Queuing Theory and Traffic Analysis
CS 552
Richard Martin
Rutgers University
Queuing theory

• View network as collections of queues
  – FIFO data-structures
• Queuing theory provides probabilistic analysis of these queues
• Examples:
  – Average length
  – Probability queue is at a certain length
  – Probability a packet will be lost
Little’s Law

- **Little’s Law:** Mean number tasks in system = arrival rate x mean response time
  - Observed before, Little was first to prove
- Applies to any system in equilibrium, as long as nothing in black box is creating or destroying tasks
Proving Little’s Law

\[ J = \text{Shaded area} = 9 \]

Same in all cases!
Definitions

• J: “Area” from previous slide
• N: Number of jobs (packets)
• T: Total time
• λ: Average arrival rate
  – = N/T
• W: Average time job is in the system
  – = J/N
• L: Average number of jobs in the system
  – = J/T
Proof: Method 1: Definition

\[ J = TL = NW \]

\[ L = (\frac{N}{T})W \]

\[ L = (\square)W \]
Proof: Method 2: Substitution

\[ L = (\emptyset) W \]
\[ L = \left( \frac{N}{T} \right) W \]
\[ \frac{J}{T} = \left( \frac{N}{T} \right) \left( \frac{J}{N} \right) \]
\[ \frac{J}{T} = \frac{J}{T} \quad \text{Tautology} \]
Example using Little’s law

- Observe 120 cars in front of the Lincoln Tunnel
  - Observe 32 cars/minute depart over a period where no cars in the tunnel at the start or end (e.g. security checks)
- What is average waiting time before and in the tunnel?

\[ W = \frac{L}{\lambda} = \left( \frac{120}{32} \right) = 3.75 \text{min} \]
Model Queuing System

Strategy:
Use Little’s law on both the complete system and its parts to reason about average time in the queue.
Kendal Notation

- Six parameters in shorthand
  - First three typically used, unless specified
  1. Arrival Distribution
     - Probability of a new packet arrives in time $t$
  2. Service Distribution
     - Probability distribution packet is serviced in time $t$
  3. Number of servers
  4. Total Capacity (infinite if not specified)
  5. Population Size (infinite)
  6. Service Discipline (FCFS/FIFO)
Distributions

- M: Exponential
- D: Deterministic (e.g. fixed constant)
- $E_k$: Erlang with parameter $k$
- $H_k$: Hyperexponential with param. $k$
- G: General (anything)

- M/M/1 is the simplest ‘realistic’ queue
Kendal Notation Examples

- **M/M/1:**
  - Exponential arrivals and service, 1 server, infinite capacity and population, FCFS (FIFO)

- **M/M/m**
  - Same, but M servers

- **G/G/3/20/1500/SPF**
  - General arrival and service distributions, 3 servers, 17 queue slots (20-3), 1500 total jobs, Shortest Packet First
M/M/1 queue model
Analysis of M/M/1 queue

• Goal: A closed form expression of the probability of the number of jobs in the queue \( P_i \) given only \( \lambda \) and \( \mu \)
Solving queuing systems

• Given:
  • \( l \): Arrival rate of jobs (packets on input link)
  • \( m \): Service rate of the server (output link)

• Solve:
  – \( L \): average number in queuing system
  – \( L_q \) average number in the queue
  – \( W \): average waiting time in whole system
  – \( W_q \): average waiting time in the queue

• 4 unknown’s: need 4 equations
Solving queuing systems

• 4 unknowns: L, L_q, W, W_q
• Relationships using Little’s law:
  – L = \lambda W
  – L_q = \lambda W_q (steady-state argument)
  – W = W_q + (1/\mu)
• If we know any 1, can find the others
• Finding L is hard or easy depending on the type of system. In general:

\[ L = \sum_{n=0}^{\infty} nP_n \]
Equilibrium conditions

inflow = outflow

1: \((P_{n-1} + P_n) = P_{n+1} + P_n\)

2: \(P_n = P_{n+1}\)

stability: 3: \(\frac{P_{n-1}}{P_n}, P_n = \frac{D}{L}, P_n \neq 1\)
Solving for $P_0$ and $P_n$

1: \[ P_1 = \square P_0 \ , \ P_2 = (\square)^2 P_0 \ , \ P_n = (\square)^n P_0 \]

2: \[ \sum_{n=0}^{\infty} P_n = 1 \ , \ P_0 \sum_{n=0}^{\infty} \square^n = 1 \ , \ P_0 = \frac{1}{\sum_{n=0}^{\infty} \square^n} \]

3: \[ \sum_{n=0}^{\infty} \square^n = \frac{1}{1 \square}, \square < 1 \quad \text{(geometric series)} \]

4: \[ P_0 = \frac{1}{\square^n} = \frac{1}{1 \square^n} = 1 \square \]

5: \[ P_n = (\square)^n (1 \square) \]
Solving for $L$

$$L = \sum_{n=0}^{\infty} n P_n = \sum_{n=0}^{\infty} n^n (1 \bar{L}) = (1 \bar{L}) \sum_{n=1}^{\infty} n^n$$

$$\left(1 \bar{L}\right) \sum_{n=0}^{\infty} \frac{1}{n!} \frac{d}{d\bar{L}} \left(\frac{1}{1 \bar{L}}\right) = \left(1 \bar{L}\right) \sum_{n=0}^{\infty} \frac{d}{d\bar{L}} \left(\frac{1}{1 \bar{L}}\right)$$

$$\left(1 \bar{L}\right) \sum_{1}^{\infty} \frac{1}{(1 \bar{L})^2} = \frac{1}{(1 \bar{L})} = \frac{1}{(1 \bar{L})}$$
Solving $W$, $W_q$ and $L_q$

\[ W = \frac{L}{q} = \left( \frac{1}{l} \right) = \frac{1}{l} \]

\[ W_q = W \Box \frac{1}{l} = \left( \frac{1}{l} \right) \Box \left( \frac{1}{l} \right) = \frac{1}{l^2} \]

\[ L_q = \Box W_q = \Box \frac{1}{l^2} = \frac{1}{l^2} \]
Response Time vs. Arrivals

\[ W = \frac{1}{\rho} \]
Stable Region

Waiting vs. Utilization

linear region
Empirical Example

RAID
Modeled vs Measured

Response Time (msec)

NFS Ops/Sec

M/M/m system
Example

• Measurement of a network gateway:
  – mean arrival rate ($\lambda$): 125 Packets/s
  – mean response time per packet: 2 ms

• Assuming exponential arrivals & departures:
  – What is the service rate, $\mu$?
  – What is the gateway’s utilization?
  – What is the probability of n packets in the gateway?
  – mean number of packets in the gateway?
  – The number of buffers so P(overflow) is $<10^{-6}$?
Example (cont)

The service rate, \( m = \frac{1}{0.002} = 500 \text{ pps} \)

utilization = \( \mu = \left( \frac{\mu}{\lambda} \right) = 0.25\% \)

\( P(n) \) packets in the gateway =

\[
P_0P_n = (1 - \mu)(\mu^n) = (0.75)(0.25^n)
\]
Example (cont)

Mean # in gateway (L) =

\[
\frac{\frac{n}{10^6}}{10^6} = \frac{0.25}{10^{0.25}} = 0.33
\]

to limit loss probability to less than 1 in a million:

\[
\frac{n}{10^6^6}
\]
Properties of a Poisson processes

- Poisson process = exponential distribution between arrivals/departures/service

\[ P(\text{arrival} < t) = 1 \Box e^{\lambda t} \]

- Key properties:
  - memoryless
    - Past state does not help predict next arrival
  - Closed under:
    - Addition
    - Subtraction
Addition and Subtraction

• Merge:
  – two poisson streams with arrival rates \( l_1 \) and \( l_2 \):
    • new poisson stream: \( l_3 = l_1 + l_2 \)

• Split:
  – If any given item has a probability \( P_1 \) of “leaving” the stream with rate \( l_1 \):
    \( l_2 = (1 - P_1) l_1 \)
Queuing Networks

\[ l_1 = l_2 + l_3 \]
\[ l_3 = l_4 + l_5 \]
\[ l_6 = l_2 + l_4 \]
\[ l_7 = l_5 \]
Bridging Router Performance and Queuing Theory

Sigmetrics 2004

Slides by N. Hohn*, D. Veitch*, K. Papagiannaki, C. Diot
Motivation

- End-to-end packet delay is an important metric for performance and Service Level Agreements (SLAs)
- Building block of end-to-end delay is through router delay
- Measure the delays incurred by *all* packets crossing a single router
Overview

• Full Router Monitoring
• Delay Analysis and Modeling
• Delay Performance: Understanding and Reporting
Measurement Environment

BackBone 1

BackBone 2

Customer 1

GPS clock signal
Packet matching

<table>
<thead>
<tr>
<th>Set</th>
<th>Link</th>
<th>Matched pkts</th>
<th>% traffic C2-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4</td>
<td>In</td>
<td>215987</td>
<td>0.03%</td>
</tr>
<tr>
<td>C1</td>
<td>In</td>
<td>70376</td>
<td>0.01%</td>
</tr>
<tr>
<td>BB1</td>
<td>In</td>
<td>345796622</td>
<td>47.00%</td>
</tr>
<tr>
<td>BB2</td>
<td>In</td>
<td>389153772</td>
<td>52.89%</td>
</tr>
<tr>
<td>C2</td>
<td>out</td>
<td>735236757</td>
<td><strong>99.93%</strong></td>
</tr>
</tbody>
</table>
Overview

- Full Router Monitoring
- Delay Analysis and Modeling
- Delay Performance: Understanding and Reporting
Definition of delay
Store & Forward Datapath

- Store: storage in input linecard’s memory
- Forwarding decision
- Storage in dedicated Virtual Output Queue (VOQ)
- Decomposition into fixed-size cells
- Transmission through switch fabric cell by cell
- Packet reconstruction
- Forward: Output link scheduler

Not part of the system
Delays: 1 minute summary

BB1-In to C2-Out

MAX
Mean
MIN
Store & Forward Datapath

- Store: storage in input linecard’s memory
- Forwarding decision
- Storage in dedicated Virtual Output Queue (VOQ)
- Decomposition into fixed-size cells
- Transmission through switch fabric cell by cell
- Packet reconstruction
- Forward: Output link scheduler

Not part of the system
Minimum Transit Time

Packet size dependent minimum delay.
Store & Forward Datapath

- Store: storage in input linecard’s memory
- Forwarding decision
- Storage in dedicated Virtual Output Queue (VOQ)
- Decomposition into fixed-size cells
- Transmission through switch fabric cell by cell
- Packet reconstruction
- Forward: Output link scheduler

\[ L_i \oplus j(L) \]

\[ \text{FIFO queue} \]
Modeling
Modeling

Fluid queue with a delay element at the front
Model Validation

U(t)
Error as a function of time
Modeling results

• A crude model performs well!
  – As simpler/simpler than an M/M/1 queue
• Use effective link bandwidth
  – account for encapsulation
• Small gap between router performance and queuing theory!
• The model defines Busy Periods: time between the arrival of a packet to the empty system and the time when the system becomes empty again.
Overview

- Full Router Monitoring
- Delay Analysis and Modeling
- Delay Performance: Understanding and Reporting
On the Delay Performance

• Model allows for router performance evaluation when arrival patterns are known

• Goal: metrics that
  – Capture operational-router performance
  – Can answer performance questions directly

• Busy Period structures contain all delay information
  – BP better than utilization or delay reporting
Busy periods metrics
Property of significant BPs
\[ d^{(T)}_{L,A,D} = D \left( 1 - \frac{L}{A} \right), \text{if } A \geq L \]
Issues

• Report (A,D) measurements
• There are millions of busy periods even on a lightly utilized router
• Interesting episodes are rare and last for a very small amount of time
Report BP joint distribution
Duration of Congestion Level-L
Conclusions

• Results
  – Full router empirical study
  – Delay modeling
  – Reporting performance metrics

• Future work
  – Fine tune reporting scheme
  – Empirical evidence of large deviations theory
Network Traffic Self-Similarity

Slides by Carey Williamson

Department of Computer Science
University of Saskatchewan
Introduction

• A classic measurement study has shown that aggregate Ethernet LAN traffic is self-similar [Leland et al 1993]

• A statistical property that is very different from the traditional Poisson-based models

• This presentation: definition of network traffic self-similarity, Bellcore Ethernet LAN data, implications of self-similarity
Measurement Methodology

- Collected lengthy traces of Ethernet LAN traffic on Ethernet LAN(s) at Bellcore
- High resolution time stamps
- Analyzed statistical properties of the resulting time series data
- Each observation represents the number of packets (or bytes) observed per time interval (e.g., 10 4 8 12 7 2 0 5 17 9 8 8 2...)
Self-Similarity: The intuition

- If you plot the number of packets observed per time interval as a function of time, then the plot looks “the same” regardless of what interval size you choose.
- E.g., 10 msec, 100 msec, 1 sec, 10 sec,...
- Same applies if you plot number of bytes observed per interval of time.
Self-Similarity: The Intuition

• In other words, self-similarity implies a “fractal-like” behavior: no matter what time scale you use to examine the data, you see similar patterns

• Implications:
  – Burstiness exists across many time scales
  – No natural length of a burst
  – Key: Traffic does not necessarily get “smoother” when you aggregate it (unlike Poisson traffic)
Self-Similarity Traffic Intuition (I)
Self-Similarity in Traffic Measurement II

Network Traffic

Poisson Traffic
Self-Similarity: The Math

- Self-similarity is a rigorous statistical property
  - (i.e., a lot more to it than just the pretty “fractal-like” pictures)
- Assumes you have time series data with finite mean and variance
  - i.e., covariance stationary stochastic process
- Must be a very long time series
  - infinite is best!
- Can test for presence of self-similarity
Self-Similarity: The Math

• Self-similarity manifests itself in several equivalent fashions:
  • Slowly decaying variance
  • Long range dependence
  • Non-degenerate autocorrelations
  • Hurst effect
Methods of showing Self-Similarity

H=0.5

R/S method

periodogram

Estimate H ≥ 0.8

H=1

variance time method

H=0.5
Slowly Decaying Variance

• The variance of the sample decreases more slowly than the reciprocal of the sample size.
• For most processes, the variance of a sample diminishes quite rapidly as the sample size is increased, and stabilizes soon.
• For self-similar processes, the variance decreases very slowly, even when the sample size grows quite large.
The “variance-time plot” is one means to test for the slowly decaying variance property.

Plots the variance of the sample versus the sample size, on a log-log plot.

For most processes, the result is a straight line with slope -1.

For self-similar, the line is much flatter.
Time Variance Plot

Variance

m
Variance-Time Plot

Variance of sample on a logarithmic scale

m
Variance-Time Plot

Sample size $m$
on a logarithmic scale
Variance-Time Plot

Variance

m
Variance-Time Plot

Variance vs. m
Variance-Time Plot

\[ \text{Slope} = -1 \text{ for most processes} \]
Variance-Time Plot

Slope flatter than -1 for self-similar process
Long Range Dependence

- Correlation is a statistical measure of the relationship, if any, between two random variables
- Positive correlation: both behave similarly
- Negative correlation: behave as opposites
- No correlation: behavior of one is unrelated to behavior of other
Long Range Dependence

• Autocorrelation is a statistical measure of the relationship, if any, between a random variable and itself, at different time lags
• Positive correlation: big observation usually followed by another big, or small by small
• Negative correlation: big observation usually followed by small, or small by big
• No correlation: observations unrelated
Autocorrelation coefficient can range between:
- +1 (very high positive correlation)
- -1 (very high negative correlation)

Zero means no correlation.

Autocorrelation function shows the value of the autocorrelation coefficient for different time lags k.
Autocorrelation Function

Maximum possible positive correlation
Autocorrelation Function

No observed correlation at all

Autocorrelation Coefficient

lag k

0 100
Autocorrelation Function

![Autocorrelation Function Diagram](chart.png)
Autocorrelation Function

Significant positive correlation at short lags

Autocorrelation Coefficient vs. lag k
Autocorrelation Function

No statistically significant correlation beyond this lag
Long Range Dependence

- For most processes (e.g., Poisson, or compound Poisson), the autocorrelation function drops to zero very quickly
  - usually immediately, or exponentially fast
- For self-similar processes, the autocorrelation function drops very slowly
  - i.e., hyperbolically, toward zero, but may never reach zero
- Non-summable autocorrelation function
Autocorrelation Function

![Graph of Autocorrelation Function](image)

- **Y-axis:** Autocorrelation Coefficient
- **X-axis:** lag k

The graph shows the autocorrelation function with a sharp decline from +1 to near 0 at lag 0, then remaining close to 0 for subsequent lags.
Autocorrelation Function

Typical short-range dependent process
Autocorrelation Function

Autocorrelation Coefficient

lag k

0 100
Autocorrelation Function

Typical long-range dependent process
Typical long-range dependent process

Typical short-range dependent process
Non-Degenerate Autocorrelations

• For self-similar processes, the autocorrelation function for the aggregated process is indistinguishable from that of the original process

• If autocorrelation coefficients match for all lags k, then called exactly self-similar

• If autocorrelation coefficients match only for large lags k, then called asymptotically self-similar
Autocorrelation Function

![Autocorrelation Function Graph]

Original self-similar process
Autocorrelation Function

![Graph showing the Autocorrelation Function with an original self-similar process indicated.](image-url)
Autocorrelation Function

![Graph showing autocorrelation function with two curves labeled 'Original self-similar process' and 'Aggregated self-similar process'.]
Aggregation

- Aggregation of a time series $X(t)$ means smoothing the time series by averaging the observations over non-overlapping blocks of size $m$ to get a new time series $X_m(t)$
Suppose the original time series $X(t)$ contains the following (made up) values

\[ 2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots \]

Then the aggregated series for $m = 2$ is:
Aggregation Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

\[2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots\]

• Then the aggregated series for $m = 2$ is:
Suppose the original time series $X(t)$ contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

Then the aggregated series for $m = 2$ is:

4.5
Suppose the original time series $X(t)$ contains the following (made up) values:

\[2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots\]

Then the aggregated series for $m = 2$ is:

\[4.5 \ 8.0\]
Suppose the original time series $X(t)$ contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1... 

Then the aggregated series for $m = 2$ is:

4.5 8.0 2.5
Suppose the original time series $X(t)$ contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

Then the aggregated series for $m = 2$ is:

4.5 8.0 2.5 5.0
Aggregation Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

• Then the aggregated series for $m = 2$ is:

4.5 8.0 2.5 5.0 6.0 7.5 7.0 4.0 4.5 5.0...
Suppose the original time series $X(t)$ contains the following (made up) values:

$2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots$

Then the aggregated time series for $m = 5$ is:
Aggregation: An Example

- Suppose the original time series $X(t)$ contains the following (made up) values:

\[
2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots
\]

Then the aggregated time series for $m = 5$ is:
Aggregation: An Example

• Suppose the original time series \( X(t) \) contains the following (made up) values:

\[
2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots
\]

Then the aggregated time series for \( m = 5 \) is:

\[
6.0
\]
Aggregation: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

Then the aggregated time series for $m = 5$ is:

6.0 4.4
Aggregation: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

\[ 2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots \]

Then the aggregated time series for $m = 5$ is:

\[ 6.0 \quad 4.4 \quad 6.4 \quad 4.8 \ldots \]
Aggregation: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

$$2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \ldots$$

Then the aggregated time series for $m = 10$ is:
Aggregation: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

```
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...
```

Then the aggregated time series for $m = 10$ is:

5.2
Aggregation: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

Then the aggregated time series for $m = 10$ is:

5.2

5.6
Autocorrelation Function

Autocorrelation Coefficient

Original self-similar process

Aggregated self-similar process

lag k
Hurst Effect

- For almost all naturally occurring time series, the rescaled adjusted range statistic (also called the R/S statistic) for sample size $n$ obeys the relationship
  
  $$E[R(n)/S(n)] = c \ n^H$$

where:

- $R(n) = \max(0, W_1, \ldots, W_n) - \min(0, W_1, \ldots, W_n)$
- $S^2(n)$ is the sample variance, and
- $W_k = \sum_{i=1}^{n} (X_i) \bigcap k \overline{X_n}$ for $k = 1, 2, \ldots, n$
Hurst Effect

- For models with only short range dependence, $H$ is almost always 0.5
- For self-similar processes, $0.5 < H < 1.0$
- This discrepancy is called the **Hurst Effect**, and $H$ is called the **Hurst parameter**
- **Single parameter** to characterize self-similar processes
R/S Statistic: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1
• There are 20 data points in this example
R/S Statistic: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:
  
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

• There are 20 data points in this example

• For R/S analysis with $n = 1$, you get 20 samples, each of size 1:
R/S Statistic: An Example

- Suppose the original time series $X(t)$ contains the following (made up) values:

  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- There are 20 data points in this example
- For R/S analysis with $n = 1$, you get 20 samples, each of size 1:
  
  Block 1: $X = 2, \ W = 0, \ R(n) = 0, \ S(n) = 0$

```
  n  1
```
R/S Statistic: An Example

• Suppose the original time series \( X(t) \) contains the following (made up) values:
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1
• There are 20 data points in this example
• For R/S analysis with \( n = 1 \), you get 20 samples, each of size 1:
  Block 2: \( X = 7, W = 0, R(n) = 0, S(n) = 0 \)
  \[
  \frac{n}{1} = \frac{1}{1}
  \]
R/S Statistic: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:
  
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

• For R/S analysis with $n = 2$, you get 10 samples, each of size 2:
R/S Statistic: An Example

- Suppose the original time series $X(t)$ contains the following (made up) values:
  $2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1$
- For R/S analysis with $n = 2$, you get 10 samples, each of size 2:
  Block 1: $X = 4.5$, $W = -2.5$, $W = 0$,
  $R(n) = 0 - (-2.5) = 2.5$, $S(n) = 2.5$,
  $R(n)/S(n) = 1.0^{1/2}$
• Suppose the original time series $X(t)$ contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

• For R/S analysis with $n = 2$, you get 10 samples, each of size 2:

Block 2: $X = 8.0$, $W = -4.0$, $W = 0$,

$R(n) = 0 - (-4.0) = 4.0$, $S(n) = 4.0$,

$R(n)/S(n) = 1.0^{\frac{1}{2}}$
R/S Statistic: An Example

- Suppose the original time series $X(t)$ contains the following (made up) values:
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1
- For R/S analysis with $n = 3$, you get 6 samples, each of size 3:
R/S Statistic: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

• For R/S analysis with $n = 3$, you get 6 samples, each of size 3:

  Block 1: $X = 4.3$, $W = -2.3$, $W = 0.3$, $W = 0$
  $R(n) = 0.3 - (-2.3) = 2.6$, $S(n) = 2.05$, $R(n)/S(n) = 1.30$
**R/S Statistic: An Example**

- Suppose the original time series \( X(t) \) contains the following (made up) values:
  \[ 2 \ 7 \ 4 \ 12 \ 5 \ 0 \ 8 \ 2 \ 8 \ 4 \ 6 \ 9 \ 11 \ 3 \ 3 \ 5 \ 7 \ 2 \ 9 \ 1 \]
- For R/S analysis with \( n = 3 \), you get 6 samples, each of size 3:
  - Block 2: \( X = 5.7, \ W = 6.3, \ W = 5.7, \ W = 0 \)
  - \( R(n) = 6.3 - (0) = 6.3, \ S(n) = 4.92, \)
  - \( R(n)/S(n) = \frac{6.3}{1.28} = 4.92 \)
R/S Statistic: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

• For R/S analysis with $n = 5$, you get 4 samples, each of size 5:
R/S Statistic: An Example

- Suppose the original time series $X(t)$ contains the following (made up) values:
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1
- For R/S analysis with $n = 5$, you get 4 samples, each of size 4:
  Block 1: $X = 6.0$, $W = -4.0$, $W = -3.0$, $W = -5.0$, $W = 1.0$, $W = 0$, $S(n) = 3.41$, $R(n) = 1.0 - (-5.0) = 6.0$, $\frac{R(n)}{S(n)} = 1.76$
R/S Statistic: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:
  2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

• For R/S analysis with $n = 5$, you get 4 samples, each of size 4:

  Block 2: $X = 4.4$, $W = -4.4$, $W = -0.8$,
  $W = -3.2$, $W = 0.4$, $W = 0$, $S(n) = 3.2$,
  $R(n) = 0.4 - (-4.4) = 4.8$, $\frac{1}{n} R(n)/S(n) = 1.5$
Suppose the original time series $X(t)$ contains the following (made up) values:

\[ 2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1 \]

For R/S analysis with $n = 10$, you get 2 samples, each of size 10:
R/S Statistic: An Example

• Suppose the original time series $X(t)$ contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

• For R/S analysis with $n = 20$, you get 1 sample of size 20:
Another way of testing for self-similarity, and estimating the Hurst parameter

Plot the R/S statistic for different values of n, with a log scale on each axis

If time series is self-similar, the resulting plot will have a straight line shape with a slope $H$ that is greater than 0.5

Called an R/S plot, or R/S pox diagram
R/S Pox Diagram

R/S Statistic

Block Size n
R/S Pox Diagram

R/S Statistic

R/S statistic \( R(n)/S(n) \) on a logarithmic scale

Block Size \( n \)
R/S Pox Diagram

Sample size $n$
on a logarithmic scale

Block Size $n$
R/S Pox Diagram

R/S Statistic vs. Block Size n
R/S Pox Diagram

R/S Statistic vs Block Size n

Slope 0.5
R/S Pox Diagram

R/S Statistic vs Block Size $n$

Slope 1.0

Slope 0.5
R/S Pox Diagram

Slope 1.0

Slope 0.5

R/S Statistic

Block Size n
R/S Pox Diagram

Slope 1.0

Self-similar process

Slope 0.5

Block Size n

R/S Statistic
R/S Pox Diagram

- Slope 1.0
- Slope 0.5
- Slope H (0.5 < H < 1.0) (Hurst parameter)
Self-Similarity Summary

- Self-similarity is an important mathematical property that has recently been identified as present in network traffic measurements.
- Important property: burstiness across many time scales, traffic does not aggregate well.
- There exist several mathematical methods to test for the presence of self-similarity, and to estimate the Hurst parameter $H$.
- There exist models for self-similar traffic.

- TCP *session* arrivals are well modeled by a Poisson process
- A number of WAN characteristics were well modeled by *heavy tailed* distributions
- *Packet* arrival process for two typical applications (TELNET, FTP) as well as aggregate traffic is *self-similar*
Another Study


- Analyzed WWW logs collected at clients over a 1.5 month period
  - First WWW client study
  - Instrumented MOSAIC
    - ~600 students
    - ~130K files transferred
    - ~2.7GB data transferred
Self-Similar Aspects of Web traffic

• One difficulty in the analysis was finding stationary, busy periods
  – A number of candidate hours were found
• All four tests for self-similarity were employed
  – $0.7 < H < 0.8$
Explaining Self-Similarity

- Consider a set of processes which are either ON or OFF
  - The distribution of ON and OFF times are heavy tailed
  - The aggregation of these processes leads to a self-similar process
- So, how do we get heavy tailed ON or OFF times?
Impact of File Sizes

- Analysis of client logs showed that ON times were, in fact, heavy tailed
  - Over about 3 orders of magnitude
- This lead to the analysis of underlying file sizes
  - Over about 4 orders of magnitude
  - Similar to FTP traffic
- Files available from UNIX file systems are typically heavy tailed
Heavy Tailed OFF times

• Analysis of OFF times showed that they are also heavy tailed
• Distinction between Active and Passive OFF times
  – Inter vs. Intra click OFF times
• Thus, ON times are more likely to be cause of self-similarity
Major Results from CB97

- Established that WWW traffic was self-similar
- Modeled a number of different WWW characteristics (focus on the tail)
- Provide an explanation for self-similarity of WWW traffic based on underlying file size distribution
Where are we now?

• There is no mechanistic model for Internet traffic
  – Topology?
  – Routing?
• People want to blame the protocols for observed behavior
• Multiresolution analysis may provide a means for better models
• Many people (vendors) chose to ignore self-similarity
  – Does it matter????
  – Critical opportunity for answering this question.