The statistics of intermittency maps and dynamical modelling of networks

D.K. Arrowsmith[*], M. Barenco [†], R.J. Mondragon[‡], M. Woolf[*]

July 12, 2004

Abstract

The paper is split into two parts. The first indicates the form of the proof of an autocorrelation result for a double intermittency map of the interval. The specific map considered is a piece-linear map of the interval. The map is then used in the second part of the paper for modelling sources of long-range dependent traffic in a network. The main ingredients are the production of packets of data at the nodes of a network and the transfer of the data around the network. The behaviour of packet lifetime and throughput is compared for both closed and open loop models, and also long-range-dependent Poisson-like traffic sources.

1 Introduction

In the early 1990's, Leland and coworkers [17] showed that packet traffic in computer networks exhibited *long range dependence* when it was previously thought that data traffic packet rates, like voice traffic, would be memoryless. This has major consequences as *long-range-traffic* is also *bursty* and poses major engineering challenges, at timescales spanning several orders of magnitude.

In order to to address these challenges, it is necessary to have models which can mimic packet traffic behaviour in networks. The first deterministic model which addressed this was proposed by Erramilli *et al* [9]. It is a one-dimensional chaotic map f defined on the unit interval by the family of Erramilli interval maps used as the basis for each *LRD* traffic source in the network.

$$f(x) = \begin{cases} x + (1-d)\,(x/d)^{m_1}\,, & x \in [0,d], \\ x - d\,((1-x)/(1-d))^{m_2}\,, & x \in (d,1], \end{cases} \tag{1}$$

where $d \in (0,1)$. The parameters $m_1, m_2 \in [3/2, 2]$ induce *map intermittency* at the two points $x = 0$ and $x = 1$ with different intermittency values.

A graph of the map is shown in Fig 1(a). To produce traffic traces, a random seed $x_0 \in [0,1]$ is used and iterated to obtain a sequence $\{x_n\}$. Every time the orbit visits the interval $(d,1]$, a packet (or '1') is produced, otherwise a '0' is produced.

The map has the following properties:

- the chaotic nature of the map guarantees the pseudo-randomness of the binary sequences it produces so that no stochastic term needs to be introduced into the model.
- the map has two non-hyperbolic unstable fixed points which allows the *LRD* output to be controlled, although only only one such fixed point is needed to obtain *LRD* output.

[*]School of Mathematical Sciences, Queen Mary, University of London, London, E1 4NS, UK.
[†]CoMPLEX, University College, London, UK.
[‡]Electronic Engineering, Queen Mary, University of London, E1 4NS, UK.

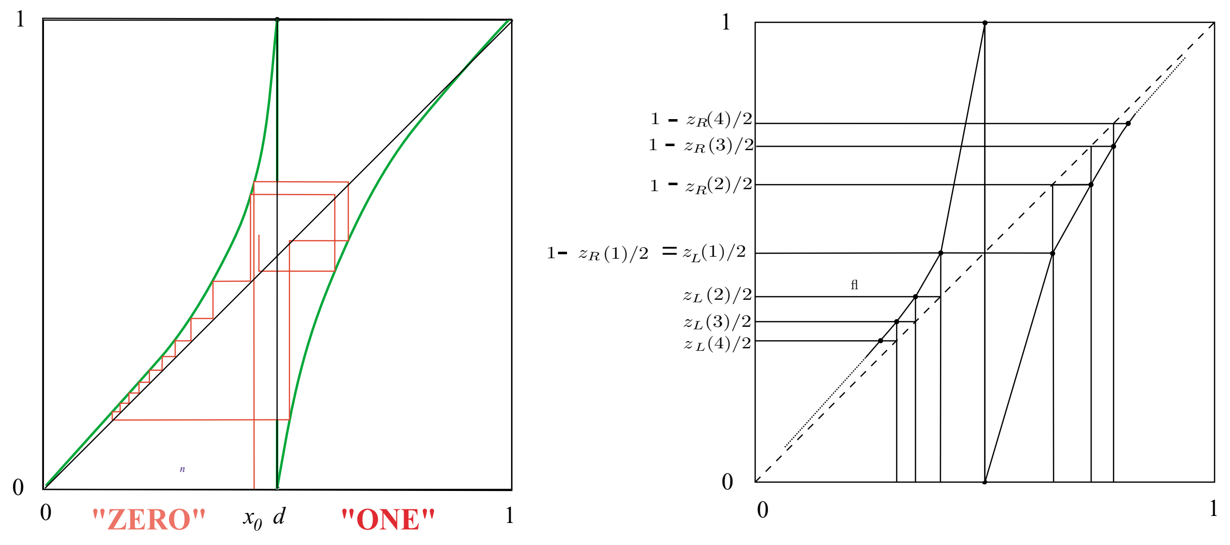

Figure 1: (a) The piece-wise smooth graph f with intermittency at $x = 0$ and $x = 1$. (b) The corresponding piece-wise linear graph g.

Erramilli [10] encountered great difficulties in obtaining analytical results for the map. In particular, there is no closed form for the invariant density - the distribution of points of a typical orbit on [0,1]. Thus, in particular there is no closed form formula for the natural invariant density, and so the only approach to calculations is numerical. The difficulties are exacerbated by the non-linear nature of the component curves in eqn. (1). The nonlinearity causes the successive lengths of spells of the same symbol to be statistically dependent on each other. The latter problem was solved by Mondragón [19], by introducing a 'random wall' between the two branches of the graph of f, avoiding some of the problems associated with the absence of a closed form for the invariant density. By slightly modifying the piece-wise smooth function f to a piece-wise linear function p an invariant density can be easily determined, [5]. The approach was first proposed the 'Chaos Engineering Group in Ferrera/Bologna [15, 16].

2 Traffic maps

2.1 Piece-wise linear map

The map p is constructed from two positive decreasing sequences z_i^L and z_i^R which converge to zero. Both sequences are set so that $z_1^L = z_1^R = 1$ and the sums $\sum_{i=1}^{\infty} z_i^* = S^*$, where the symbol * is a wildcard which denotes either L or R, are assumed finite. The unit interval is now partitioned into the subintervals

$$J_i^L = (z_{i+1}^L/2, z_i^L/2], \;\; J_i^R = (1 - z_i^R/2, 1 - z_{i+1}^R/2], \tag{2}$$

$i = 1, 2 \ldots$. The map is determined by requiring f to be affine on each sub-interval, so that

$$p(J_i^*) = J_{i-1}^*, \;\; i = 2, 3 \ldots$$

$$p(J_i^*) = \bigcup_{i=1}^{\infty} J_i^{\bar{*}}$$

where $\bar{*} = R$ if $* = L$ and vice-versa. To satisfy these conditions, the map $p : [0,1] \to [0,1]$ is given by

$$p(x) = \begin{cases} 0 & \text{if } x = 0, \\ \frac{\Delta^L i-1}{\Delta_i^L}(x - \frac{z_{i+1}^L}{2}) + \frac{z_i^L}{2} & \text{if } x \in J_i^L \\ \frac{\Delta^R i-1}{\Delta_i^R}(x - 1 + \frac{z_{i+1}^R}{2}) + 1 - \frac{z_i^R}{2} & \text{if } x \in J_i^R \\ 1 & \text{if } x = 1, \end{cases} \tag{3}$$

where $\Delta_i^* = z_i^* - z_{i+1}^*$.

In practice, the numerical iteration of the map does not require the application of the formulae above, as orbital points retain their relative position in the intervals J_i^* as they are iterated by p. Computation has to be carried out only when iterations move the orbit from one interval to the other. The graph of the map p is given in Fig. 1(b). By construction, it is isomorphic to a *Markov* chain. The state i^* in the Markov chain corresponds to the interval J_i^* in the map. If Y_t is the state at time t, the non-zero transition probabilities in the Markov chain are given by

$$Prob(Y_{t+1} = *_i | Y_t = *_{i+1}) = 1,$$

$$Prob(Y_{t+1} = *_i | Y_t = \bar{*}_1) = \Delta_i^*.$$

2.2 Invariant measure

The probability of the image of the map $p|J_1^*$ to land in the i-th interval $J_i^{\bar{*}}$ is $\Delta_i^{\bar{*}}$. The expected sojourn times are from the '0' and '1' regions:

$$\sum_{i=1}^{\infty} i\Delta_i^L = \sum_{i=1}^{\infty} z_i^L = S_L,$$

and

$$\sum_{i=1}^{\infty} i\Delta_i^R = \sum_{i=1}^{\infty} z_i^R = S_R.$$

The expected number of packets per unit time (or average load) μ is

$$\mu = \frac{S_R}{S_L + S_R}.$$

The expected *cycle length* is the total sojourn time $S_L + S_R$. This gives that the probability P_1 of visiting the interval J_1^L is $P_1 = 1/(S_L + S_R)$. The probability for visiting J_1^R is also P_1. We can then use

$$Prob(x \in J_i^*) = P(Y_t = *_i) = P_1 z_i^*, i = 1, 2 \ldots.$$

These probabilities give the stationary probability of the Markov chain as well as the invariant measure of the map ϕ which is constant on each subinterval J_i^*. The corresponding equivariant density ϕ is defined by

$$\phi(x) = \frac{P_1 z_i^*}{\Delta_i^*}, x \in J_i^*$$

on each interval J_i^* for the map associated with a given subinterval by its width.

2.3 Exact correlation

The auto-correlation $\gamma(k)$, where k is the lag, computed on the binary variable X_t is

$$\gamma(k) = \frac{E(X_t X_{t+k}) - E(X_t)E(X_{t+k})}{\sqrt{\mathrm{Var}(X_t)\mathrm{Var}(X_{t+k})}}.$$

Since $X_t = X_t^2$, we have $E(X_t^2) = E(X_t) = \mu$. It follows that

$$\gamma(k) = \frac{E(X_t X_{t+k}) - \mu^2}{\mu(1-\mu)}$$

and the term of interest is clearly $E(X_t X_{t+k})$. The series X_t is binary and so the only non-trivial contribution is when $X_t = X_{t+k} = 1$. Thus, the uncentered covariance is

$$Prob(X_{t+k} = 1 | X_t = 1).$$

There are various recursions between different collections of symbol sequence types:

$$T_{01}(n,1) = \sum_{i=1}^{n-1} \Delta_i^L \Delta_{n-i}^R \tag{4}$$

$$T_{01}(n) = \sum_{i=2}^{n-2} T_{01}(i,1) T_{01}(n-i) \tag{5}$$

$$T_{00}(n) = \Delta_n^L + \sum_{i=2}^{n-1} T_{01}(i) \Delta_{n-i}^L \tag{6}$$

$$Z(n) = P_1 \sum_{i=1}^{n-1} z_i^R z_{n-i}^R \tag{7}$$

$$P_{11}(n) = P_1 \sum_{i=n}^{\infty} z_i^R \tag{8}$$

$$C(n) = P_{11}(n) + \sum_{i=1}^{n-2} T_{00}(i) Z(n-i). \tag{9}$$

where $T_{01}(n,1)$ is the total transition property associated with all transition sequences of length n, starting with a '0' and finishing with a '1' with a single uninterrupted spell of either symbol; $T_{01}(n)(T_{00}(n))$ corresponds to *all* transition sequences of length n from a '0' to a '1' ('0' to a '0'); $Z(n)$ is the probability associated with the extremities of a sequence whereby the cumulative length of those two uninterrupted spells of '1's is equal to n; $P_{11}(n)$ is simply the probability of having a sequence of '1's of length n.

The number of operations is now of length $O(n^2)$ rather than a straight count of the subset of the 2^{n-1} sequences that satisfy $X_t X_{t+n} = 1$ which is $O(2^n)$.

2.4 Intermittency

To create intermittency we need to choose sequences $\{z_i^L\}$ and $\{z_i^R\}$ with power-law decay to zero. This is equivalent to choosing a power law tangency for the map in eqn.(1). The case where just one sequence has the required decay was considered by [21] and [27]. We let $z_i^* \sim K_* i^{-\alpha}$ which gives $\Delta_i^* \sim K_* i^{-(\alpha_*+1)}$. There is a link between the continuous parameters m_1 and m_2 and the PL case with α_L and α_R by choosing

$$\alpha_* = \frac{1}{m_* - 1}.$$

It can be shown that the autocorrelation decays are polynomial with respect to the lag and that the rate of decay (the exponent) depends solely on the minimum value of α_1 and α_2. This minimum property arises from having to use the following result:

if two sequences satisfy $a_i \sim K_a i^{-\alpha}$ and $b_i \sim K_b i^{-\beta}$ for $K_a, K_b > 0$ then the convolution

$$C_m = \sum_{k=1}^{m-1} a_k b_{m-k}$$

has polynomial asymptotics given by

$$C_m \sim \begin{cases} K_a S_b m^{-\alpha} & \text{if} \quad \alpha < \beta \\ K_b S_a m^{-\beta} & \text{if} \quad \alpha > \beta \\ (K_a S_b + K_b S_a) m^{-\alpha} & \text{if} \quad \alpha = \beta \end{cases} \tag{10}$$

where $S_a = \sum_{i=1}^{\infty} a_i$ and $S_b = \sum_{i=1}^{\infty} b_i$. It should be noted that convolutions of this type will arise in the six recurrence relations. The following theorem can be proven (see [5] for more details):
Theorem The autocorrelation $\gamma(k)$, for lag(k), computed on the binary traces produced by the piece-wise linear map p satisfies

$$\gamma(k) \sim K k^c,$$

where the exponent is given by

$$c = 1 - \text{Min}(\alpha_L, \alpha_R).$$

and where the contant K can be obtained explicitly.

3 Packet traffic modelling on networks

3.1 Topology of networks

Many different topologies appear in communication networks. Square lattices, toroidal lattices, meshes and hypercubes arise on multiprocessor computers (e.g. [18]), scale-free networks in the WWW [1] and the Internet [11]. The way that the elements of the network are connected to each other and the nodal degree properties have an impact on its functionality. The representation and study of the connectivity of a network is carried out using concepts from graph theory.

A communications network can be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (vertices) and $\mathcal{E}$ is the set of links (edges). The hosts, routers and switches are represented by nodes and the physical connections between them are represented by links. The links can have a direction, but here we are only going to consider undirected links. A node can transfer information to another node in the form of data-packets if there is a link between them. If there is no direct link between the nodes, then a path in the network is the sequence of distinct nodes visited when transferring data packets from one node to another. We consider networks where there exists at least one path connecting any pair of nodes of the network.

The degree, k, of a node is the number of links which have the node as an end-point. The degree of a node is a local quantity. However, the node degree distribution of the entire network gives important information about the global properties of a network and can be used to characterise different network topologies.

3.2 Scale-free Networks

Many technological networks are not described by a random or a regular network; instead they are better described by a network where the degree distribution is described by a power law where

$$P(k) \sim C k^{-\beta}, \tag{11}$$

for $\beta > 1$ and C constant. The probability that a node has k edges connected to it is given by $P(k)$. In practical terms a power law distribution means that the majority of the nodes will have very few neighbours, but there is a very small set of the nodes with a very large number of neighbours. Networks with this property are known as *scale-free* because power-laws are free of a characteristic scale, that is, there is no characteristic node degree.

The diameter of a scale-free network scales as $\ln \ln S$ [7], where S is the size of the network. This is due to the existence of "far–reaching" links which are shortcuts when going from a source to a destination. Any node that contains one of these far-reaching links is going to be highly used when transferring packet-data.

In 1999, Barabási and Albert [4] showed that it is possible to create a scale-free network by using two generic mechanisms: *growth*, the network grows by attaching a new node with m links to m different nodes present in the network; and *preferential attachment*, where new nodes are attached preferentially to nodes that are already well connected. Barabási and Albert showed further that if the probability that a new node will be connected to node i with degree k_i is

$$\Pi(i) = \frac{k_i}{\sum_j k_j}. \tag{12}$$

then the network has a power law link distribution $P(k) \propto k^{-3}$.

3.3 Model of Networked Data Traffic

The model considered here has been studied by several authors [20, 26, 28, 12, 14, 22, 3]. The network consists of two types of nodes; router nodes that store and forward packets; host nodes that store and forward packets and are also sources and sinks of traffic. Given the network has S nodes, and a density $\rho \in [0, 1]$ of hosts then ρS is the number of total hosts. The host nodes are randomly distributed in the network.

- Traffic generation: A host creates a packet whose destination is another host. A host creates a packet using either a uniform random distribution (Poisson) or a *LRD* distribution defined by a chaotic map. Each source generates its traffic independently of the other sources.

- Queue: Each node keeps a queue of unlimited length where the packets are stored. Any packet that is generated is put at the end of the host's queue. If a packet arrives at a router it is put at the end of the hierarchical router's queue. The packet is not queued when it arrives at its destination node.

- Routing: Each node picks a packet at the head of the queue and forwards the packet to the next node. From the source/destination information that each packet carries, the forwarding is done by one of the following routing algorithms:

  1. For regular networks:
     - a neighbour closest to the destination node is selected
     - if more than one neighbour is at the minimum distance from the destination, the link to which the smallest number of packets has been forwarded is selected.
     - if more than one of these link shares the same minimum number of packets forwarded then a random selection is used.
  2. for general networks:
     - from the state of the queues and the number of hops from source to destination, the routing is done by minimising the time–delay by considering the number of hops and the size of the queues that a packet visits when crossing the network.

The process of packet generation, hop movement, queue movement and updating of the routing table occurs at one time step (see figure 2).

3.4 Transmission Control Protocol Dynamics

The dynamics of the packet production model can be extended to incorporate packet window dynamics [10]. If the map is in the '*ON*' state, each iteration of the map represents a packet generated. One sojourn period in the '*ON*' side of the map represents a whole file. These *files* are then *windowed* using the *slow-start* algorithm, adding another dynamical layer to the system. The algorithm is as follows:

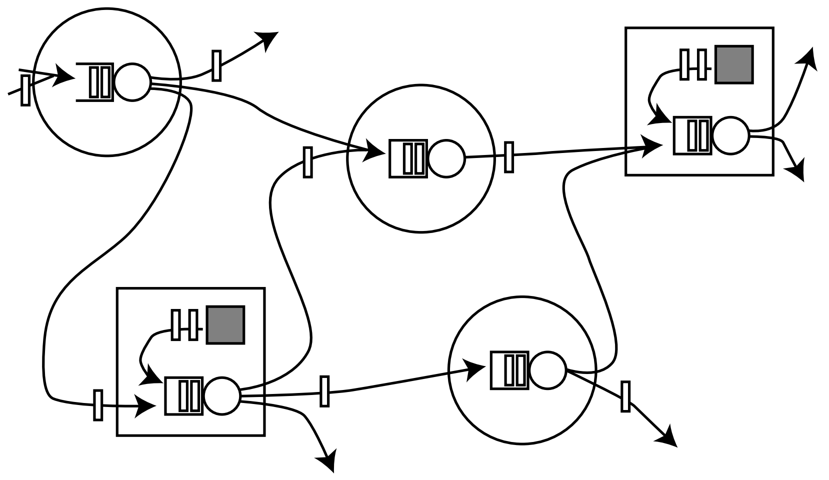

Figure 2: The hosts, represented by large squares, are sources and sinks of traffic. The traffic source is represented by a small grey rectangle. The routers, large circles, only route traffic. Both hosts and routers have a queue where traffic can be stored during its transit through the network.

At a given host i in the network, and time $t = n$, there is a current state, $x_i(n)$, and a current window size, $w_i(n)$, for the number of packets that can be sent at time $t = n$. The window size has a maximum value w_{max}. There is also a residual file size, $s_i(n)$, at node i which is given by the number of iterates of f such that

$$f^{s_i(n)}(x_i(n)) > d,$$

and $f^{s_i(n)+1}(x_i(n)) < d$. The source will send $p_i(n) = \min\{w_i(n), s_i(n)\}$ packets. The full dynamics therefore takes the form, (see Erramilli *et al.*:
For $x(n) > d$, i.e. packet generated-

$$w_i(n+1) = \begin{cases} 1, & \text{if } x_i(n-1) > d, \\ \min\{2w_i(n),\, w_{max}\}, & \text{otherwise,} \end{cases} \tag{13}$$

and $x_i(n+1) = f^{p_i(n)}(x_i(n))$.

For $x_i(n) < d$, i.e. no packet generated - $w_i(n+1) = 0$, and $x_i(n+1) = f(x_i(n))$.

This algorithm applies if all packets in a window are acknowledged before the retransmission timeout (RTO) limit is reached. If packets take longer than this to be acknowledged the window of packets is sent again with the RTO doubled and the window size set to zero. When the map is '*OFF*' the window size is zero and no packets are sent.

This initial value of RTO is calculated using the exponential averaging method [23]. This method keeps a running average of all round trip times. This average is weighted towards more recent round trips, and is used in calculating the RTO.

4 Control

4.1 Control of queue sizes

The simplest way to control packet traffic is to limit the length of queues [2]. As grid bar charts of node queue size have shown by Woolf *et al.* [29], long queues in the network invariably occur at hosts. For this reason it was decided to reduce the rate of packet production at hosts with long queues. The simulation keeps count of packets produced, so the actual or *carried* load is known. Knowledge of the d value being used for the map f gives the maximum rate of packet production, or *offered* load.

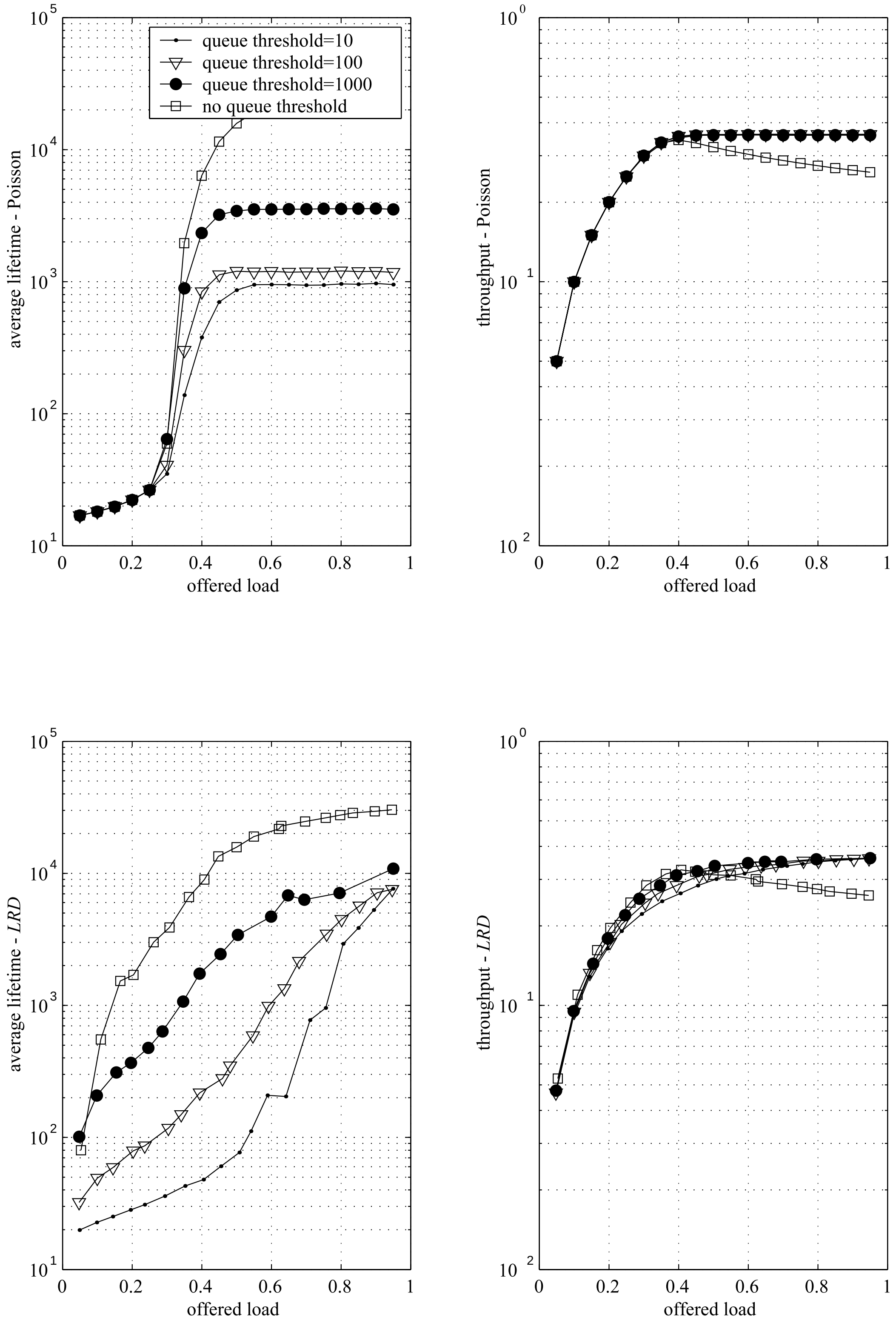

Figure 3: Plots showing the behaviour of the network parameters average lifetime and throughput as carried load is varied when control is applied to a network containing either all Poisson or all *LRD* sources. The network has $S = 164$

Figs 3(a) and 3(b) show average lifetime and throughput plotted against the offered load. The data apply to a 32×32 node rectangular network with a host density 0.16 and Poisson-like traffic sources. A comparison is made between simulations with and without a queue limit. There is little difference between the two cases below a carried load of $\lambda = 0.3$. Average lifetime and throughput are very similar.

Average queue lengths are closely linked to average lifetime. As has been shown previously, this is the point at which congestion sets in. Above this point average lifetimes and host queue lengths are much lower in the controlled case. In the case of host queue length this would be expected, because when packets are created at a host they are immediately added to the queue for that host. The queue limit device prevents longer queues from building up. This also explains the longer average lifetimes. Long host queues no longer exist, so host packets are not delayed so long and average latency times are therefore lower. The average router queue length is longer in the controlled case because there is no limit on router queue lengths, so that packets that would have been in host queues become distributed over router queues instead. The effect on throughput is much smaller, but there is a slight increase when control is used. It should be noted that the plots extend beyond a carried load of about $\lambda = 0.36$, which is the maximum load with queue limiting. In a real network this would equate to bandwidth being traded or a reduced end-to-end delay.

Figs 3(c) and 3(d) show similar plots for the same network with *LRD* sources. In this case the average lifetime is very much lower in the controlled case for the whole range of carried load values. As before, the average queue lengths are linked closely to average lifetime. We have seen in the last section that, in the case of *LRD* sources, congestion starts at very low values of load. The queue limiting prevents this early onset of congestion, modifying the system to behave as the same network with Poisson sources would. This can be seen in the plots of average lifetime. Again, higher queue thresholds have less influence. The throughput is slightly higher in the controlled case at high values of the load. As before, the highest possible load when control is used is about $\lambda = 0.36$.

Plot of average lifetime versus offered load (rather than carried load) also show that in the controlled case average lifetimes remain low even when they would become very high in the uncontrolled case. The same applies to average host queue sizes. In effect, the control mechanism prevents the network from becoming congested, no matter how high the offered load, but at the expense of restricting the carried load to one that the network can manage. Similar conclusions can be drawn for *LRD* sources.

4.2 A local control mechanism

Valverde and Solé [26] introduced a control mechnism where hosts modify their rates of packet release depending on the detected *local* rate of congestion. The sources adjust their local packet release by querying how busy their neighbours are. The number of congested neighbours a node can have is

$$\zeta = \sum_{v \in \mathcal{G}} \theta(n_v(t)) \tag{14}$$

where $\mathcal{G}$ is the set containing the neighbours of node, $n_v(t)$ is the number of packets in node v at time t and θ is the Heaviside function $\theta(x > 0) = 1$ and $\theta(x \leq 0) = 0$. The control mechanism consists of adjusting the rate of packet production $\lambda_i(t)$ using

$$\lambda_i(t+1) = \begin{cases} \min\{1, \lambda_i(t) + \mu\}, & \zeta < 4 \\ 0 & \zeta \geq 4. \end{cases} \tag{15}$$

If the neighbouring nodes are not congested the load increases at a rate of μ, and drops to zero if all the neighbours are congested. This control mechanism is inspired by the "additive increase/multiplicative decrease" [24] used in packet networks. Valverde and Solé studied this control method in a Manhattan network with *Poisson* traffic and unlimited queue capacity. It was noticed that the packet release and the density of hosts satisfies the relation $\lambda \sim \rho^{-1}$. On average, the packet release reaches a steady critical state. This does not mean that the local traffic creation rates are all similar, as the congestion levels vary widely between nodes. An interesting observation made by Valverde and Solé is the existence of a synchronization effect in the congestion state of distinct nodes.

4.3 Control in scale-free networks using *TCP*

The above model is an oversimplified packet network as the topology of a real network is, of course, not a regular network. Also, the model does not allow for packets dropped in transit (as occurs in the Internet) because of the unlimited queues on the nodes. In this section we introduce a more realistic model by considering a scale-free network generated using the *IG* algorithm with a power law decay index of 2.22. Similarly to the previous simulations, each node is designated as either a host or a router. Routers have a single routing queue that receives packets in transit across the network, and releases them back onto the network at a rate governed by the connectivity of the node. The difference with the previous examples is that for this network the simulation is of the fixed-increment time advance type rather than next-event time advance. This allows the routing queue service rate to be set as Ck^{α}, C constant packets per time tick of the simulation, where k is the degree of the router node. This means that nodes with larger degrees produce more traffic. The index α has been chosen to be between 1 and 2 here. Hosts have identical routing queues and function in the same way, but additionally act as sources. They have transmit buffers that hold packets generated by *LRD* and Poisson traffic sources until they have been acknowledged. The exact mechanism for this is described below.

The control mechanism uses a simplified version of *TCP* Reno [24] as the network protocol. This is the predominate protocol used on the Internet at present. This version is derived from that described in Erramilli *et al* [10]. It concentrates on the slow-start mechanism because in real networks this not only affects all connections, but is also the dominant effect for most connections. It also has a more marked effect on the network dynamics than congestion avoidance.

As in our previous sections, a double intermittency map is used as the basis for each *LRD* traffic source, and a uniform random number generator for each Poisson source. However, they are used in a slightly different way. One sojourn period in the 'ON' side represents a whole file which is then *windowed* using the *TCP* slow start algorithm: at the start of the file the window size is set to 1. Only a single packet is sent at that time tick. When a packet reaches its destination an acknowledgement is returned to the source: once this packet has been acknowledged, the window size is doubled and the next two packets are sent. When both these packets have been acknowledged, the window size is doubled again and a new window of packets is sent. The doubling process is repeated until the end of the file or the maximum window size (a fixed value). This is described mathematically in section 3.4 on *TCP* dynamics.

The routing algorithm uses a pre-calculated look-up table of shortest paths. All links between nodes are assumed to have unit length. At each time step packets are forwarded from the head of each routing queue. If an acknowledgement packet reaches its destination, this triggers the release of the next window of packets from that host.

In Fig. 4 we see comparisons are made between the different topologies and packet transport algorithms used in this and the previous work [28]. Four combinations for networks are considered:

- scale-free(produced using the *IG* model) with a closed-loop algorithm (*TCP*).
- Manhattan with an open-loop algorithm.
- scale-free(produced using the *IG* model) with an open-loop algorithm.
- Manhattan network with with a closed-loop algorithm.

The network has 1024 nodes and the host density is taken to be 589/1024 for all cases. The index α of the server strength $0.25n^{\alpha}$ is fixed at 1 for all four combinations. The host distribution is random for the Manhattan network; the *IG* network connectivity 1 and 2 nodes are selected to be hosts.

In Fig. 4(a) average lifetimes are shown as a function of the load for the four cases with *LRD* traffic sources at each host. Average lifetimes include the waiting time in transmit buffers and measurements from actual networks would not include this. Also, loads quoted are those offered to the transmit buffers, rather than the loads exiting these buffers which enables end users experience to be modelled.

Clearly the closed-loop *TCP* algorithm increases lifetimes for both regular and scale free networks. The requirement of *TCP* that packets be acknowledged before the next window of packets is sent is restrictive in the sense that

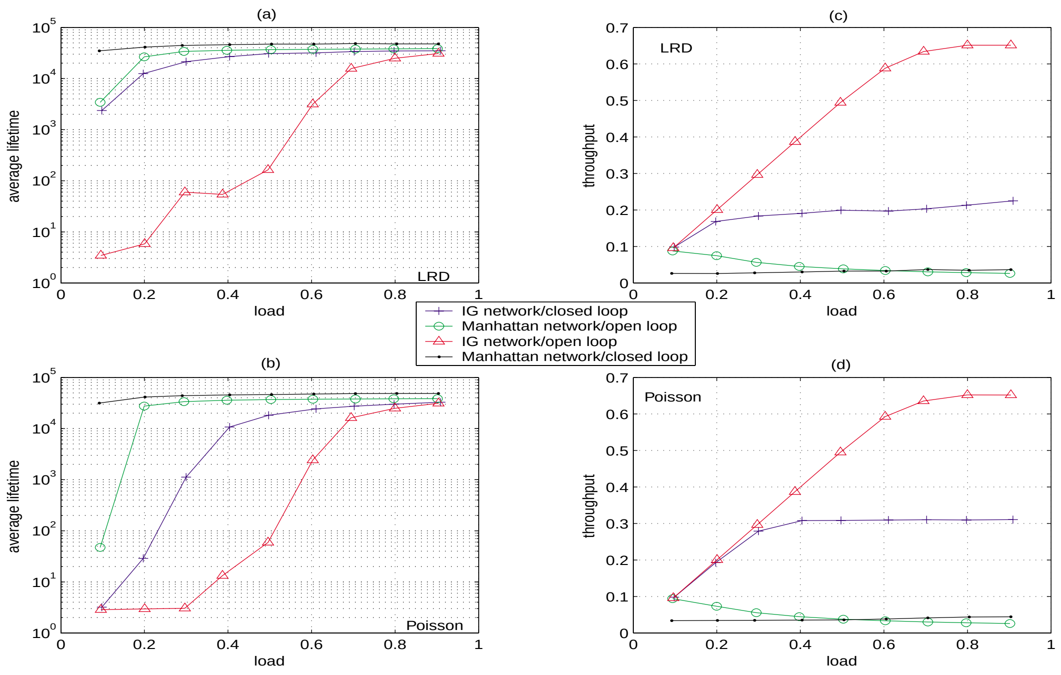

Figure 4: Comparisons between the open-loop simulations and closed-loop simulations on regular networks and scale-free *IG* networks. Network sizes and host densities are the same for all simulations.

new windows cannot be sent by hosts more frequently than the round trip times (RTT) or ping times. Congestion is reacted to immediately because it causes an increase in RTT's and makes sources back off. Since file sizes are not that great and window sizes are often reset to 1 due to the round-trip time maximum RTO limit, throughput can never be that high. This results in packets being delayed in the transmit buffer and is the primary cause of increased packet lifetimes. By contrast the open-loop algorithm does not react to congestion and allows unlimited queues to build up at routers. For sufficiently high loads open-loop networks become congested and lifetimes approach those of the closed-loop networks.

In addition, Fig. 4(a) shows that the network topology is also a very important factor. In fact a Manhattan network with an open-loop algorithm has longer lifetimes than an *IG* network with a closed-loop algorithm primarily because of the much shorter average path lengths in the *IG* network. The average path length in the Manhattan network is 16 'hops' (one 'hop' is the distance between neighbouring nodes). By contrast in an *IG* network ordinary nodes connect to rich nodes with a high probability, and rich nodes also connect to one another preferentially. This leads to much shorter average path lengths.

In Fig. 4(b) the same measurements are made with Poisson traffic sources substituted for *LRD* sources. Results are very similar.

Fig. 4(c) shows throughput plotted against load. Results are consistent with those for average lifetime. Identical networks using open-loop algorithms have higher throughputs; the *IG* network performs more efficiently for both types of algorithm. Similar behaviour is observed for Poisson sources. Fig. 4(d) shows a substantial decrease in the throughput for scale-free TCP networks when *LRD* is introduced into the network. The correspoonding strong increase in packet lifetimes is seen in Fig. 4(c).

Thus, regular open-loop simulations are fundamentally different to the closed-loop simulations of *IG* networks and so detailed comparisons are not very useful.

5 Discussion

The theorem for double intermittency throws up serious problems for the discussion of ethernet traffic by the use of the Hurst parameter. The theorem shows that the Hurst parameter can take its value from either the intermittency effect at $x = 0$ or at $x = 1.$, i.e. *LRD* from long 'OFF', repsectively 'ON', packet sequences. However, the problems of *LRD* effects transit to the queues in the case of 'On' *LRD* dependency.

The theorem shows that that the Hurst parameter can can take its value from from either the intermittency effect at $x = 0$ or at $x = 1$, i.e. *LRD* from long 'OFF', respectively 'ON' packet sequences. However the problems of *LRD* effect transit to the queues in the case of only 'ON' *LRD* dependency. R. Mondragon *et al* [19] have shown that a queuing system exhibited a completely different behaviour depending on which kind of incoming traffic it was confronted with, even though this incoming traffic was long range dependent in every case. In the first case, the *LRD* traffic was characterized by heavy tailed silent periods and exponentially decaying active periods and the resulting queue behaved in the same way as it would have done under Poisson-like conditions with exponential decay of queue length. An inversion of these characteristics resulted in power law decay for queue-length distributions. This confirms that we need a deeper understanding is needed since two pseudo-traffic traces, both long-range dependent but with a different underlying feature, can produce such different outcomes. For example, M. Crovella *et al.* [8] discuss the physical origins of self-similarity and show that, while active periods are the dominant cause of self-similarity, there are some signs of heavy-tailed silent periods.

The technique of using chaotic maps to simulate packet traffic entry into a network is now established. This paper shows that the techniuqe can be adapted and extended to dela with a wide variety of topologies of networks and robust comparisons can be made of both packet lifetime and throughput characteristics for these networks. The differences in traffic characteristics between closed loop(*TCP*) and open-loop and regular and scale-free are clearly distinguished.

References

[1] R. H. Albert and A. Jeong and A.-L. Barabási, Diameter of the world wide web, *Nature*,1999, **401**, 130-131.

[2] D.K. Arrowsmith and R.J. Mondragón and J.M. Pitts and M.Woolf, Phase Transitions in Packet Traffic on Regular Networks: a comparison of source types and topologies, *Preprint*, 2004.

[3] D.K. Arrowsmith and M.Woolf, Modelling of TCP packet traffic in a large interactive growth network, *to appear IEEE Proceedings of Systems and Circuits, Vancouver*, 2004.

[4] A. L. Barabási and R. Albert, Emergence of Scaling in Random Networks, Science, 1999, **266** 509-512.

[5] M. Barenco and D.K. Arrowsmith, The autocorrelation of double intermittency maps and the simulation of computer packet traffic, *Jnl of Dynamical Systems*,**19**(1) 2004, 61-74.

[6] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W.Willinger, "The Origin of Power Laws in Internet Topologies (Revisited)," *Proc. of INFOCOM 2002*, 2002.

[7] R. Cohen and S.Havlin, Scale–free Networks are Ultrasmall, *Phys. Rev. Lett.*, 2003, **90**(5), 058701.

[8] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic, evidence and possible causes. *Proc. ACM Sigmetrics, Philadelphia*, 1996.

[9] A. Erramilli and R. P. Singh and P. Pruthi, An application of deterministic chaotic maps to model packet traffic, *Queueing Systems*, 1995, **20**, 171–206.

[10] A. Erramilli, M. Roughan, D. Veitch and W. Willinger, Self-Similar Traffic and Network Dynamics, *Proc. IEEE*, **90**, 2002, 800-819.

[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos, On Power-Law Relationships of the Internet Topology, *Proc. ACM/SIGCOMM, Comput. Commun. Rev.*, **29**, 251-262, 1999.

[12] H. Fuks and A.T. Lawniczak, Performance of data networks with random links, *Math and Computers in Simulation*, **51** 1999, 101-17.

[13] H.E. Hurst, Long-range storage capacity of reservoirs, *Trans Am. Soc. Civ. Eng.* **116**, 1951, 770-99.

[14] H. Fuks, A.T. Lawniczak and S. Volkov, Packet delay in data network models, ACM Transactions on Modeling and Computer Simulation, 2001, **11**(3) 233–250.

[15] A. Giovanardi, G. Mazzini, R. Rovatti, and G. Setti. Features of chaotic maps with self- similar trajectories. *Proc. NOLTA 1998*, 203-206.

[16] A. Giovanardi, G. Mazzini, and R. Rovatti, Chaos based self-similar traffic generators, *Proc. NOLTA 2000*, 747-50.

[17] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, On the Self-Similar Nature of Ethernet Traffic. *Proc. ACM SIGCOMM* 93, 1993.

[18] H. Li and M. Maresca , Polymorphic-torus network, *IEEE Transs Comp.*, 1989, **38**(9) , 1345-1351.

[19] R.J. Mondragón, A Model of Packet Traffic using a Random Wall, *Int. Jou. of Bif. and Chaos*, 1999, **9** (7), 1381-1392.

[20] T. Ohira and R. Sawatari, Phase Transition in Computer Network Traffic Model, Physical Review E, 1998, **58**, 193–195.

[21] H.G. Schuster, *Deterministic Chaos: An Introduction*, 3^{rd} Edition VCH 1995, 91-97.

[22] R. V. Solé and S. Valverde, Information transfer and phase transitions in a model of internet traffic, *Physica A*, 2001, 289, 595–605.

[23] W. Stallings, Data and Communications, *Prentice Hall Int. Inc.*, 2000.

[24] W. R. Stevens, TCP/IP Illustrated, Volume 1: The Protocols, *Addison-Wesley*, 1994.

[25] A. Vázquez, R. Pastor–Satorras and A. Vespignani, Large–scale topological and dynamical properties of the Internet, Physical Rev. E, **.65** 2002, 066130.

[26] S. Valverde and R. V. Solé, Self-organized critical traffic in parallel computer networks, *Physica A*, 2002, **312**, 636.

[27] X-J Wang, Statistical physics of temporal intermittency, *Phys Rev A* **40**(11), 1989, 6647-61.

[28] M. Woolf and D.K. Arrowsmith and R.J. Mondragón and J. M. Pitts, Optimization and phase transition in a chaotic model of data traffic, *Physics Reviews E*, 2002, 66, 056106.

[29] M. Woolf and D.K. Arrowsmith and S. Zhou and R.J.Mondragón and J.M. Pitts, Dynamical modelling of *TCP* packet traffic on scale-free networks, *submitted to Phys Rev E*, 2004.

[30] S. Zhou and R. J. Mondragón, 'Towards modelling the Internet topology – the Interactive Growth model,' *Proc. of the 18th International Teletraffic Congress (cs.NI/0308029)*, Sept. 2003.

[31] S. Zhou and R. J. Mondragón, 'Analyzing and modelling the AS–level Internet topology,' *Proc. of HET-NETs' 03*