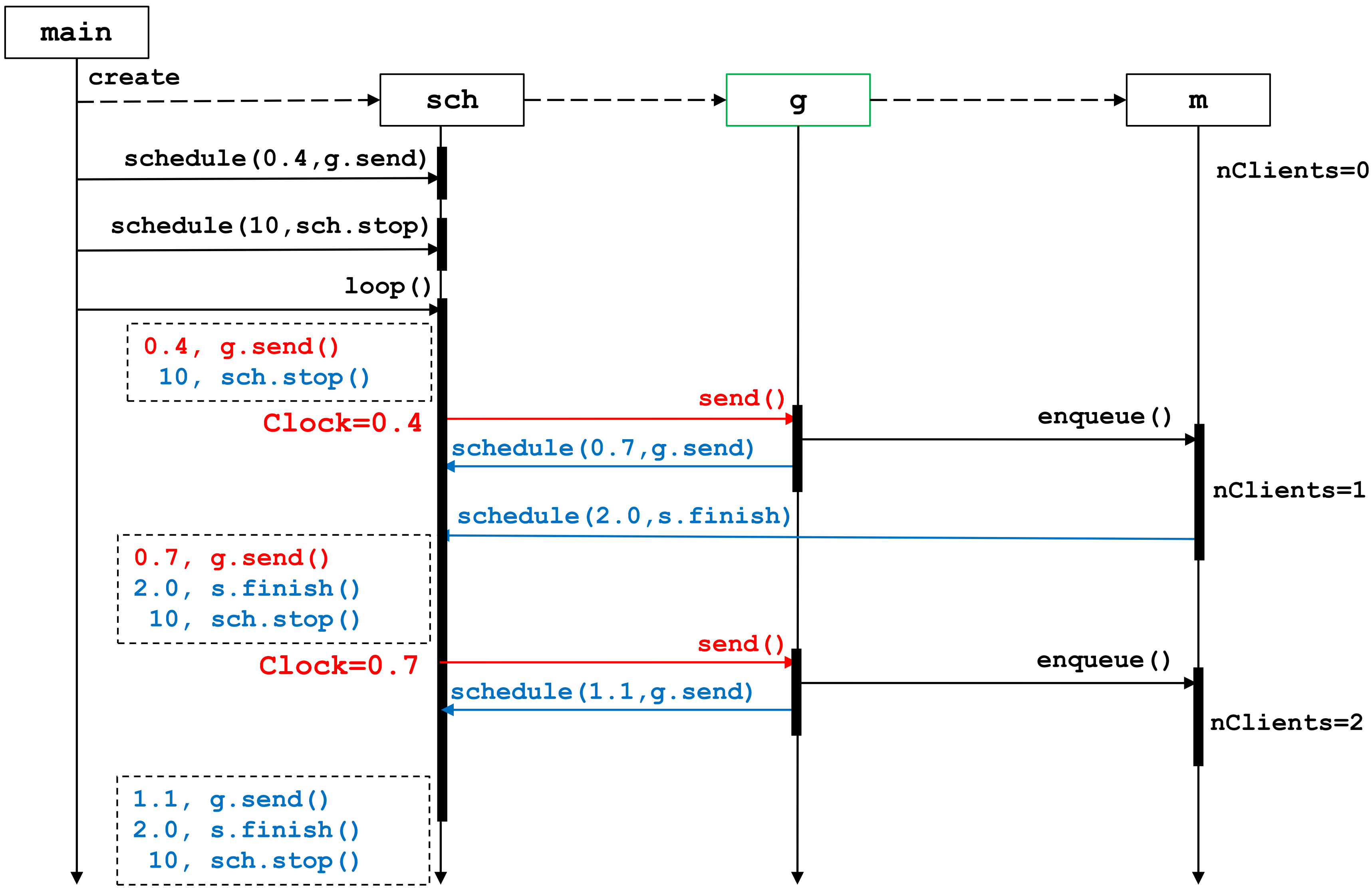
G
send()

MM1
nClients enqueue() finish()

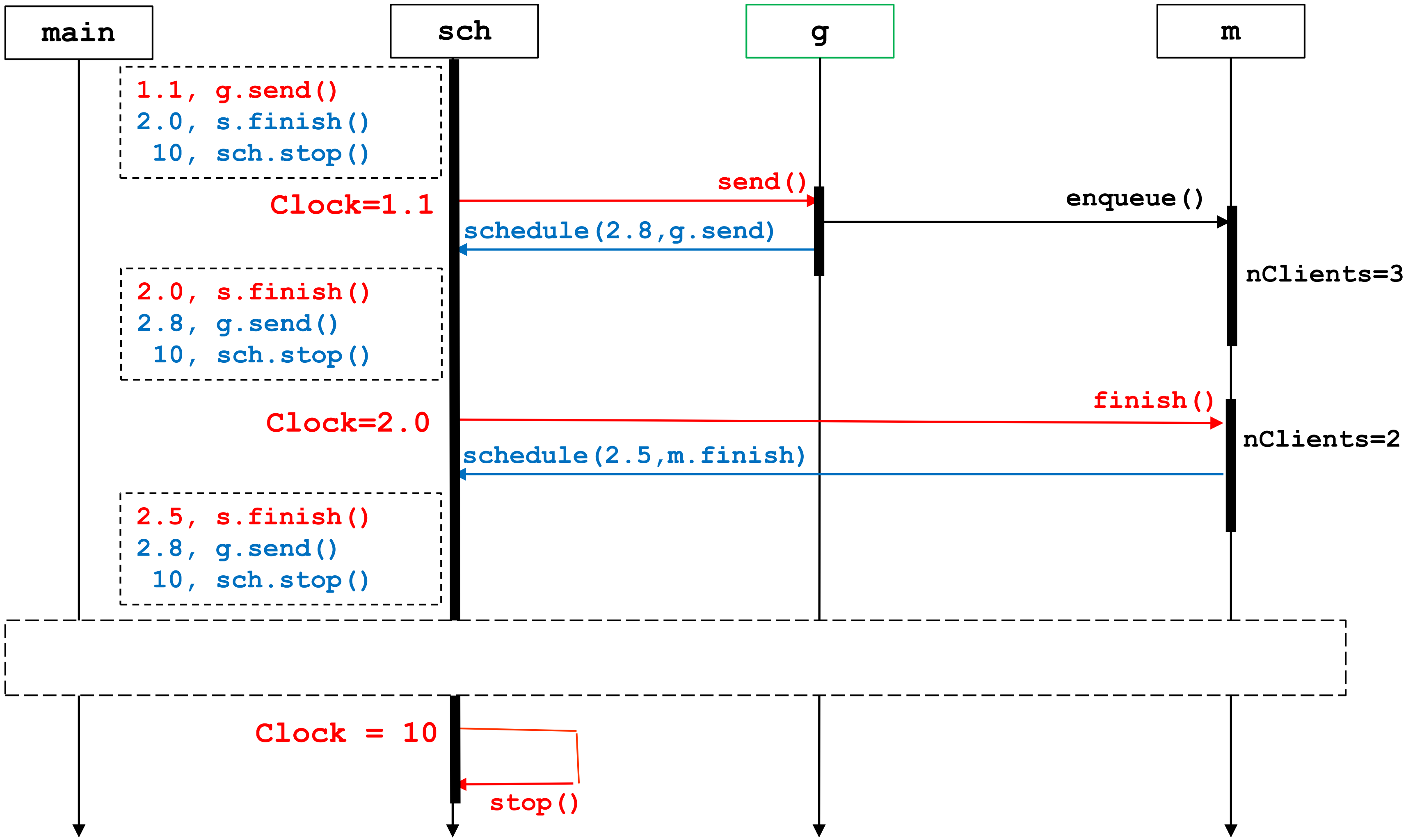
Scheduler
schedule() loop() getFirst() stop()

```
Scheduler::loop()
{
  do
  {
    event=getFirst() // fist event;
    clock = event.scheduledTime;
    event.function();
  } while (eventListNotEmpty && noStop)
}
```

Discrete Event Simulation in Execution (./..)



Discrete Event Simulation in Execution (../..)



What to Study?

- Formulate a question
  - » E.g.: How long does it take to serve N customers ?
- Answer
  - » This quantity is a random variable → no single answer exists
  - » It depends on
    - – the various inter-arrival times for these customers
    - – their corresponding service times
- Let us suppose we carry out one simulation
  - » ending after N customers are served
  - » and observe the termination time T_N
- T_N is an "estimate" of the quantity of interest

Performance Measures

- Expected Average System Time
- Server Utilization
- Mean Queue Length

Expected Average System Time

- Random variable S_N
- Time spent in the system by every single customer $S_1, S_2, \ldots, S_N$
- The estimate is $\hat{S}_N = \frac{1}{N} \sum_{k=1}^{N} S_k$

Server Utilization

- Probability the server is busy during total time it takes to serve N customers: ρ_N
- Fraction of time during which the queue length is positive

- T(i)
  - » total observed time during which the queue length is i, i = 0, 1, . . .

- E*stimate* $\hat{\rho}_N$

$$\hat{\rho}_N = \frac{\sum_{i=1}^{\infty} T(i)}{T_N} = 1 - \frac{T(0)}{T_N} \qquad T_N = \sum_{i=0}^{\infty} T(i)$$

Mean Queue Length

- Q_N : average queue length over the time interval required to serve N customers

- $p_N(i)$ = probability that queue length is i over the time interval required to serve N customers, i = 0, 1, . . .

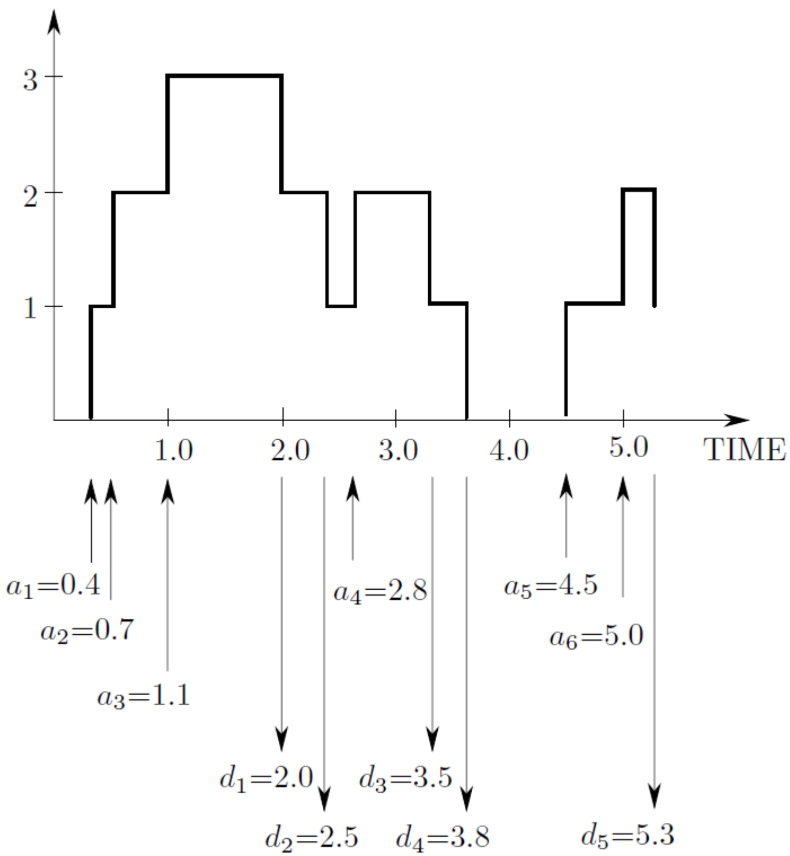
- By definition $Q_N = \sum_{i=0}^{\infty} i p_N(i)$

- *Estimate* $\hat{Q}_N$ $\hat{p}_N(i) = \frac{T(i)}{T_N}$ $\hat{Q}_N = \frac{1}{T_N} \sum_{i=0}^{\infty} iT(i)$

- During simulation we should maintain data variables

T(0), *T*(1), . . . for all observed queue lengths *i* = 0, 1, . . .

Sample path for N=5



QUEUE LENGTH OCCUPANCY TIMES	
T(0)	1.1
T(1)	1.4
T(2)	1.9
T(3)	0.9
T(4)	0.0

$$T_N = 5.3\,\text{s}$$

Arrival and departure times are a_k and d_k

Estimates of the Performance Measures

- Expected Average System Time $$\hat{S}_5 = \frac{1.6 + 1.8 + 2.4 + 1.0 + 0.8}{5} = 1.52$$

- Server Utilization $$\hat{\rho}_5 = 1 - \frac{1.1}{5.3} \approx 0.79$$

- Mean Queue Length $$\hat{Q}_5 = \frac{1}{5.3} \sum_{i=0}^{\infty} iT(i)$$ $$= \frac{(0 \cdot 1.1) + (1 \cdot 1.4) + (2 \cdot 1.9) + (3 \cdot 0.9) + 0 + \cdots}{5.3} \approx 1.49$$

Output Analysis

- Simulation output data is used to estimate random variables
  - Moments, percentiles
  - Distribution functions, if possible

- Random variable X
  - Function of the state of the system
  - Becomes a stochastic process {X (t)} ➔ X (t) distribution depends on t

Non-terminating Simulation

- Study the steady-state behavior of a system

- The longer the simulation, the better, but there are limitations

- A stopping rule
  - » Let simulation length be T and the estimate X (T)
  - » Extend simulation length to 2T and obtain X (2T)
  - » If $|X(2T) - X(T)| < \text{Error}$, STOP
  - » Otherwise, extend the simulation to 3T and repeat for $|X(3T) - X(2T)|$
  - » and so on …

Point estimation

- $X_1, X_2, \ldots, X_n$ is a stochastic sequence of IID random variables
  - There is a single probability distribution function
  - Let θ be the mean of that distribution
  - We aim to estimate θ from $X_1, X_2, \ldots, X_n$
  - We can obtain the point estimate $\hat{\theta}_n$ based on n samples

$$\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad E[\hat{\theta}_n] = \frac{1}{n}\sum_{i=1}^{n} E[X_i] = \frac{1}{n}\sum_{i=1}^{n} \theta = \theta$$

- For the stochastic process $\{X(t)\}$ $\qquad \hat{\theta}_T = \frac{1}{T}\int_0^T X(t)dt$

Point Estimation

- The sample mean $\hat{\theta}_n$ is also a random variable

$$Var[\hat{\theta}_n] = E[(\hat{\theta}_n - \theta)^2] = E\left[\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \theta\right)^2\right] = \frac{1}{n^2}E\left[\left(\sum_{i=1}^{n}(X_i - \theta)\right)^2\right] = \frac{\sigma^2}{n}$$

- We resort to the *sample variance*

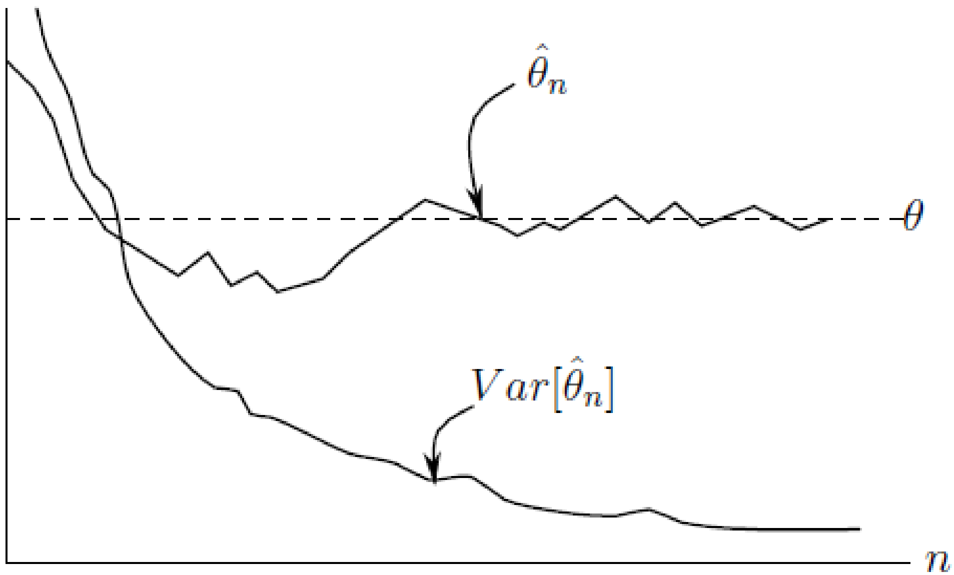
$$\hat{\sigma}^2(\hat{\theta}_n) = \frac{S_n^2}{n} = \frac{1}{n(n-1)}\sum_{i=1}^{n}(X_i - \hat{\theta}_n)^2 \qquad S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \hat{\theta}_n\right)^2$$

Conclusion

Let $X_1, X_2, \ldots, X_n$ form an iid sequence of random variables with mean $\theta < \infty$. Then,

$$\hat{\theta}_n \to \theta \quad \text{with probability 1, as } n \to \infty$$

where $\hat{\theta}_n$ is the sample mean of $X_1, X_2, \ldots, X_n$.



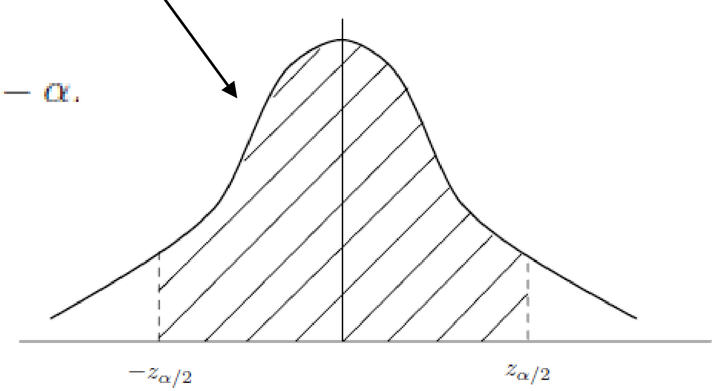
Interval estimation

- $X_1, X_2, \ldots, X_n$, form an IID sequence $\hat{\theta}_n \to \theta$ as $n \to \infty$

$$Z_n = \frac{\hat{\theta}_n - E\left[\hat{\theta}_n\right]}{\sqrt{\mathrm{Var}\left[\hat{\theta}_n\right]}} = \frac{\hat{\theta}_n - \theta}{\sqrt{\sigma^2/n}} \qquad n \to \infty,$$

F_n comes arbitrarily close to a normal distribution with mean zero and variance equal to 1.

$$P\left[-z_{\alpha/2} \le Z \le z_{\alpha/2}\right] = 1 - \alpha.$$



Interval Estimation

- $(1-\alpha)$ confidence interval for θ is given by

$$P\left[\hat{\theta}_n - z_{\alpha/2}\sqrt{S_n^2/n} \le \theta \le \hat{\theta}_n + z_{\alpha/2}\sqrt{S_n^2/n}\right] \approx 1-\alpha$$

- Confidence interval is specified by its lower and upper endpoints

$$L_{n,\alpha} = \hat{\theta}_n - z_{\alpha/2}\sqrt{S_n^2/n} \quad \text{and} \quad U_{n,\alpha} = \hat{\theta}_n + z_{\alpha/2}\sqrt{S_n^2/n}$$

- If n is not $very\ large$, we should use Student´s distribution

$$P\left[\hat{\theta}_n - t_{n-1,\alpha/2}\sqrt{S_n^2/n} \le \theta \le \hat{\theta}_n + t_{n-1,\alpha/2}\underline{\sqrt{S_n^2/n}}\right] \approx 1-\alpha$$

Estimated standard deviation

- Where
  - » n is the number of samples
  - » $1-\alpha$ is the confidence interval (ex. 90%)

For Non-IID Sequences …

- IF $X_1, X_2, \ldots, X_n$ do not form IID sequence
  - » E.g. X_k is the time customer k spends in system and it depends on X_{k-1}
- We should define some performance measure of interest
  - » $L(X_1, X_2, \ldots, X_M)$
  - » For instance $$L(X_1, X_2, \ldots, X_M) = \frac{1}{M}\sum_{i=1}^{M} X_i$$
- By repeating simulation *n* times, using different random numbers
  - » We can assume $L_1, L_2, \ldots, L_n$ *is an IID sequence*
  - » estimate $$\hat{\theta}_n = \frac{1}{n}\sum_{j=1}^{n} L_j$$
  - » and proceed as before (confidence interval, Student´s distribution)

Confidence Interval - Example

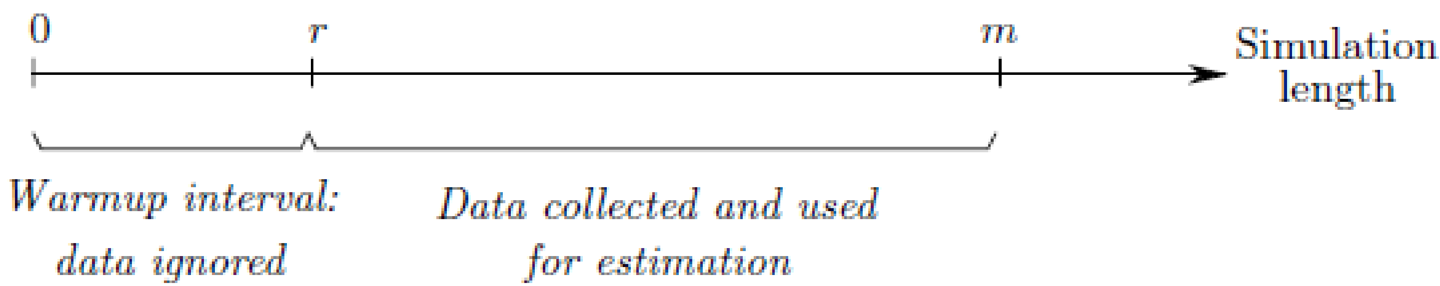
- Example: 10 samples:
  - X_1=1.20; X_2=1.50; X_3=1.68; X_4=1.89; X_5=0.95;
  - X_6=1.49; X_7= 1.58; X_8=1.55 ; X_9= 0.50; X_{10}=1.05
  - $\hat{\theta}_{10} = \frac{1}{10}\sum_{i=1}^{10} X_i = 1.34$
  - $\hat{\sigma}\left(\hat{\theta}_{10}\right) = \sqrt{\frac{1}{10*(10-1)}\sum_{i=1}^{10}[X_i \; - 1.34]^2} = 0{,}13$
  - $CI(1-\alpha\ ) = \hat{\theta}_{10} \pm z_{\alpha/2}\hat{\sigma}\left(\hat{\theta}_{10}\right) = CI(90\%) = 1.34 \pm 1.65 \times 0.13 = 1.34 \pm 0.21$

$(\mathbf{1-\alpha})$	$z_{\alpha/2}$
$\mathbf{80\%}$	$\mathbf{1.28}$
$\mathbf{90\%}$	$\mathbf{1.65}$
$\mathbf{95\%}$	$\mathbf{1.96}$
$\mathbf{98\%}$	$\mathbf{2.33}$
$\mathbf{99\%}$	$\mathbf{2.58}$

- *Assuming a Normal distribution*
- *For a small number of samples we should use t-student´s distribution (see Wikipedia)*

Non-terminating simulations – Replication with Deletions

- We are interested in the steady state
- Avoid transient state!



$$\hat{\theta}_{m,r} = \frac{1}{m-r} \sum_{i=r+1}^{m} X_i$$

References

- Christos G. Cassandras, Stéphane Lafortune, Introduction to Discrete Event Systems, Springer, 2008
- Jean-Yves Le Boudec, Performance Evaluation of Computer and Communication Systems, EPFL Press, 2010