Modelling and Measuring LRD

Richard Clegg (richard@manor.york.ac.uk)

Networks and Nonlinear Dynamics Group,

Department of Mathematics,

University of York

Slides prepared using the Prosper package and LaTeX

Introduction

- This is the York side of an EPSRC funded project joint with Queen Mary, and BT Exact.
- The aim of the project is to investigate sources of Long-Range Dependence in the Internet.
- The project is to involve three activities:
  1. Data Collection and Analysis of real Internet data.
  2. Mathematical Modelling.
  3. Simulation Modelling (using ns as well as our own tools).
- The project is still at an early stage and input and advice is greatly welcomed.
- `http://gridlock.york.ac.uk/lrdsources/`

Causes of LRD in teletraffic

- Four causes for LRD in teletraffic (in bytes/unit time) are commonly identified in the literature:
  1. Traffic which is LRD at source (e.g. streaming video traffic).
  2. Traffic which arises from the transfer of data where the transferred files sizes have heavy tails.
  3. LRD arises from feedback mechanisms in the TCP protocol. (the timeout mechanism for example).
  4. LRD arises as a result of the natural aggregation of traffic in a network.
- In addition, the claim is sometimes made that LRD can transfer between multiplexing data streams.
- A main aim of this project is to attempt to identify the relative importance of these sources.

Project Overview

- Data Collection and Analysis
  - (Work started) Collection of `tcpdump` data from York's main internet connection.
  - (Preliminary work only) Collection of `traceroute` style data from a distributed network of machines — `pppd`.
- Mathematical Modelling
  - (Work started) A new model for LRD using Markov Chains.
  - (Not yet started) Extension to an existing model of TCP traffic.
- Simulation Modelling
  - (Work started) Simulation using ns to replicate long-range dependent traffic.

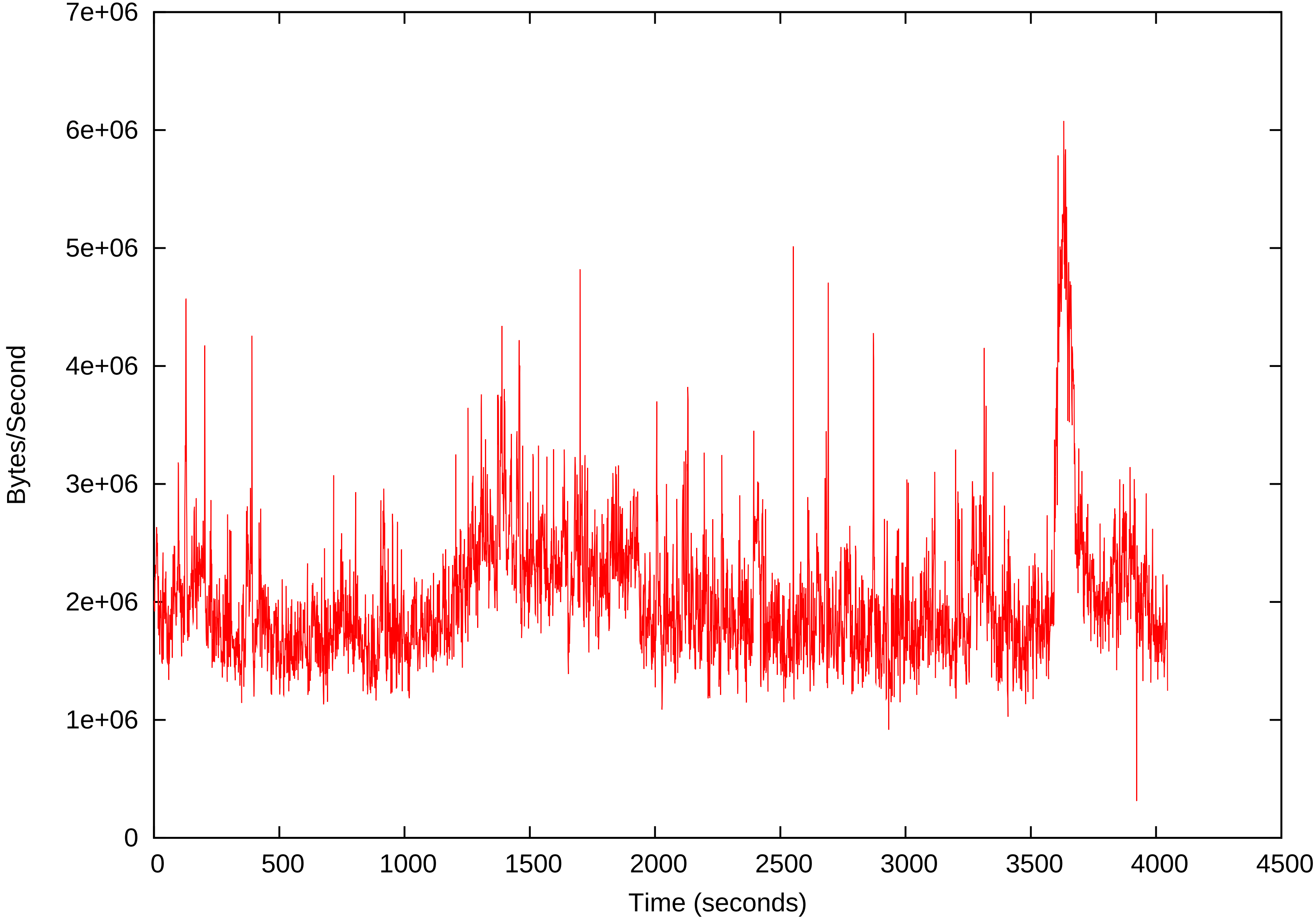
Data Collection — Data Facts

- Data collected at incoming/outgoing pipe at University of York.
- 8.23 GB of data in 13.6 million packets — 67 minutes of data.
- 7.81 GB of this data is TCP. 0.6MB of data is ICMP. 0.4GB of data UDP.
- Outgoing data: 1.95GB of data in 6.0 million packets (av size: 323 bytes).
- Incoming data: 6.29GB of data in 7.7 million packets (av size: 821 bytes).
- Data has been anonymised and is available for researchers on request (1Gb `tcpdump` format file).

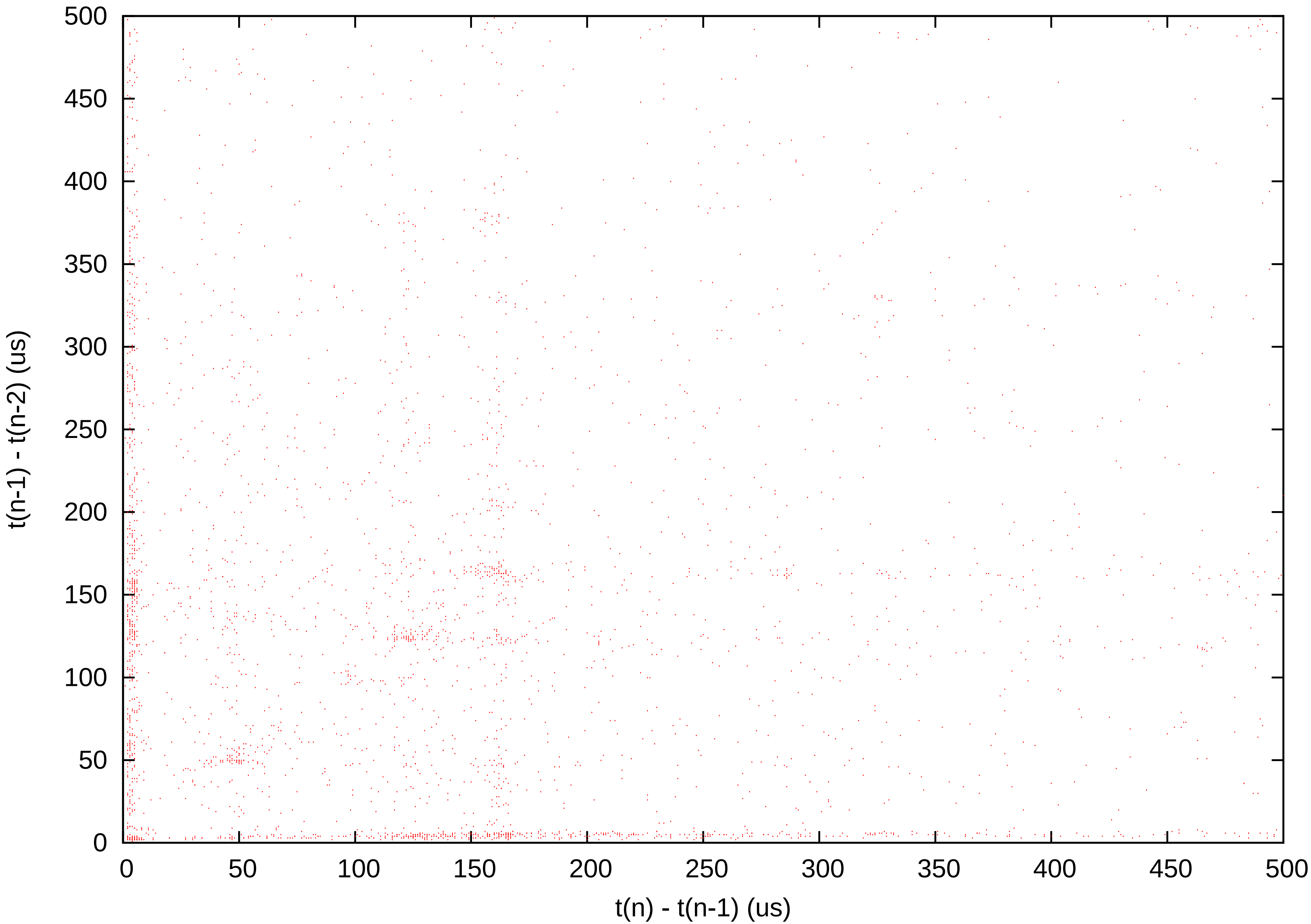
Disaggregating the data

- In addition to inbound and outbound we can break up traffic by port number.
- Ports are usually associated with particular services.
- Port 80 HTTP (5.78GB) — Web traffic is by far the bulk of the traffic
- Port 25 SMTP (226MB) — Email is large but not by comparison
- Port 21 and 20 FTP (230MB) — FTP is insignificant.
- Port 53 DNS (33MB) — DNS data doesn't seem to account for much (but this may be due to where we are looking).

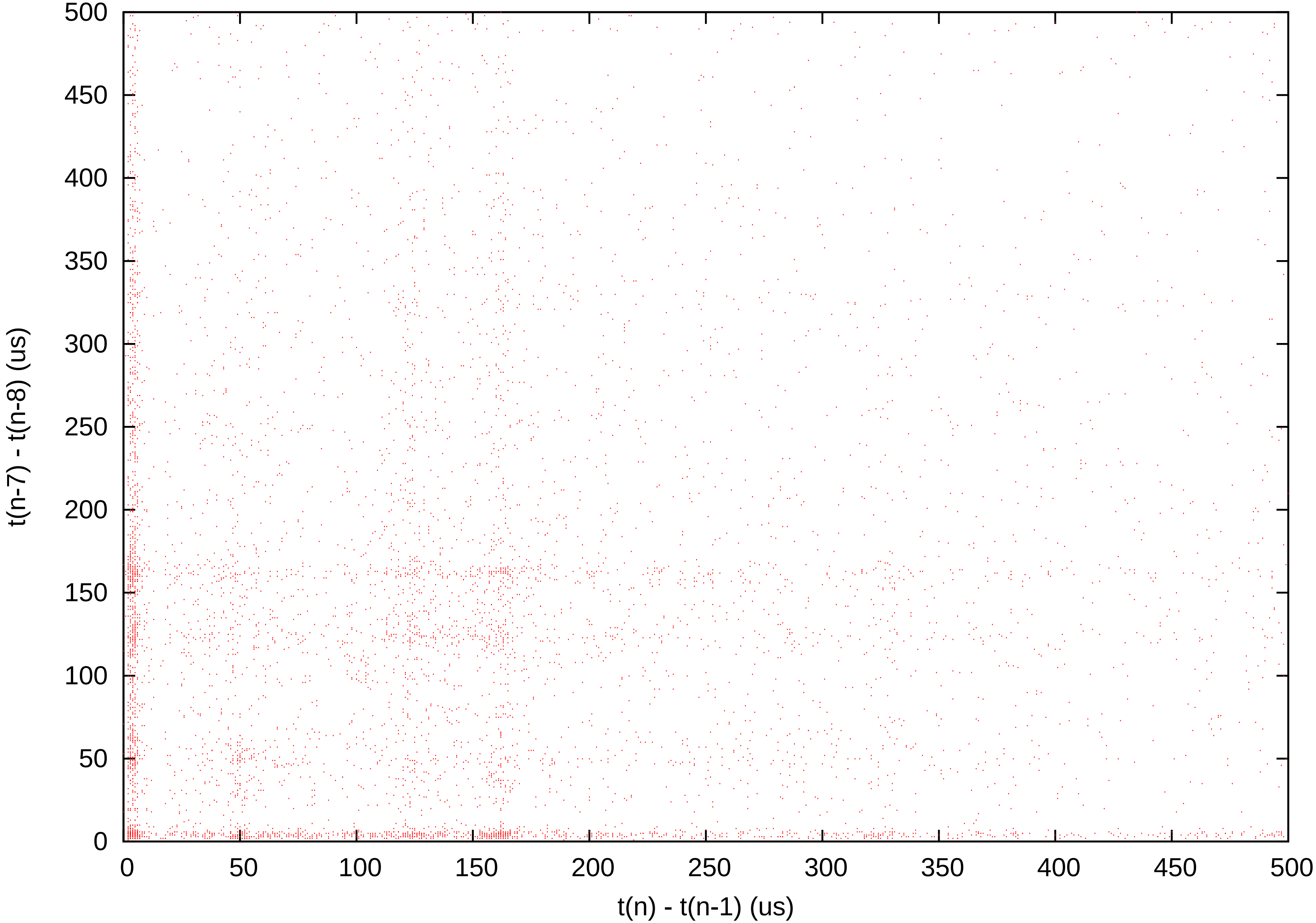
Raw Data — bytes/second



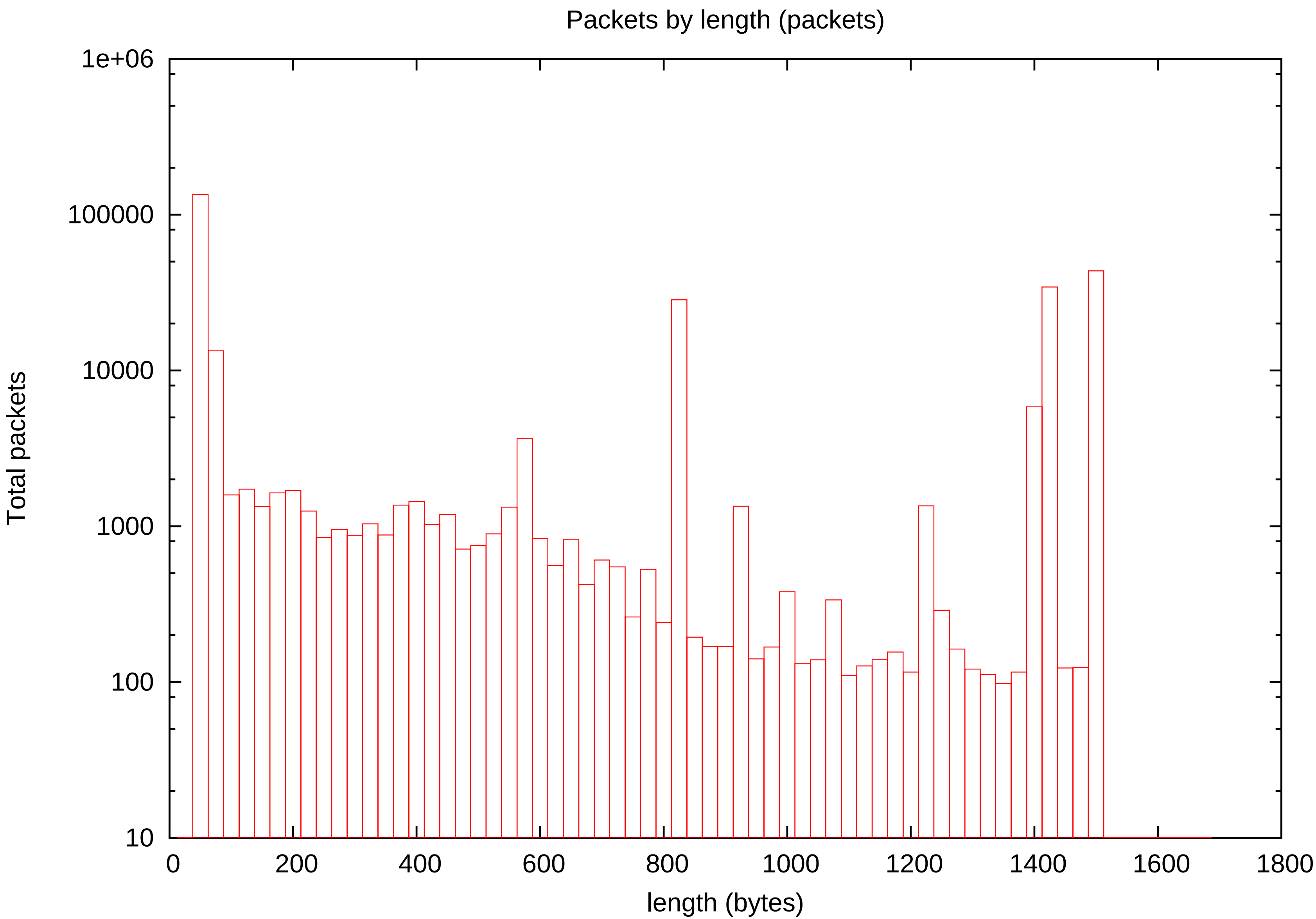
Time differences in data



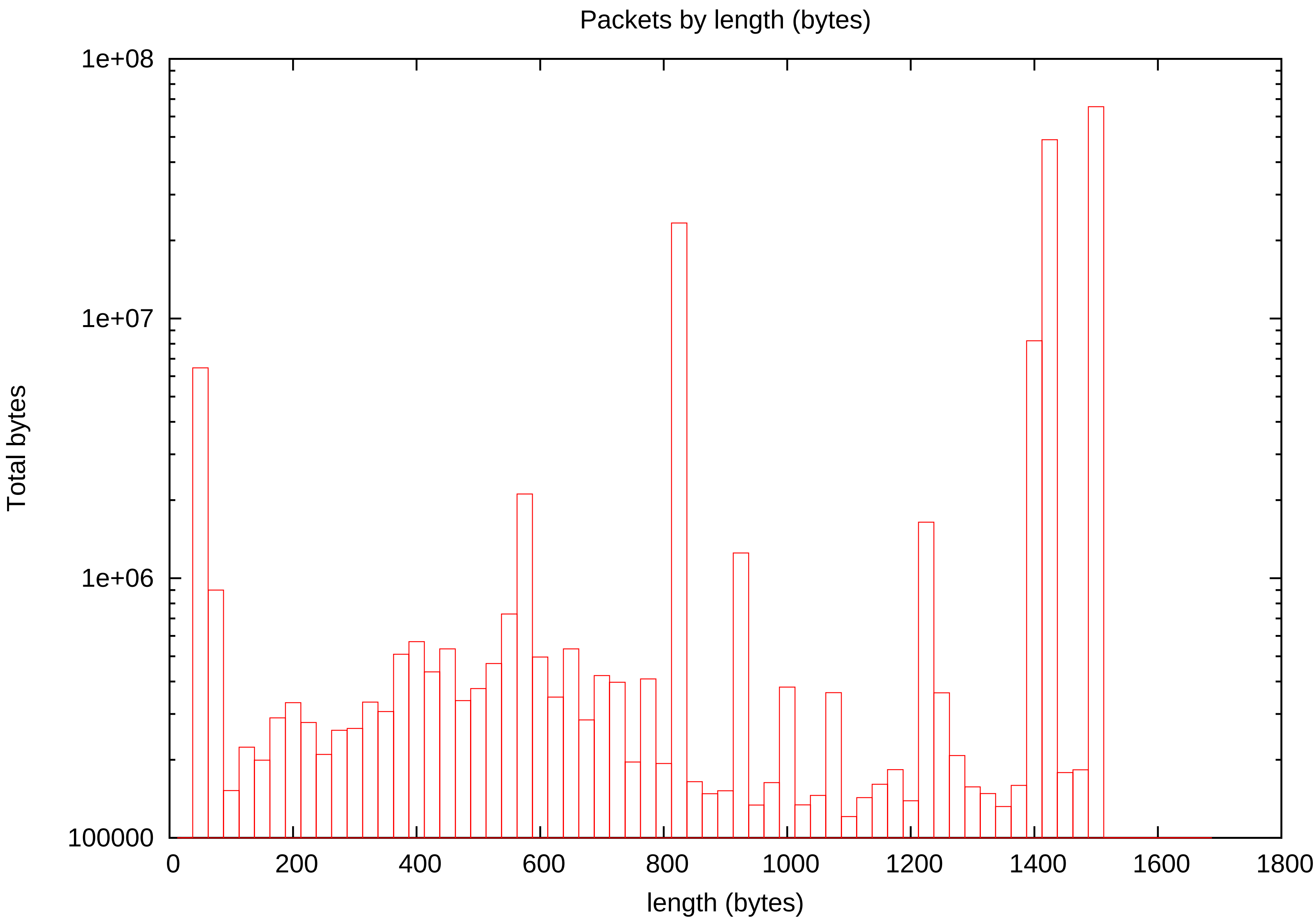
Time differences in data(2)



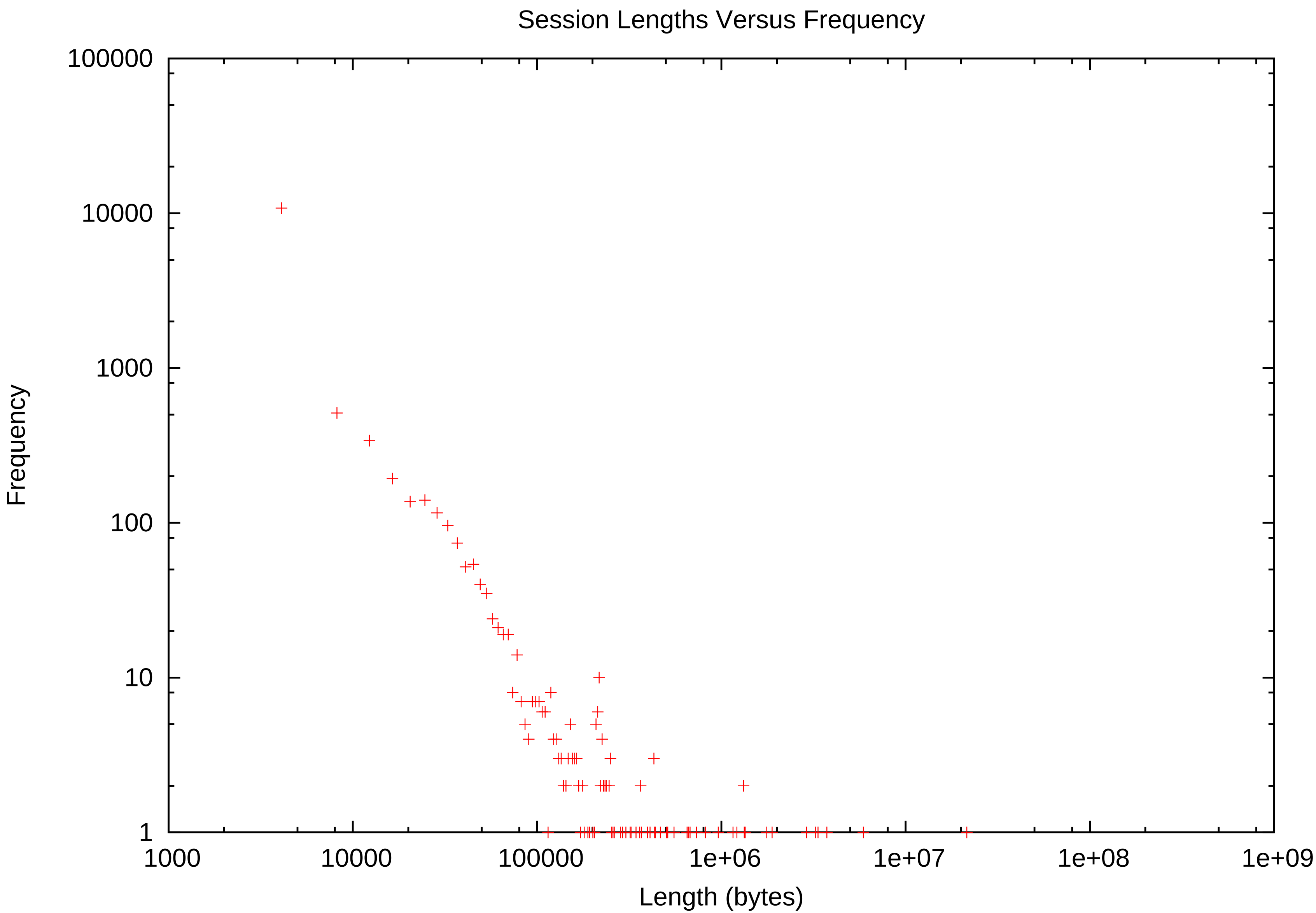
Packet lengths — by no of packets



Packet lengths — by no of bytes



TCP Session Lengths



`pppd` data collection

- The Python Parallel Pairwise Pinging Daemon is a small set of machines which simultaneously perform a `traceroute` (`ping`) to each other pairwise.
- Four or five machines take it in turns to attempt to measure the congestion between each other using `traceroute`.
- Python code will run on each computer and trigger the `traceroute` at the appropriate times.
- This will give us a collection of data which provides both spatial and temporal information about congestion on the internet.
- York, BTExact, Queen Mary (University of London) and Imperial College are all taking part.

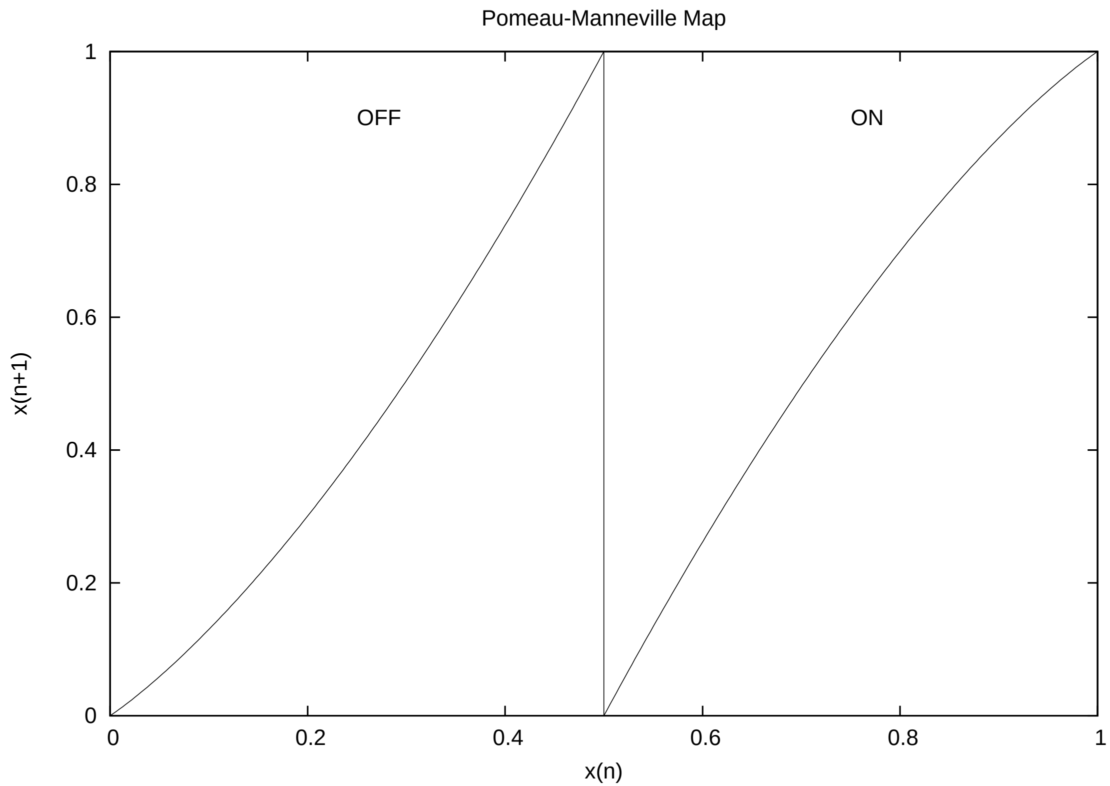
The data set to be collected?

- `lft` (a variant on `traceroute`) collects ping times to all the hops on its journey.
- By triggering this simultaneously from two computers targetting each other we can (hopefully) get a good measure of the intervening congestion.
- We can imagine the time series obtained as a vector representing the congestion at each point in space between the two machines.
- We could use cross correlations to measure the spatial and temporal correlation of the data.
- The data will be made available to other interested researchers in the area.

Markov Model of LRD

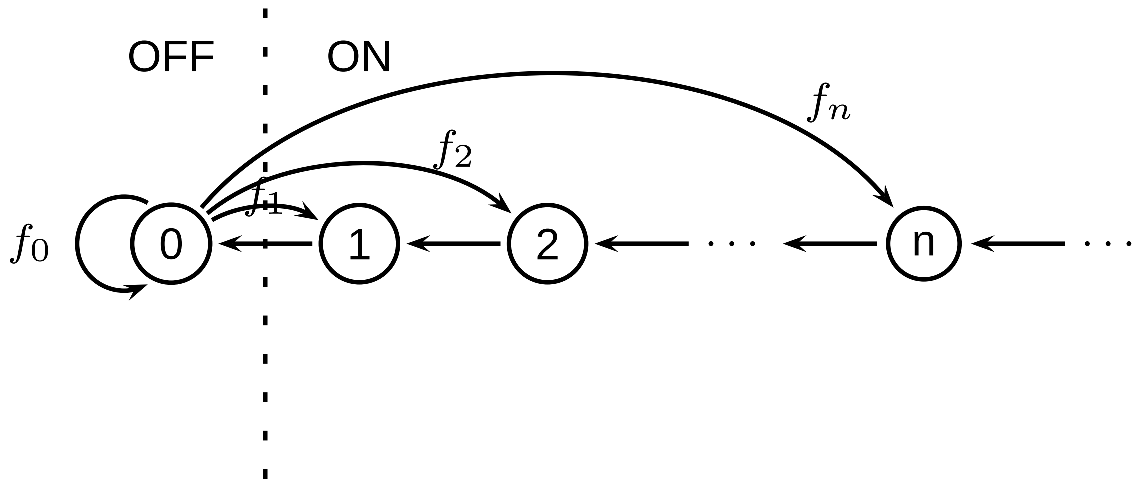
- A number of methods exist for generating LRD data streams.
- The work has its inspiration from work done on linearisation of the Pomeau-Manneville map at QMUL.
- While a finite Markov chain can never exhibit LRD (because the correlations must always die off) an infinite chain can.
- The chain presented here is similar to one investigated by Feller and later Wang but may prove more tractable for the investigation of LRD.
- The map has been developed for computer simulation in such a way that rounding errors are not a problem.
- The computational implementation is extremely efficient in memory and run time.

Intermittency Map For LRD



$$x(n+1) = \begin{cases} x(n) + \frac{1-d}{d^{m_1}} x^{m_1} & 0 \leq x(n) < d \\ x(n) - \frac{d}{(1-d)^{m_2}} (1 - x(n))^{m_2} & d \leq x(n) \leq 1 \end{cases}$$

The Infinite Markov Model



$$\mathbf{P} = \begin{bmatrix} f_0 & f_1 & f_2 & \cdots & f_n & \cdots \\ 1 & 0 & 0 & \cdots & 0 & \cdots \\ 0 & 1 & 0 & \cdots & 0 & \cdots \\ 0 & 0 & 1 & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

Remember f_i is the pr. of transition to i and π_i is the equilibrium pr. of i.

Some Statements (without proof here)

- Notation X_t is state of chain at t but Y_t is the binary process.
- Clearly $\sum_{i=0}^{\infty} f_i = 1$ and $\sum_{i=0}^{\infty} \pi_i = 1$.
- The chain is ergodic if:
  1. $f_0 > 0$ (actually we probably don't NEED this condition),
  2. for all $i \in \mathrm{N}$ there exists $j > i : f_j > 0$,
  3. and $\sum_{i=0}^{\infty} i f_i < \infty$.
- π_0 is the probability $X_t = 0$ (and therefore $Y_t = 0$).
- $1 - \pi_0$ is therefore the mean of the process $Pr(Y_t = 1)$.
- $\pi_i = \pi_{i+1} + f_i \pi_0 = \pi_{i+2} + f_{i+1}\pi_0 + f_i \pi_0 = \ldots$.
- Therefore $\pi_i = \pi_0 \sum_{j=i}^{\infty} f_j$.

Inducing a Correlation Structure

- The ACF R(k) depends on $Pr(Y_{t+k} = 1|Y_t = 1)$.
- Unbroken runs of k 0s will clearly decay exponentially with k. The f_i values set the decay of unbroken runs of 1s.
- Part of a run of k or more if $X_t \geq k$.
- Control decay of $\sum_{i=k}^{\infty} \pi_i$.
- For LRD $\sum_{i=k}^{\infty} \pi_i \sim k^{-\alpha}$ for $k > 0$ [a].
- Strict condition $\sum_{i=k}^{\infty} \pi_i = Ck^{-\alpha}$ for $k > 0$.
- Since $\pi_0 = 1 - \sum_{i=1}^{\infty} \pi_i$ then $C = 1 - \pi_0$.

[a] The actual *proof* is underway... due to work by Feller and Wang.

Generating the Correlation Structure

- This system is trivially solved and we can calculate the values of f_k.
- For $k > 0$ we have (note problems with some values):

$$f_k = \frac{1-\pi_0}{\pi_0}\left[k^{-\alpha} - 2(k+1)^{-\alpha} + (k+2)^{-\alpha}\right]$$

- The attractive thing about this series is that it is telescoping. For example.

$$f_0 = 1 - \sum_{i=1}^{\infty} f_i = 1 - \frac{1-\pi_0}{\pi_0}\left[1 - 2^{-\alpha}\right]$$

Directly Using The Infinite Chain

- We can directly use the infinite chain in calculations if we use a simple algorithm. First define $F(j,k) = \sum_{i=j}^{k} f_i$ where $(j \leq k)$.
- We can see that if $X_t = 0$ then $Pr\{X_{t+1} \in [i,j] | X_t = 0\} = F(j,k)$.
- The telescoping property makes $F(j,k)$ easy to calculate. For $j > 0$ and $k < \infty$ we have:

$$F(j,k) = \frac{1-\pi_0}{\pi_0}\left[j^{-\alpha} - (j+1)^{-\alpha} - (k+1)^{-\alpha} + (k+2)^{-\alpha}\right]$$

- We can also calculate the conditional probability: $Pr\{X_{t+1} \in [i,j] | X_{t+1} \in [k,l] \cap X_t = 0\}$ where $k \leq i$ and $l \geq j$.

Algorithm for N state Finite Chain

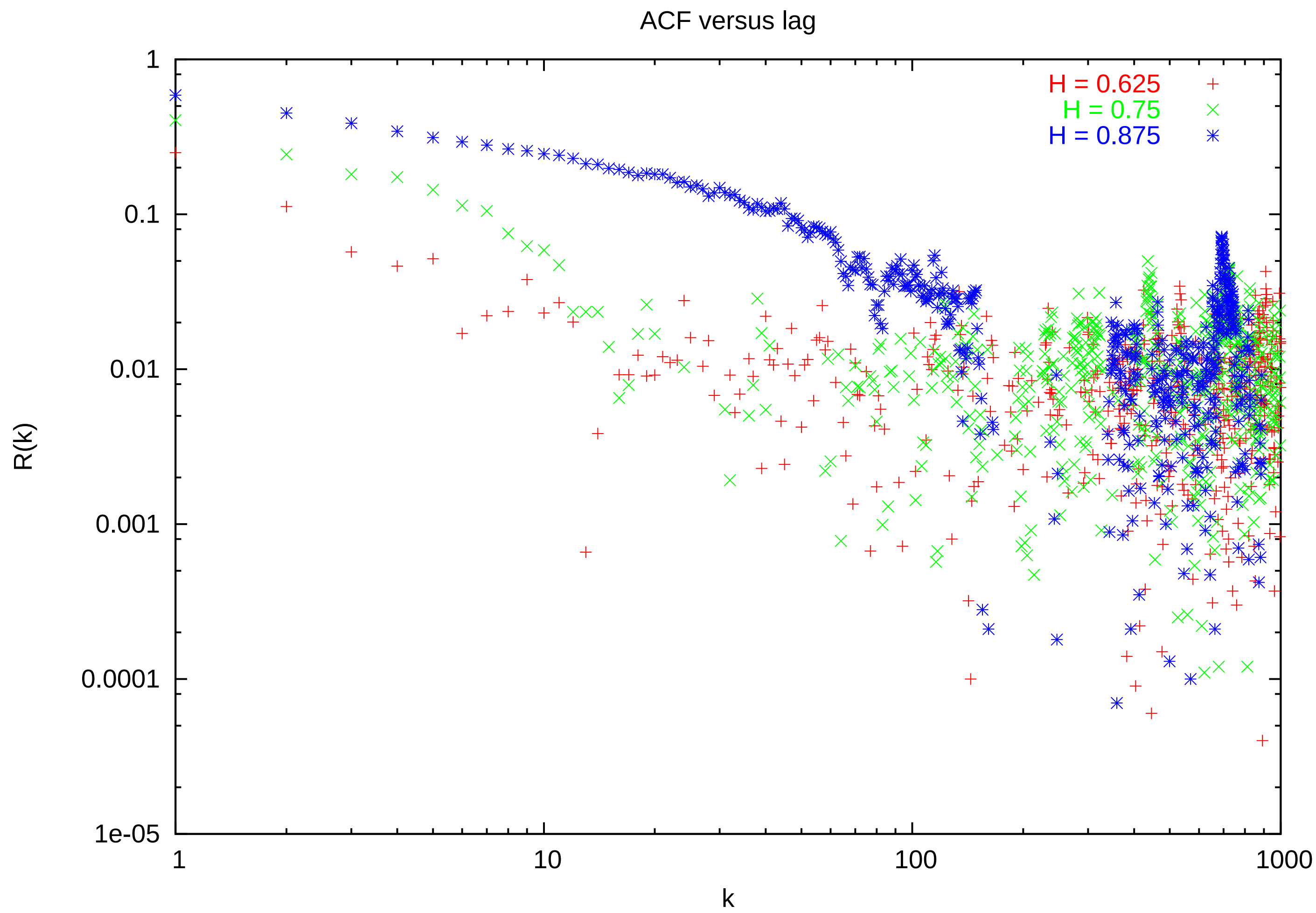
1. If $X_t > 0$ then $X_{t+1} = X_t - 1$. Exit here.
2. Otherwise, choose a new random number $R \in [0, 1]$ with a flat dist.
3. Set $j = 1$.
4. If $R < F(j, N)$ then $X_{t+1} = j - 1$. Exit here.
5. Increase j by 1. If $j > N$ then $X_{t+1} = N$ (or larger). Exit here.
6. Go to step 4.

- But there are considerable problems with rounding errors in a computer.

Algorithm for Infinite Chain

1. If $X_t = 0$, explicitly calculate if $X_{t+1} \in [0, N-1]$ (where N is a small integer) using a single random no as previously.
2. Generate a new random number $R \in [0, 1]$ with a flat dist.
3. Calculate $Pr\{X_{t+1} \in [N, 2N-1] | X_{t+1} \in [N, \infty]\}$ if R is less than or equal to this probability then X_{t+1} is in the required range.
4. If X is in the required range then refine down by generating a new random number and use a binary search until X is found.
5. Otherwise increase the value of N to $2N$ and go to step 2.

ACF from Process



Runs are 10000 aggregated points each generated from 100 points.

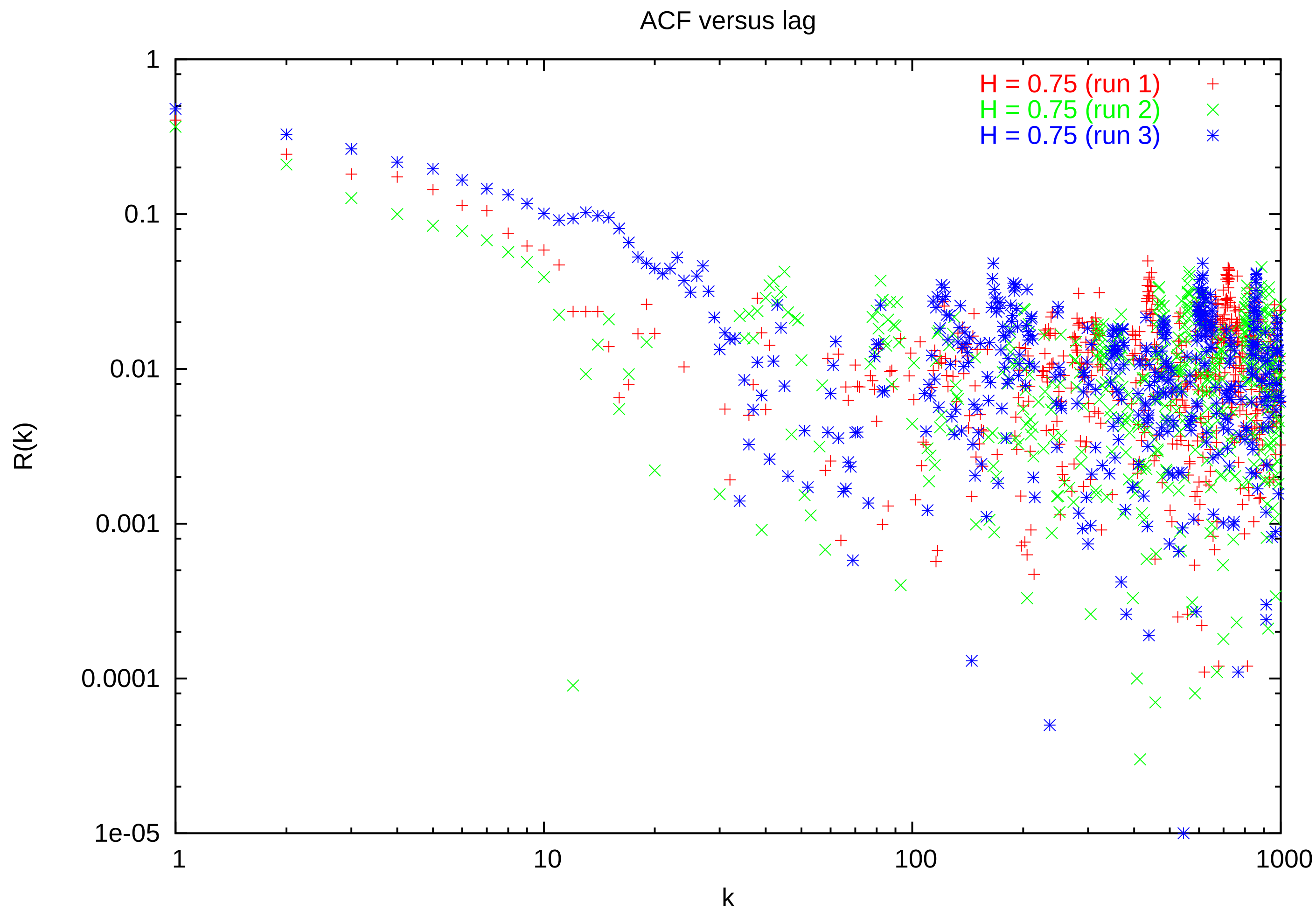
A Major Problem with LRD

- A major problem occurs when dealing with LRD processes — weak convergence.
- This is a trivial but informative example:
- Sample mean of 100,000 points generated by Markov chain mean 0.5 (10 experiments):

  0.4937, 0.5090, 0.4913, 0.5029, 0.4772, 0.4955, 0.5045, 0.4946, 0.4921, 0.4937
- Sample mean of 100,000 random no.s flatly dist. in 0-1, mean 0.5 (10 experiments):

  0.5003, 0.4995, 0.5014, 0.5001, 0.5003, 0.5009, 0.5005, 0.5007, 0.4989, 0.5012
- Computational experiments with LRD suffer very poor repeatability **even with very long runs**.

ACF from Process



Runs are 10000 aggregated points each generated from 100 points.

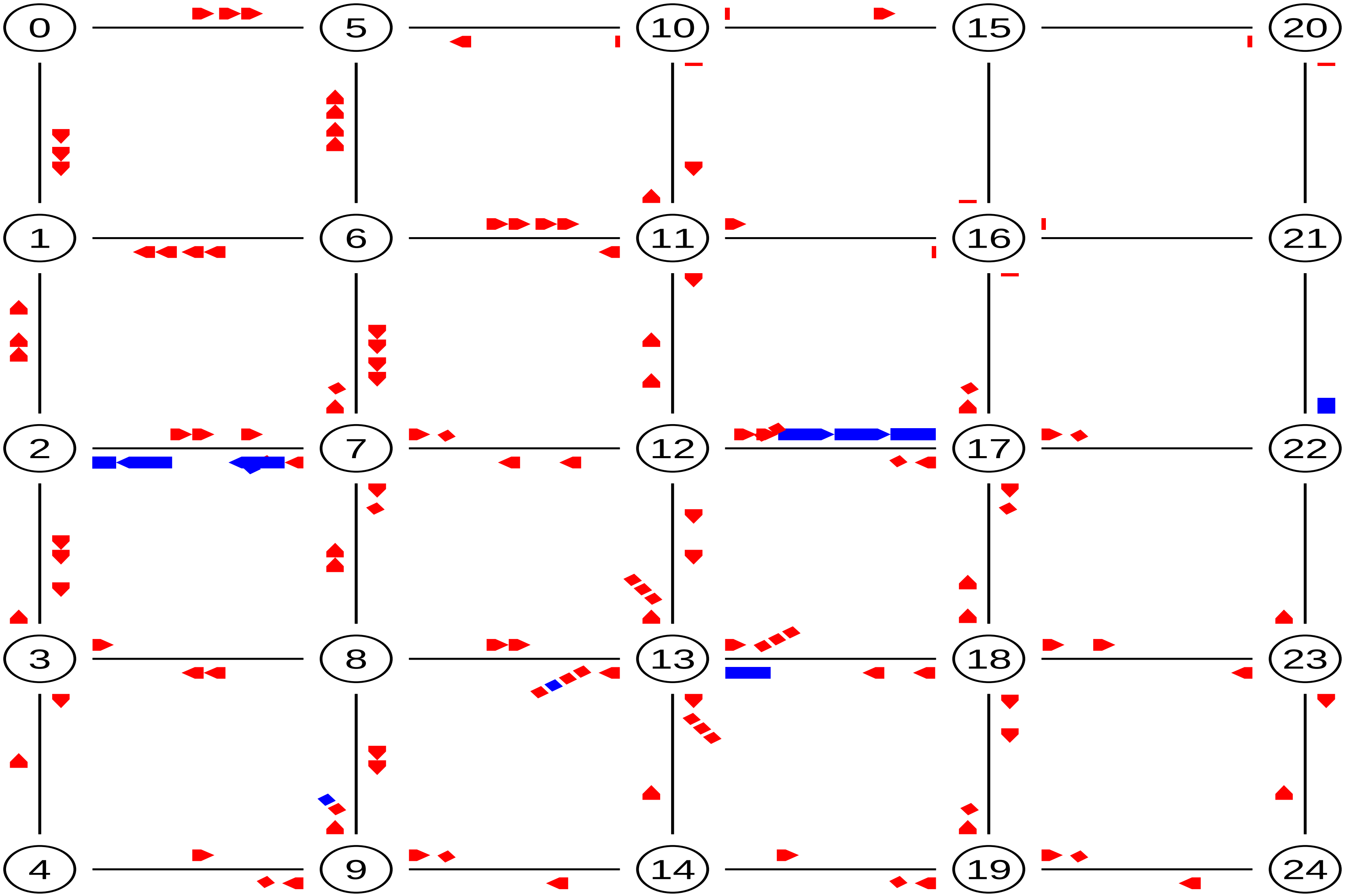
Other Mathematical Modelling

- Padhye et al developed a mathematical model of TCP traffic.
- The model in question estimates bandwidth based upon assumptions about window size, packet loss probabilities etc.
- The model accounts for TCP feedback mechanisms such as timeout and TD ACK.
- An assumption of the model is that p the probability of packet loss is constant.
- Computational work has shown that a number of these sharing the same hardware will cause instability.
- It is hoped to extend this model mathematically rather than computationally.

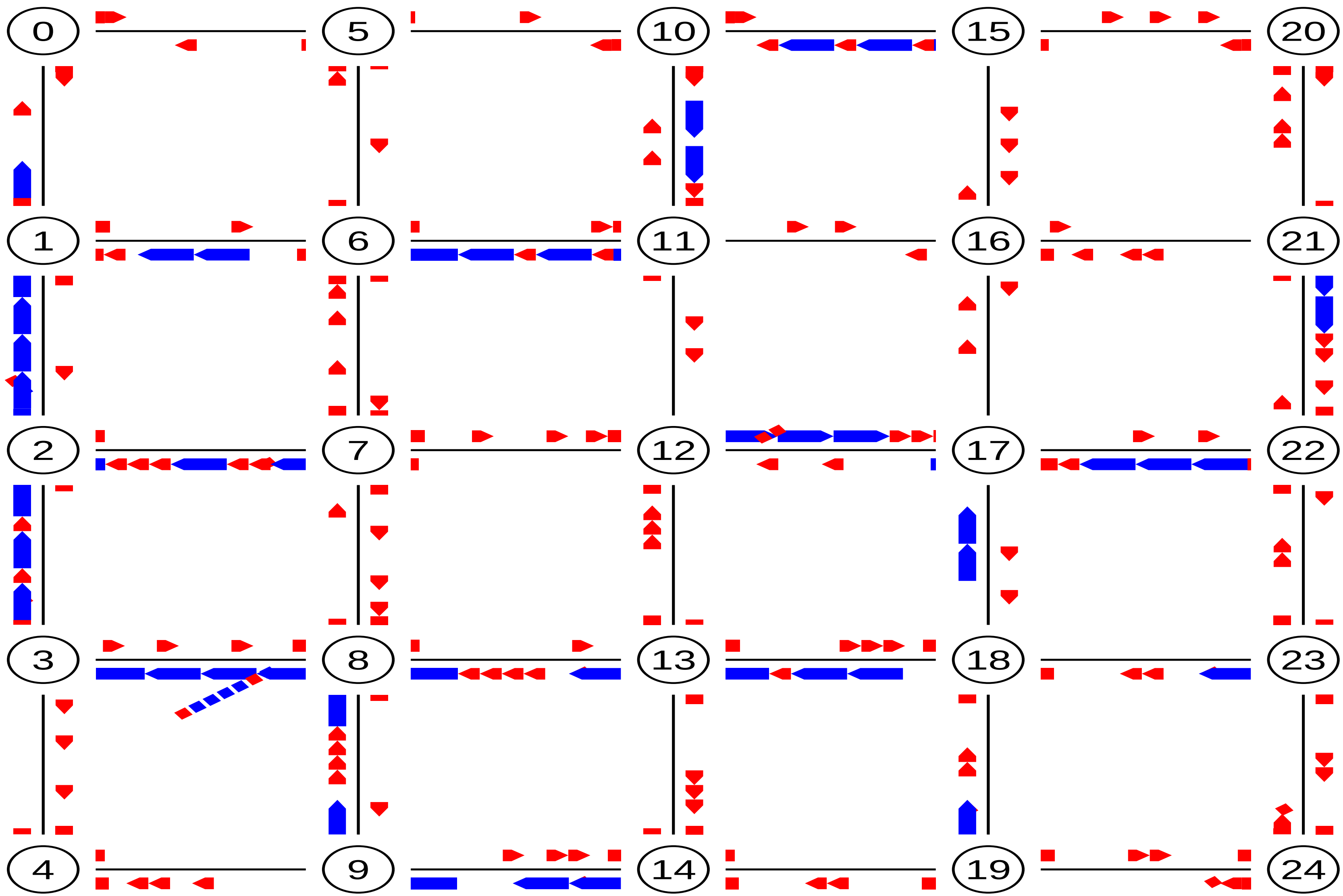
ns modelling

- ns is a freeware packet level simulation of networks.
- `http://www.isi.edu/nsnam/ns/`.
- It is written in C++ and configured with tcl — it is highly versatile although not always easy to use.
- The program has been extended to use the Markov method described as a source.
- Other sources have been added such as the intermittency map above.
- If there is demand, I will do a demonstration of ns and its capabilities as part of a subsequent meeting at York.

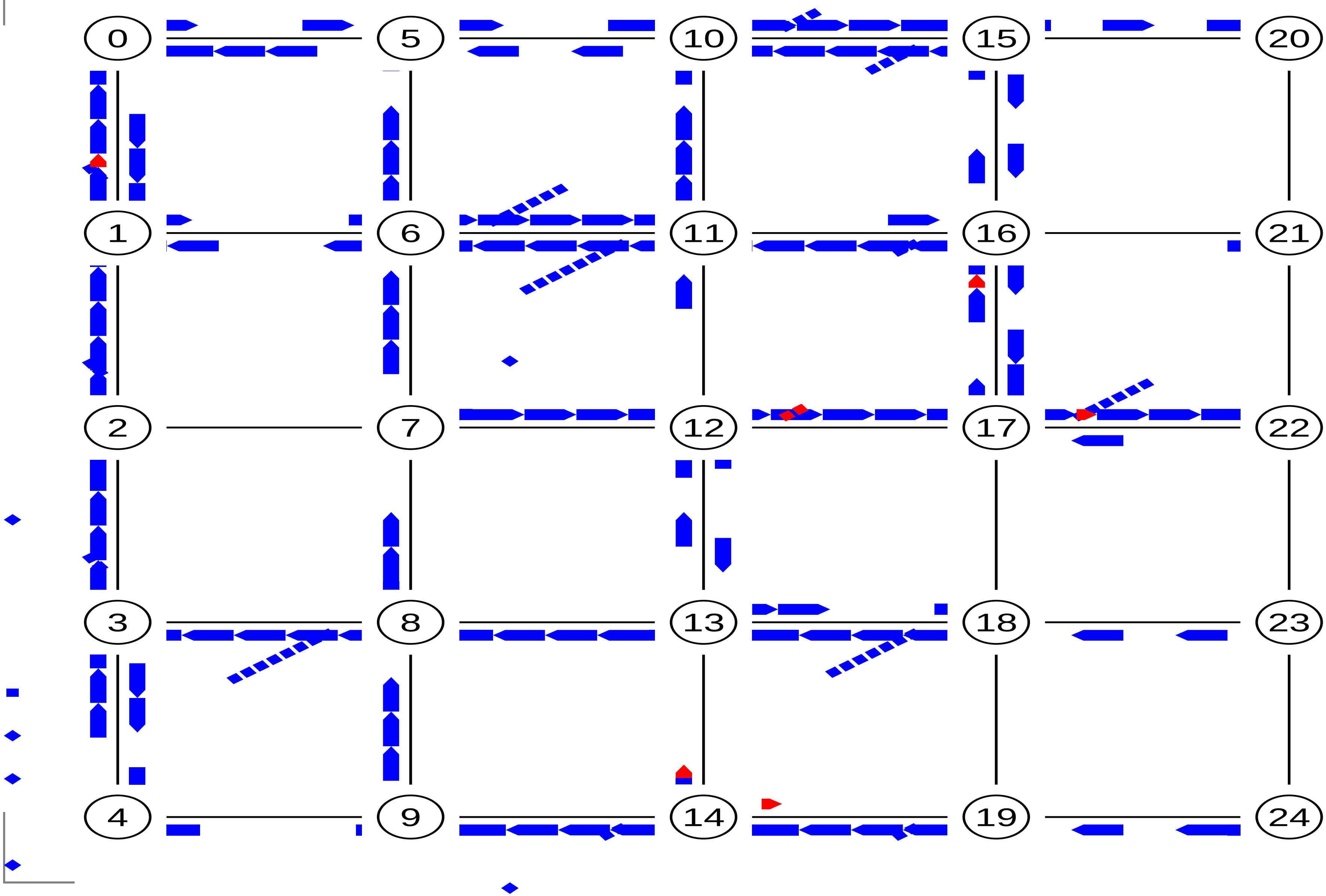
Simulation - Early network



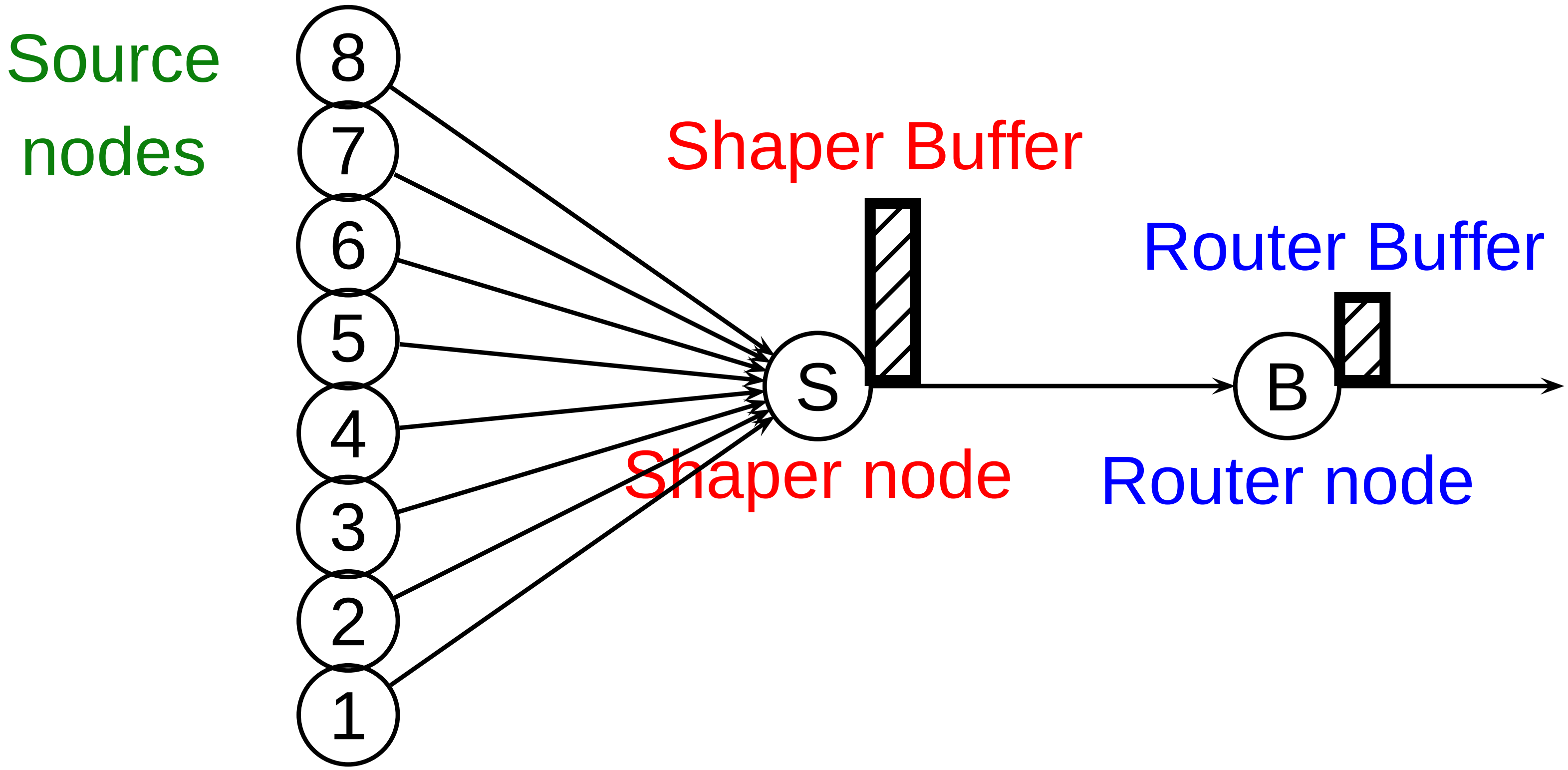
Simulation - Getting set up



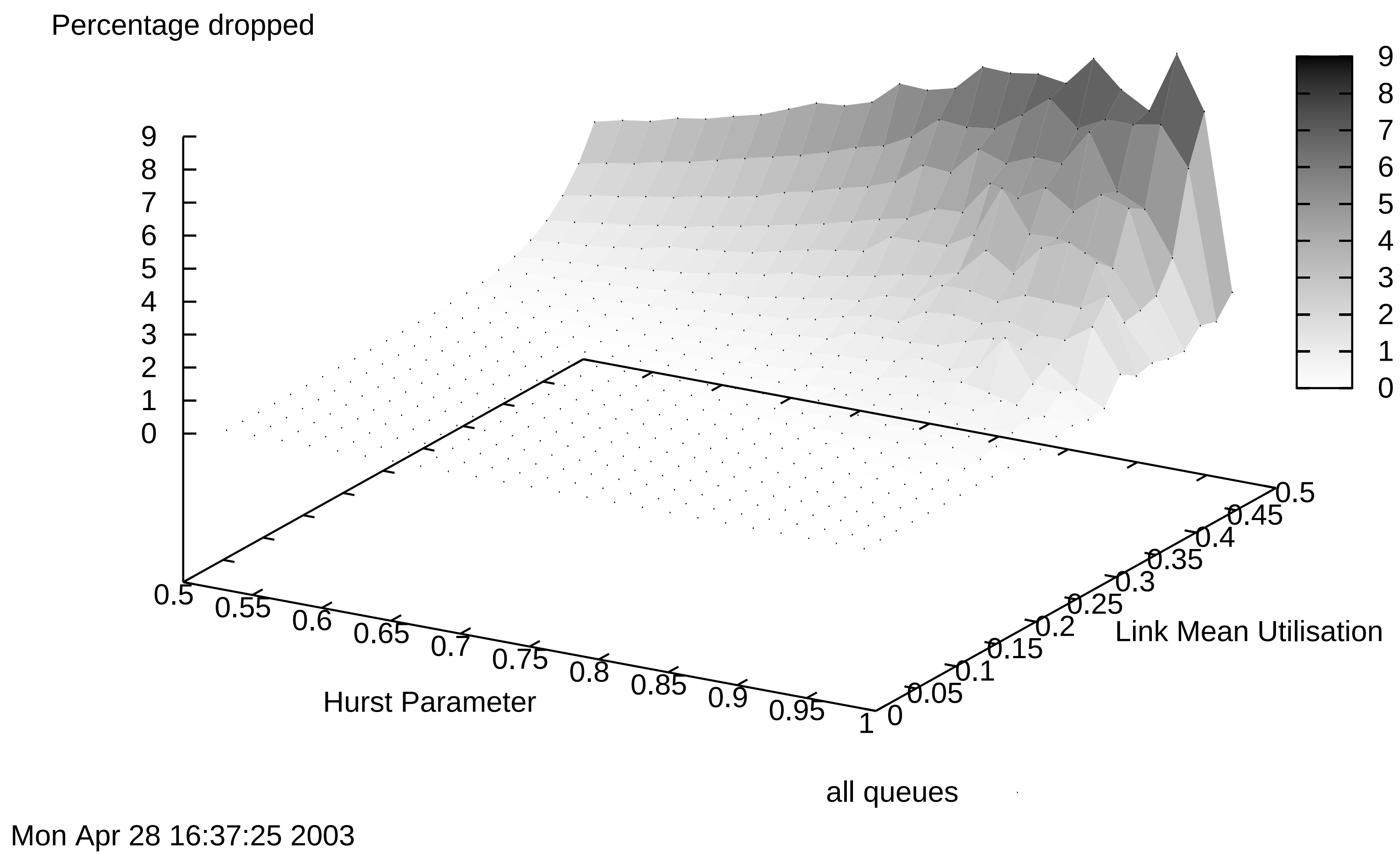
Simulation - Running



Modelled Scenario



Packet Loss at all Queues



Conclusions

- The project is at too early a stage to make anything but tentative conclusions.
- UDP traffic is an extremely small contributor to the York network overall. It would require additional explanation for this to be a main cause.
- A considerable amount of TCP traffic is not part of a stream or part of only short streams.
- The need for a robust measure of the Hurst parameter is important in the project at this stage.
- Computational experiments on LRD are difficult due to the nature of the phenomenon.
- Some useful data is being generated which is available on request and should be of use to researchers in the area.