

Analysis of Simulation Results

Raj Jain
Washington University
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

<http://www.cse.wustl.edu/~jain/cse567-17/>



- ❑ Analysis of Simulation Results
- ❑ Model Verification Techniques
- ❑ Model Validation Techniques
- ❑ Transient Removal
- ❑ Terminating Simulations
- ❑ Stopping Criteria: Variance Estimation
- ❑ Variance Reduction

Model Verification vs. Validation

- ❑ Verification \Rightarrow Debugging
- ❑ Validation \Rightarrow Model = Real world

- ❑ Four Possibilities:
 1. Unverified, Invalid
 2. Unverified, Valid
 3. Verified, Invalid
 4. Verified, Valid

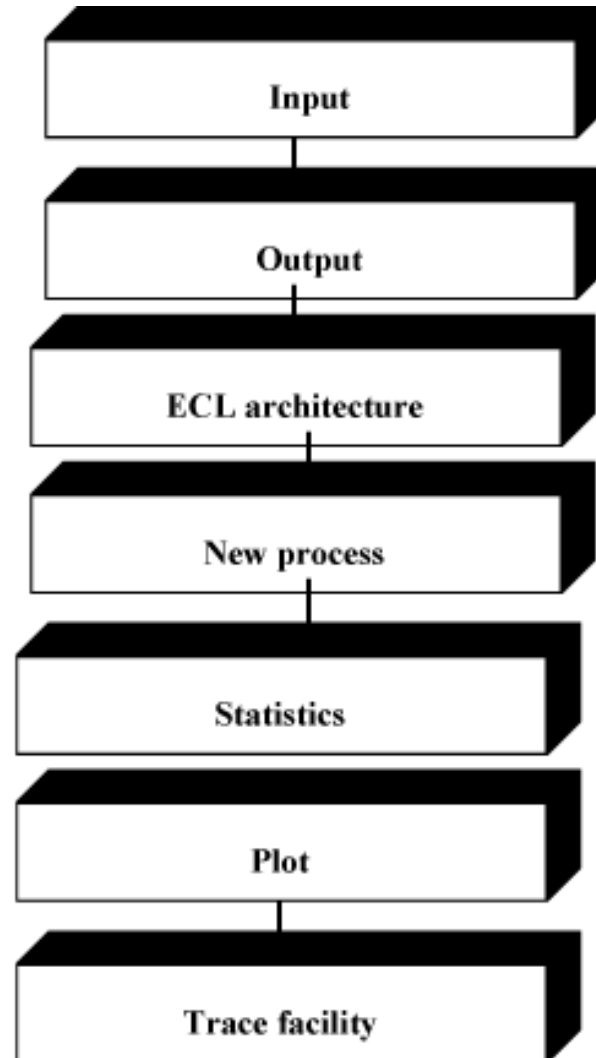
Model Verification Techniques

1. Top Down Modular Design
2. Anti-bugging
3. Structured Walk-Through
4. Deterministic Models
5. Run Simplified Cases
6. Trace
7. On-Line Graphic Displays
8. Continuity Test
9. Degeneracy Tests
10. Consistency Tests
11. Seed Independence

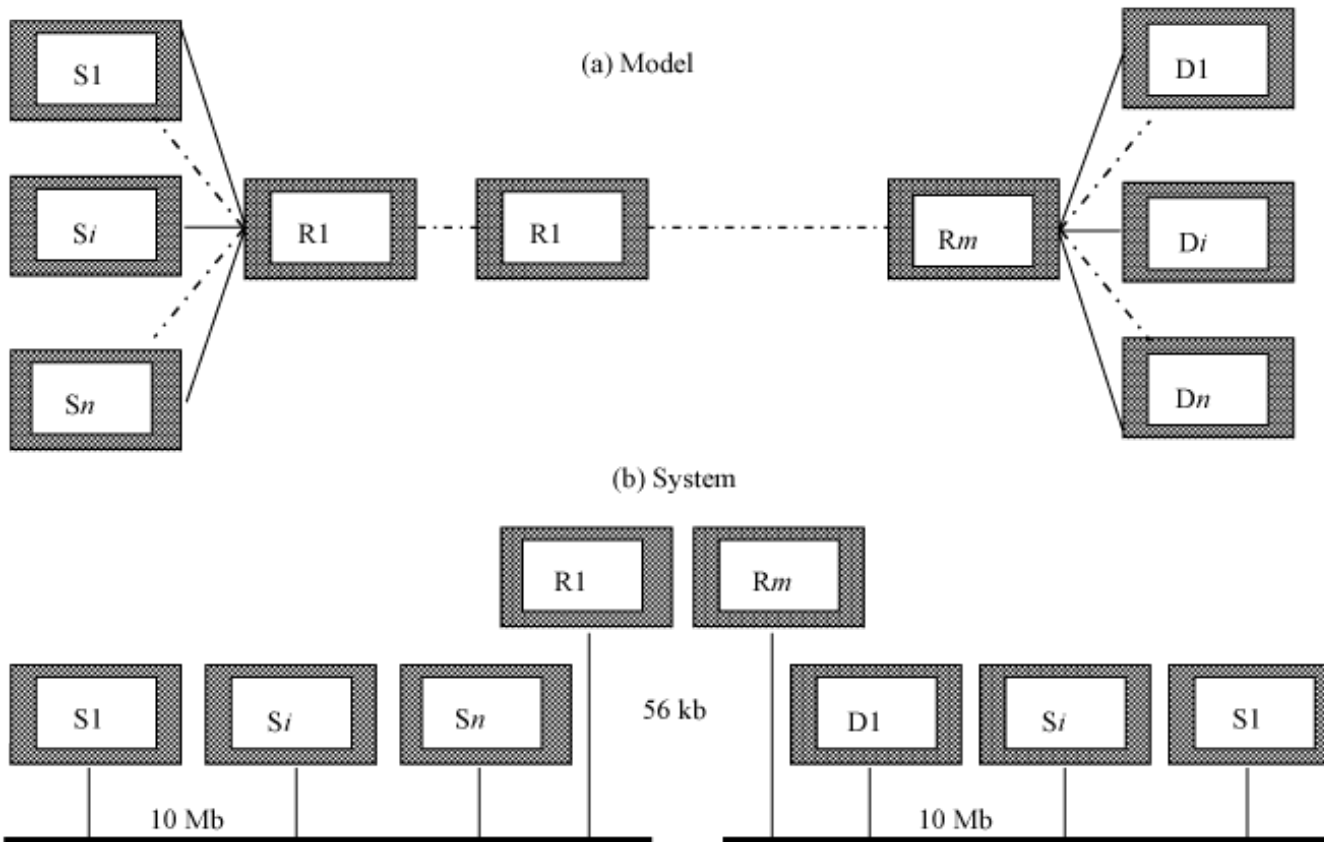
Top Down Modular Design

- ❑ Divide and Conquer
- ❑ Modules = Subroutines, Subprograms, Procedures
 - Modules have well defined interfaces
 - Can be independently developed, debugged, and maintained
- ❑ Top-down design
 - ⇒ Hierarchical structure
 - ⇒ Modules and sub-modules

Top Down Modular Design (Cont)



Top Down Modular Design (Cont)



Verification Techniques

- ❑ **Anti-bugging:** Include self-checks:
 - Probabilities = 1
 - Jobs left = Generated - Serviced
- ❑ **Structured Walk-Through:**
 - Explain the code another person or group
 - Works even if the person is sleeping
- ❑ **Deterministic Models:** Use constant values
- ❑ **Run Simplified Cases:**
 - Only one packet
 - Only one source
 - Only one intermediate node

Trace

- ❑ Trace = Time-ordered list of events and variables
- ❑ Several levels of detail:
 - Events trace
 - Procedure trace
 - Variables trace
- ❑ User selects the detail
 - Include on and off

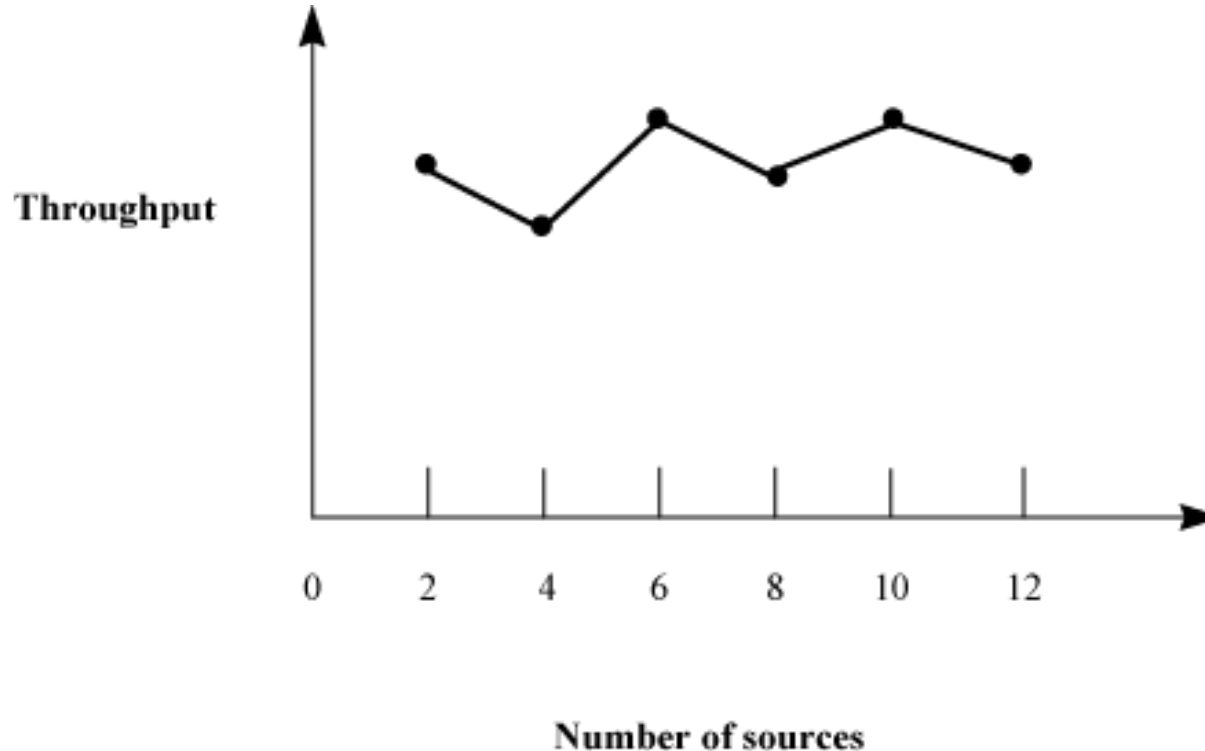
- ❑ See Fig 25.3 in the Text Book on page 418 for a sample trace

On-Line Graphic Displays

- ❑ Make simulation interesting
- ❑ Help selling the results
- ❑ More comprehensive than trace

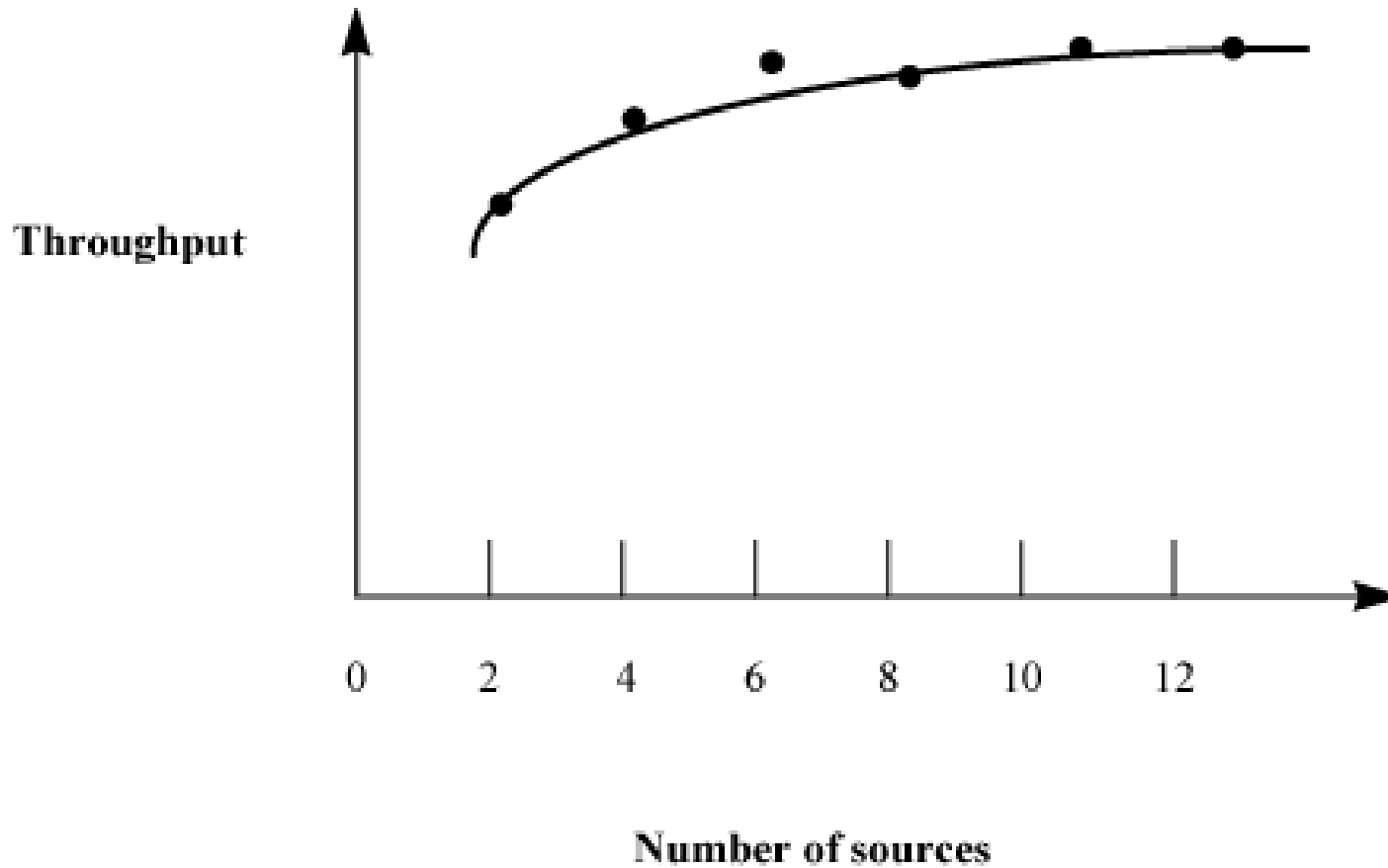
Continuity Test

- ❑ Run for different values of input parameters
- ❑ Slight change in input \Rightarrow slight change in output
- ❑ Before:



Continuity Test (Cont)

□ After:



More Verification Techniques

- ❑ **Degeneracy Tests:** Try extreme configuration and workloads
 - ❑ One CPU, Zero disk
- ❑ **Consistency Tests:**
 - Similar result for inputs that have same effect
 - ❑ Four users at 100 Mbps vs. Two at 200 Mbps
 - Build a test library of continuity, degeneracy and consistency tests
- ❑ **Seed Independence:** Similar results for different seeds

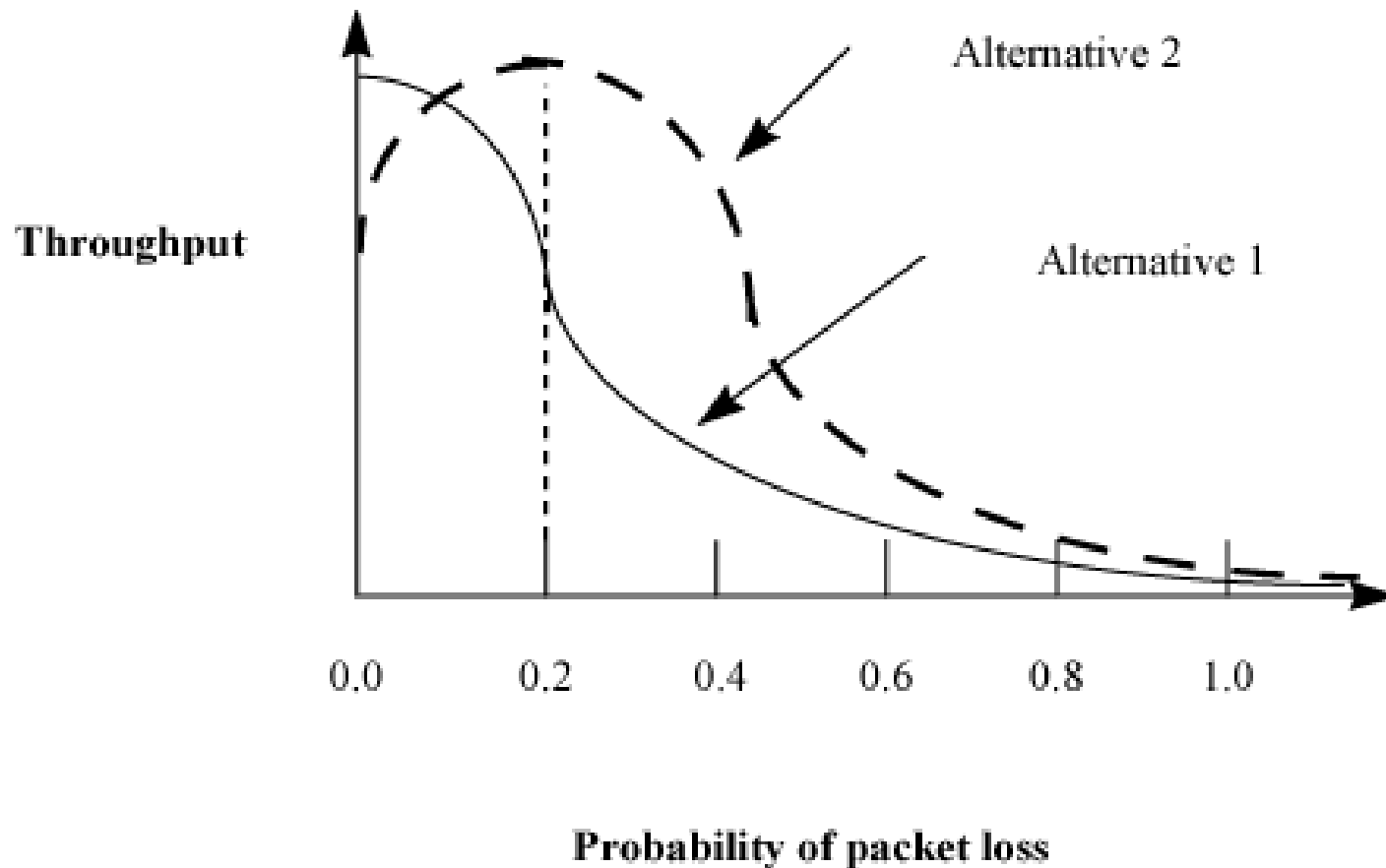
Model Validation Techniques

- ❑ Validation techniques for one problem may not apply to another problem.
 - ❑ Aspects to Validate:
 1. Assumptions
 2. Input parameter values and distributions
 3. Output values and conclusions
 - ❑ Techniques:
 1. Expert intuition
 2. Real system measurements
 3. Theoretical results
- ⇒ $3 \times 3 = 9$ validation tests

Expert Intuition

- ❑ Most practical and common way
- ❑ Experts = Involved in design, architecture, implementation, analysis, marketing, or maintenance of the system
- ❑ Selection = fn of Life cycle stage
- ❑ Present assumption, input, output
- ❑ Better to validate one at a time
- ❑ See if the experts can distinguish simulation vs. measurement

Expert Intuition (Cont)



Real System Measurements

- ❑ Compare assumptions, input, output with the real world
- ❑ Often infeasible or expensive
- ❑ Even one or two measurements add to the validity

Theoretical Results

- ❑ Analysis = Simulation
- ❑ Used to validate analysis also
- ❑ Both may be invalid
- ❑ Use theory in conjunction with experts' intuition
 - E.g., Use theory for a large configuration
 - Can show that the model is not invalid

Exercise 25.1

Imagine that you have been called as an expert to review a simulation study. Which of the following simulation results would you consider non-intuitive and would want it carefully validated:

1. The throughput of a system increases as its load increases.
2. The throughput of a system decreases as its load increases.
3. The response time increases as the load increases.
4. The response time of a system decreases as its load increases.
5. The loss rate of a system decreases as the load increases.

This is not a homework. Try but do not submit. Check answer in the book.

Transient Removal

- ❑ Generally steady state performance is interesting
- ❑ Remove the initial part
- ❑ No exact definition \Rightarrow Heuristics:
 1. Long Runs
 2. Proper Initialization
 3. Truncation
 4. Initial Data Deletion
 5. Moving Average of Independent Replications
 6. Batch Means

Transient Removal Techniques

❑ Long Runs:

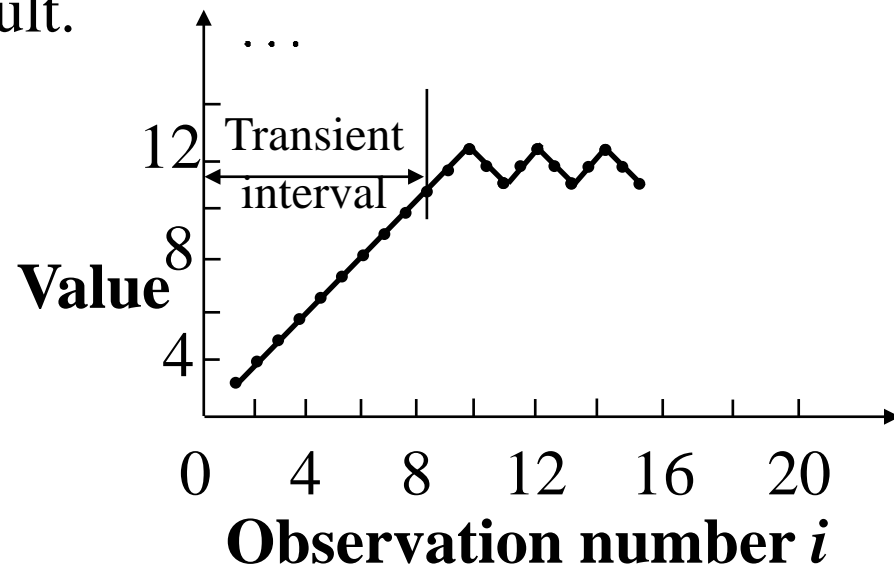
- Wastes resources
- Difficult to insure that it is long enough

❑ Proper Initialization:

- Start in a state close to expected steady state
⇒ Reduces the length and effect of transient state

Truncation

- ❑ Assumes variability is lower during steady state
- ❑ Plot max-min of $n-l$ observation for $l=1, 2, \dots$
- ❑ When $(l+1)$ th observation is neither the minimum nor maximum \Rightarrow transient state ended
- ❑ At $l=9$, Range = (9, 11), next observation = 10
- ❑ Sometimes incorrect result.



Terminating Simulations

- ❑ Transient performance is of interest
E.g., Network traffic
- ❑ System shuts down \Rightarrow Do not need transient removal.
- ❑ Final conditions:
 - May need to exclude the final portion from results
 - Techniques similar to transient removal

Treatment of Leftover Entities

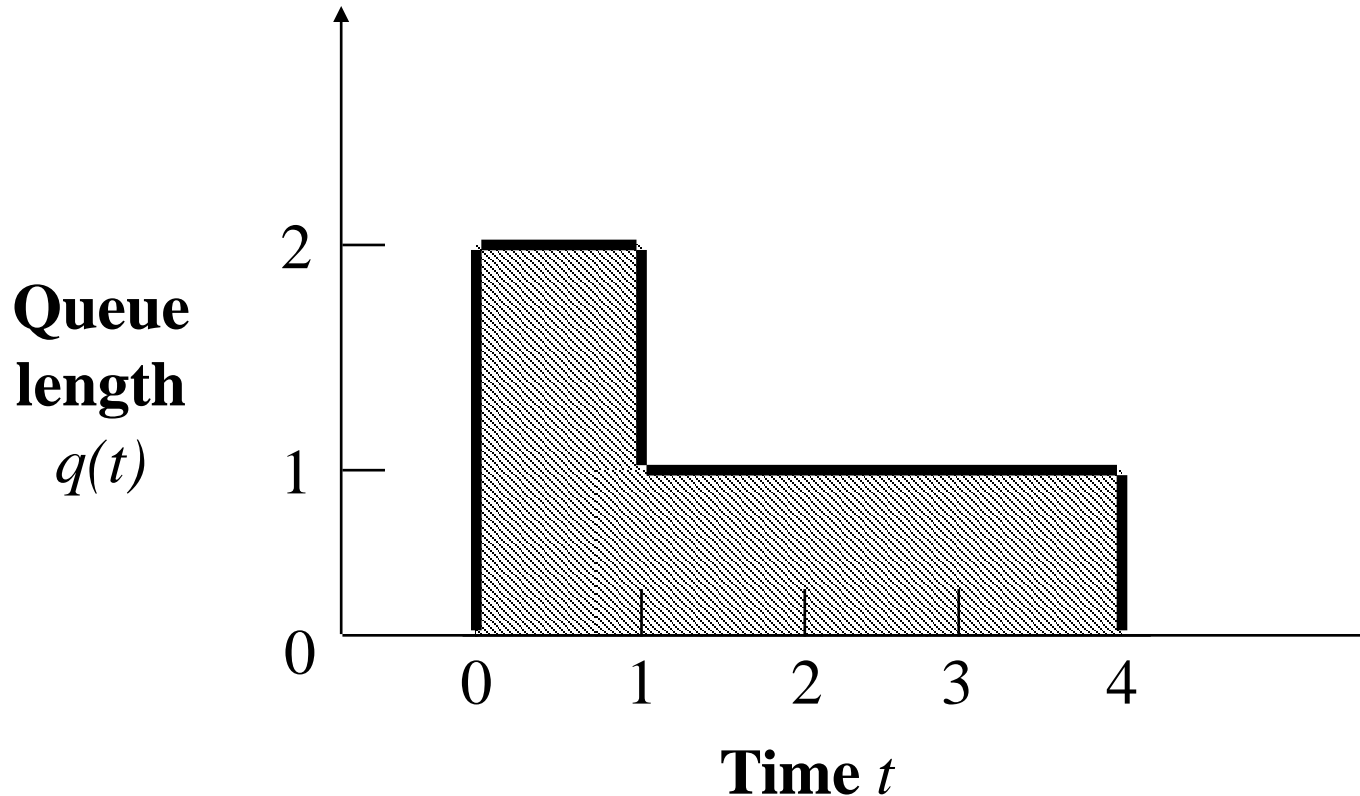
□ Mean service time $= \frac{\text{Total service time}}{\text{Number of jobs that completed service}}$

□ Mean waiting time $= \frac{\text{Sum of waiting time}}{\text{Number of jobs that received service}}$

□ Mean Queue Length $\neq \frac{\sum_{j=1}^n \text{Queue length at event } j}{\text{Number of events } n}$

$$= \frac{1}{T} \int_0^T \text{Queue_length}(t) dt$$

Example 25.3: Treatment of Leftover Entities



- ❑ Three events: Arrival at $t=0$, departures at $t=1$ and $t=4$
- ❑ $Q = 2, 1, 0$ at these events. Avg $Q = (2+1+0)/3 = 1$
- ❑ Avg $Q = \text{Area}/4 = 5/4$

Stopping Criteria: Variance Estimation

- ❑ Run until confidence interval is narrow enough

$$\bar{x} \pm z_{1-\alpha/2} \sqrt{\text{Var}(\bar{x})}$$

- ❑ For Independent observations:

$$\text{Var}(\bar{x}) = \frac{\text{Var}(x)}{n}$$

- ❑ Independence not applicable to most simulations.
- ❑ Large waiting time for i th job
⇒ Large waiting time for $(i+1)$ th job
- ❑ For correlated observations:

$$\text{Actual variance} \gg \frac{\text{Var}(x)}{n}$$

Variance Estimation Methods

1. Independent Replications
2. Batch Means
3. Method of Regeneration

Independent Replications

- Assumes that means of independent replications are independent
- Conduct m replications of size $n+n_0$ each
 1. Compute a mean for each replication:

$$\bar{x}_i = \frac{1}{n} \sum_{j=n_0+1}^{n_0+n} x_{ij} \quad i = 1, 2, \dots, m$$

2. Compute an overall mean for all replications:

$$\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i$$

Independent Replications (Cont)

3. Calculate the variance of replicate means:

$$\text{Var}(\bar{x}) = \frac{1}{m-1} \sum_{i=1}^m (\bar{x}_i - \bar{\bar{x}})^2$$

4. Confidence interval for the mean response is:

$$\left[\bar{\bar{x}} \mp z_{1-\alpha/2} \sqrt{\text{Var}(\bar{x})/m} \right]$$

- ❑ Keep replications large to avoid waste
- ❑ Ten replications generally sufficient

Batch Means

- ❑ Also called method of sub-samples
- ❑ Run a long simulation run
- ❑ Discard initial transient interval, and Divide the remaining observations run into several batches or sub-samples.
 1. Compute means for each batch:

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_{ij} \quad i = 1, 2, \dots, m$$

2. Compute an overall mean:

$$\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i$$

Batch Means (Cont)

3. Calculate the variance of batch means:

$$\text{Var}(\bar{x}) = \frac{1}{m-1} \sum_{i=1}^m (\bar{x}_i - \bar{\bar{x}})^2$$

4. Confidence interval for the mean response is:

$$\left[\bar{\bar{x}} \mp z_{1-\alpha/2} \sqrt{\text{Var}(\bar{x})/m} \right]$$

- ❑ Less waste than independent replications
- ❑ Keep batches long to avoid correlation
- ❑ Check: Compute the auto-covariance of successive batch means:

$$\text{Cov}(\bar{x}_i, \bar{x}_{i+1}) = \frac{1}{m-2} \sum_{i=1}^{m-1} (\bar{x}_i - \bar{\bar{x}})(\bar{x}_{i+1} - \bar{\bar{x}})$$

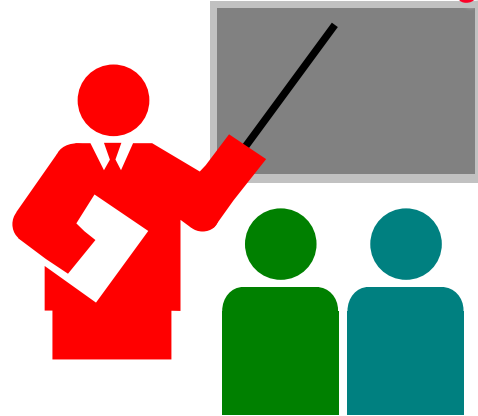
- ❑ Double n until autocovariance is small.

Case Study 25.1: Interconnection Networks

- ❑ Indirect binary n-cube networks:
Used for processor-memory interconnection
- ❑ Two stage network with full fan out.
- ❑ At 64, autocovariance
< 1% of sample variance

Batch Size	Autocovariance	Variance
1	-0.18792	1.79989
2	0.02643	0.81173
4	0.11024	0.42003
8	0.08979	0.26437
16	0.04001	0.17650
32	0.01108	0.10833
64	0.00010	0.06066
128	-0.00378	0.02992
256	0.00027	0.01133
512	0.00069	0.00503
1024	0.00078	0.00202

Summary



1. Verification = Debugging
⇒ Software development techniques
2. Validation ⇒ Simulation = Real ⇒ Experts involvement
3. Transient Removal: Initial data deletion, batch means
4. Terminating Simulations = Transients are of interest
5. Stopping Criteria: Independent replications, batch means, method of regeneration
6. Variance reduction is not for novice

Scan This to Download These Slides



Raj Jain

<http://rajjain.com>

Related Modules



CSE567M: Computer Systems Analysis (Spring 2013),

https://www.youtube.com/playlist?list=PLjGG94etKypJEKjNAa1n_1X0bWWNyZcof

CSE473S: Introduction to Computer Networks (Fall 2011),

https://www.youtube.com/playlist?list=PLjGG94etKypJWOSPMh8Azcg5e_10TiDw



Wireless and Mobile Networking (Spring 2016),

https://www.youtube.com/playlist?list=PLjGG94etKypKeb0nzyN9tSs_HCd5c4wXF

CSE571S: Network Security (Fall 2011),

<https://www.youtube.com/playlist?list=PLjGG94etKypKvzfVtutHcPFJXumyyg93u>



Video Podcasts of Prof. Raj Jain's Lectures,

<https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw>