

Connection-level Analysis and Modeling of Network Traffic

Shriram Sarvotham, Rudolf Riedi, Richard Baraniuk

June 30, 2001

Abstract

Most network traffic analysis and modeling studies lump all connections together into a single flow. Such aggregate traffic typically exhibits long-range-dependent (LRD) correlations and non-Gaussian marginal distributions. Importantly, in a typical aggregate traffic model, traffic bursts arise from many connections being active simultaneously. In this report, we develop a new framework for analyzing and modeling network traffic that moves beyond aggregation by incorporating connection-level information. A careful study of many traffic traces acquired in different networking situations reveals (in opposition to the aggregate modeling ideal) that traffic bursts typically arise from a single high-volume connection that dominates all others. We term such dominating connections *alpha traffic*. Alpha traffic is caused by large files transmissions over high bandwidth links and is extremely bursty (non-Gaussian). Stripping the alpha traffic from an aggregate trace leaves a *beta traffic* residual that is Gaussian, LRD, and shares the same fractal scaling exponent as the aggregate traffic. Beta traffic is caused by both small file transmissions and large files transmissions over low bandwidth links. In our alpha/beta traffic model, the heterogeneity of the network resources give rise to burstiness and heavy-tailed connection durations give rise to LRD.

1 Introduction

Network traffic analysis and modeling play a major role in characterizing network performance. Models that accurately capture the salient characteristics of traffic are useful for analysis and simulation, and they further our understanding of network dynamics and so aid design.

Most traffic analysis and modeling studies to date have attempted to understand *aggregate traffic*, in which all simultaneously active connections are lumped together into a single flow. Typical aggregate time series include the number of packets or bytes per time unit over some interval. Numerous studies have found that aggregate traffic exhibits *fractal* or *self-similar* scaling behavior, that is, the traffic “looks statistically similar” on all time scales [18]. Self similarity endows traffic with *long-range-dependence* (LRD) [18]. Numerous studies have also shown that traffic can be extremely bursty, resulting in a non-Gaussian marginal distribution [29]. These findings are in sharp contrast to classical traffic models such as Markov or homogeneous Poisson. LRD and non-Gaussianity can lead to much higher packet losses than predicted by classical Markov/Poisson queueing analyses [6, 18].

The discovery of self-similar behavior in traffic led immediately to new fractal aggregate traffic models (see [2, 31], for example). *Fractional Gaussian noise* (fGn), the most widely applied fractal model, is a Gaussian process with strong scaling behavior. Due to its Gaussianity, it lends itself to rigorous analytical studies of queueing behavior. Also, approximate fGn can be synthesized rapidly by a variety of different techniques, including wavelets.

A strong argument for fGn in networks is that often aggregate traffic can be viewed as a superposition of a large number of independent individual ON/OFF sources, with the ON durations heavy-tailed [3, 34].

Unfortunately, fGn is unrealistic for bursty non-Gaussian traffic. For instance, when the standard deviation of the traffic exceeds its mean, a considerable portion of an fGn traffic synthesis is negative. These failings have motivated more complicated models for aggregate traffic such as multifractals and infinitely divisible cascades [29]. However, while more statistically accurate, these models lack network relevance in their parameterizations.

Aggregate analysis and modeling lump all traffic bytes or packets together into a single flow. In this report, we exploit the *connection-level information* contained in most publicly available traffic traces that aggregate models ignore. Define a connection as the unique four-tuple comprising a source IP address, destination IP address, source port number, and destination port number.

Connection-level information enables us to conduct a refined analysis of traffic bursts. In aggregate traffic models (including the ON/OFF model), traffic bursts arise from a large number of connections transmitting bytes or packets

simultaneously. That is, bursts stem from a kind of “constructive interference” of many connections. With connection-level information, we can test this hypothesis. If it were true, then we should observe in real traffic traces a large number of active connections during bursts. However, this was not the case. The bursts in the bytes-per-time plot generally do not coincide with large values in the connections-per-time plot.

Quite to the contrary, a careful analysis of a great many real traces reveals that generally *a single high-rate connection dominates during a burst*. This surprising finding has far-reaching implications for traffic analysis and modeling.

To explore further, we propose a new analysis technique that exploits connection-level information to separate traffic into two distinct components at a time-scale T of interest. We call the traffic corresponding to the dominant connections the *alpha* component. The residual traffic is called *beta* component.¹ Our procedure decomposes an aggregate traffic trace into

$$\text{total traffic} = \text{alpha traffic} + \text{beta traffic.} \quad (1)$$

We have applied the alpha/beta traffic decomposition to many real-world traffic traces (from Auck [24] to LBL [16]) and found tremendous consistency in our results.

Beta traffic: At time-scales coarser than the round-trip time, the beta component is very nearly Gaussian and strongly LRD (i.e., approximately fGn), provided a sufficiently large number of connections are present. Moreover, the beta component carries the same fractal scaling (LRD) exponent as the aggregate traffic.

Alpha traffic: The alpha component constitutes a small fraction of the total workload, but is entirely responsible for the bursty behavior.

Our alpha/beta decomposition technique suggests an intuitive and natural traffic model that takes into account both the network topology and user behavior. A major conclusion of our analysis is that the burstiness in network traffic is caused by the *heterogeneity* in link speeds and computational power within the network (including the networking software/hardware of the clients) and user behavior.

Consider a simplified taxonomy of a network system. There are roughly two kinds of file sizes: large ones such as jpeg images and small ones such as text emails. There are also roughly two speeds of connections: fast ones such as Ethernet or DSL lines and slow ones such as 56k modems. We argue that *only large files over fast links contribute to alpha traffic*. The remaining combinations aggregate together into fGn beta traffic. Therefore, we model traffic as a sum of alpha and beta traffic. Such traffic is simple to both analyze and synthesize.

The beta component succinctly collects all the “average” connections and is well modeled as fGn with LRD parameter equal to that of the overall traffic. The alpha component consists of the dominant, burst causing connections and contributes low traffic volume but essentially all bursts, which arrive in a pattern according to the capabilities of network and requests of the clients routing through the given point of measurement. The alpha burst arrivals are not exactly Poisson but can to a first approximation be modeled as such (most likely, they are compound Poisson). Furthermore, the dominant connections are uncorrelated with the overall connection arrivals and thus can be modeled as an independent process.

Careful analysis of many data sets has ruled out the possibility of other possible mechanisms that could cause alpha traffic bursts, such as unruly TCP slow starts, sudden rerouting, or transient behavior due to starting and stopping of connections. Rather, as a rule, we found all bursts to be caused by source-destination pairs with fast connections and large-volume transfers. Moreover, we found that any source-destination pair that once causes a burst will always cause a burst when transferring a large load.

Our alpha/beta model is predictive, since for a given topology and user behavior, we can determine a priori which connections will aggregate into the fGn background and which connections will cause bursts. Syntheses from our model then closely match real traffic and controlled *ns* simulations. For example, under heavy utilization (when the router at which we take our measurements becomes itself the bottleneck link for all connections) or with a considerably homogeneous clientele, we should and do observe fewer bursts and more Gaussian traffic. We observe exactly the contrary in the LBL data [16]; there are not enough active connections to give a full fGn beta component once the dominant connection is stripped away.

Finally, through queueing simulations, we have demonstrated that the beta component affects the tail queue probability for *small* queue sizes, whereas the alpha component determines the tail queue probability for *large* queue sizes.

¹By analogy to the dominating *alpha males* and submissive *beta males* observed in the animal kingdom.

2 Background on aggregate traffic models

There exist a wide range of mathematical models of self-similar or long-range dependent traffic each with its own idiosyncrasies [2, 4, 8, 19, 25, 29, 34]. Some are physically motivated and others show that long-range dependence may be generated in diverse ways. We review some of the important traffic models. In the fGn model [6], the traffic is modeled as a fGn process that is characterized by strong scaling behavior and Gaussianity. The *wavelet domain independent Gaussian* (WIG) model [15, 20] synthesizes a Gaussian process capturing both the long and short-term correlations, using wavelet transforms. In the *multifractal wavelet model* (MWM) [29], the multifractal scaling behavior of the traffic is captured. Finally, we look at the physically motivated ON/OFF model [34] in which the traffic is modeled as an aggregate of ON/OFF sources with heavy tailed ON and OFF times.

2.1 fGn model

Fractional Gaussian noise (fGn) processes possess self-similar scaling properties that are also observed in network traffic. Hence fGn serves as a good model for network traffic.

A process Y is H -self-similar with stationary increments (H -sssi) if it has stationary increments and for all $a > 0$

$$Y(at) \stackrel{fd}{=} a^H Y(t) \quad (2)$$

A random process that satisfies 2 is the *fractional Brownian motion* (fBm) $B(t)$. This process is uniquely defined through two properties: H -sssi and Gaussianity [21]. The Hurst parameter lies in the range $0 < H < 1$.

The increments of fBm, $G(n)$ are known as the *fractional Gaussian noise* (fGn), defined by

$$G[n] = B(n\delta t) - B((n-1)\delta t), \quad (3)$$

for finite increment δt . While fBm is non-stationary, fGn is stationary.

It can be shown [9] that fGn has an autocorrelation function

$$r_G[k] = \frac{\sigma^2}{2} |\Delta|^{2H} (|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}), \quad (4)$$

with δ a constant. Moreover, fGn satisfies the scaling property

$$m^{1-H} G^{(m)} \stackrel{fd}{=} G \quad (5)$$

where $G^{(m)}$ is the aggregate

$$G^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} G(i) \quad (6)$$

Network traffic modeling using fGn has the big advantage of providing a complete description of the resulting traffic with just three parameters: mean, variance, and the Hurst parameter. Erramilli et al [5] note that the fGn model can be expected to be an appropriate model for data traffic provided (i) the traffic is aggregated over a large number of independent and not too wildly fluctuating connections (to ensure Gaussianity), (ii) the effects of flow control on any one connection is negligible and (iii) the time scales of interest coincide with the scaling region where (5) holds. In practice, these conditions are often satisfied in the backbone (i.e., high levels of aggregation) and for time scales larger than the typical round-trip time of a packet in the network [30].

2.2 Wavelet-domain Independent Gaussian (WIG) model

Wavelets serve as an approximate Karhunen-Loève or decorrelating transform for fBm [9], fGn, and more general LRD signals [15]. This property can be exploited for modeling LRD traffic, by modeling the data in the wavelet domain instead of the time domain.

The WIG model synthesizes Gaussian LRD data by generating the parent node of the scaling coefficient tree, $U_{0,0}$, with a required Gaussian distribution and wavelet coefficients as independent (and hence uncorrelated) zero-mean Gaussian random variables, identically distributed within scale according to

$$W_{j,k} \sim N(0, \sigma_j^2), \quad (7)$$

with σ_j^2 the wavelet-coefficient variance at scale j [9, 14, 20, 33, 35]. Scaling coefficients at finer scales on the tree are then recursively computed until the finest scale scaling coefficients and hence the required signal is obtained. The result is a fast $O(N)$ algorithm for generating a length- N signal.

An attractive feature of the WIG model is its flexibility in matching different correlation structures of LRD processes. A power-law decay for the σ_j^2 's leads to approximate wavelet synthesis of fBm or fGn [9, 35]. However, while network traffic may exhibit LRD consistent with fBm or fGn, it may have short-term correlations that vary considerably from pure fBm or fGn scaling. Such LRD processes can be modeled by setting σ_j^2 to match the measured or theoretical variances of the wavelet coefficients of the desired process [20]. Thus, for a length- N signal, the WIG is characterized by approximately $\log_2(N)$ parameters.

2.3 Multifractal wavelet model

Unlike the WIG model, the *multifractal wavelet model* (MWM) is based on a multiplicative cascade in the wavelet domain. MWM matches the qualitative visual appearance, queueing behavior and the marginals of bursty network traffic better than the Gaussian models like fGn and WIG [29], especially at fine time scales.

MWM is based on a multiplicative cascade in which refinements from coarse to finer time scales is done multiplicatively. Given the approximation to the signal $X(t)$ at resolution 2^{-j} , we compute the wavelet coefficients $W_{j,k} = A_{j,k}U_{j,k}$ with random $A_{j,k}$. Here j and k refer to the scale (higher j is finer time resolution) and location of the wavelet coefficient. The approximation to $X(t)$ at resolution $2^{-(j+1)}$ is then obtained from scaled sums and differences of the $U_{j,k}$ and $W_{j,k}$. This process can be iterated until any desired resolution is reached, with a total cost associated $O(N)$ for N point output.

Multiplicative cascades like the MWM synthesize data with approximately lognormal marginals, due to the central limit theorem.

2.4 ON/OFF model

The ON/OFF model, hinted in [17] and formally introduced in [34] is a mathematical abstraction which provides a foundation for physical traffic modeling based on physically verifiable properties. The basis of this model is that the network traffic is a superposition of a large number of independent ON/OFF sources with heavy tailed ON and OFF periods. This model gives rise to self-similarity in the aggregate process — an fGn process — whose LRD is determined by the heavy-tailed nature of the ON and OFF periods. The ON/OFF model has its roots in a certain renewal reward process introduced by Mandelbrot [22] (and further studied in [32]) and provides the theoretical underpinnings for much of the recent works on the physical modeling of network traffic. K. Park et al [26] have shown that the application layer property of heavy-tailed file sizes is preserved by the protocol stack and mapped to approximately heavy-tailed busy periods at the network layer. The ON/OFF model is able to induce both Gaussianity and asymptotic second order self similarity.

The weakness of the ON/OFF model is that it assumes that all connections send data at the same rate during the ON period. It also assumes independence of ON/OFF sources. The resulting trace obtained from ON/OFF models is Gaussian, and thus does not model bursty traffic well.

3 Connection level traffic analysis

Network traffic is highly non-Gaussian [1, 29] and as a result the classical models such as fGn and ON/OFF are inadequate to describe it. In this chapter, we present a connection-level analysis of network traffic to understand the cause of non-Gaussianity. We show that the aggregate traffic can be separated into two components: the Gaussian (Beta) component and the bursty (Alpha) component. Whereas the majority of connections constitute the Beta component of traffic, a very small number of burst causing connections add up to give the Alpha traffic that has heavy tailed marginals. We look at several schemes to separate the Alpha and Beta components of traffic. In cases where connection level details are not available, we perform fGn denoising using a wavelet based colored denoising technique [13] for burstiness extraction. We also study the hourly variation of burstiness in traffic during the course of the day. The analysis indicates that network traffic is more Gaussian (i.e., kurtosis² closer to 3) when the link utilization is high.

²Please see appendix for a review on kurtosis

Table 1: Network trace files used in the study

Dataset	Filename	Duration (sec)	Number of packets	Packets recorded
Auck-1	20000125-143640-0	2086	650,732	all
Auck-2	19991207-125019-0	3182	1,000,000	all
Auck-3	19991207-125019-1	3182	971,673	all
Auck-4	19991201-192548-0	86400	16,690,066	all
Auck-5	19991201-192548-1	86400	15,561,266	all
DEC-PKT-3	DEC-PKT-3	3600	2,873,588	TCP
LBL-PKT-4	LBL-PKT-4	3600	862,945	TCP
LBL-PKT-5	LBL-PKT-5	3600	677,845	TCP

Table 2: Trace file details

Dataset	overall data sent (MBytes)	Mean data rate (KBytes/sec)	Number of connections	Utilization (percent)
Auck-1	341.14	163.51	43728	65.40
Auck-2	571.23	179.47	64087	71.78
Auck-3	227.82	71.57	65372	28.63
Auck-4	7934.7	36.73	Not available	36.73
Auck-5	4717.0	21.83	Not available	21.84
DEC-PKT-3	650.32	180.66	66252	Not available
LBL-PKT-4	130.93	36.38	6238	Not available
LBL-PKT-5	94.13	26.16	4777	Not available

3.1 Network traces used

Three sets of traces, namely Auck, DEC and LBL are used in the analysis, as summarized in Table 1.

The first set of traces were recorded at the University of Auckland [24]. The traces are part of a large collection of GPS synchronized IP header traces captured with a DAG2 system at the University of Auckland Internet uplink by the WAND (Waikato Applied Network Dynamics) research group, which is based in the University of Waikato Computer Science Department. The University of Auckland ITSS department operates an OC3 ATM link to carry a wide variety of services off the main campus. A single ATM channel is used to connect the university to the global Internet, and since it is the only connection, all packets for all external connections pass the measurement point. The connection has a packet peak rate of 2 Mbits/sec in each direction. For more details, see [24].

The DEC-PKT-3 trace was gathered at the primary Internet access point for the Digital Equipment Corporation. Digital's Palo Alto research groups operate the access points. The trace captured all TCP packets and lasted 1 hour.

The last set of data are the LBL-PKT-4 and LBL-PKT-5 gathered at the Lawrence Berkeley Laboratory's wide-area Internet gateway [16]. The traces captured all TCP packets and lasted 1 hour each. For more details on the LBL and DEC traces, see [16].

All traces have details on individual packets, such as the timestamp (with resolution 1 microsecond), packet size, source IP address and port number, and destination IP address and port number. The traces used are summarized in Table 1.

3.2 Analysis of burstiness in network traffic

To analyze burstiness in network traffic, consider the Auck-2 trace as an example. Figure 1(a) shows the bytes per time plot of this trace at 500ms time-scale. The traffic is clearly spiky at this time scale. The histogram of the traffic, shown in Figure 1(b) has a slowly decaying tail. This is also reflected in the kurtosis (see appendix for a review of kurtosis) value of 5.81, which indicates that the traffic trace is non-Gaussian.

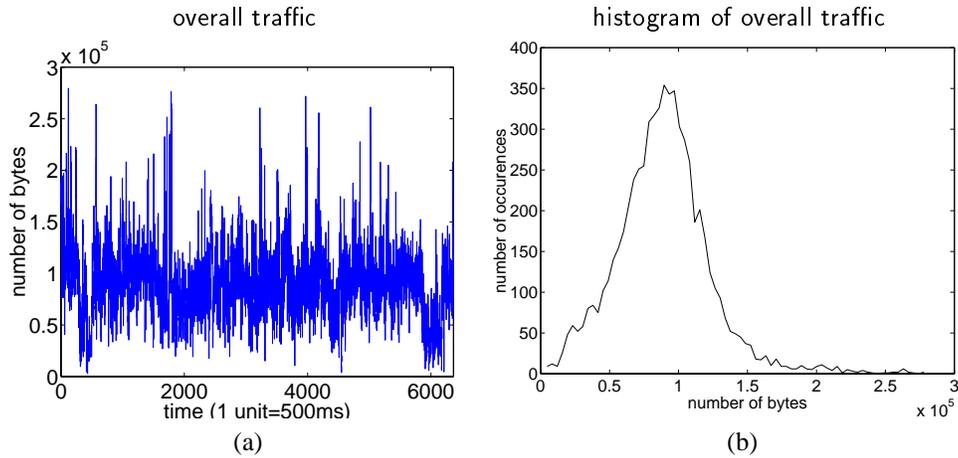


Figure 1: (a) Bytes-per-time arrival process at 500ms aggregation level for Aggregate network traffic (Auck-2) and (b) Histogram of aggregate traffic.

Next, we look at the number of active connections in each time bin. We define a connection as a unique four-tuple comprising a source IP address, destination IP address, source port number and destination port number. Figure 2(a) shows the number of active connections at each time bin of 500ms. This trace looks significantly less bursty, and its histogram, shown in Figure 2(b), looks more Gaussian. Indeed, the kurtosis for this trace is 3.98. If the ON/OFF model was accurate, we would expect the bytes per time plot to be a scaled version of the connections per time plot. This is because, in the connections per time plot, each active connection contributes a constant value (namely *one*) in that time bin, similar to what the ON/OFF model does.

We note that the bytes per time plot and the connections per time plot look similar except for the spikiness in the former. This is reflected in the correlation coefficient between the two traces, which is 0.53.

Next we look at the connection level contribution of bytes at the Alpha components of the traffic. In the framework of the classical ON/OFF model, the peaks can be explained only through the presence of a very large number of active connections each sending data at a constant rate. To test whether this hypothesis is true, let us choose the peak at time bin 568 (corresponding to 284 seconds from the start of the trace). Figure 3(a) gives the connections strengths in bytes per time for the 90 active connections present in this time bin. The connections are numbered in descending order of connection strengths. It is clear from this figure that we have *one* connection that sends a significantly large number of bytes compared to the other active connections in this time bin. This reveals that the peak at time bin 568 is caused by a single dominant connection. To verify if all the spikes are due to single dominant connections, we do the following. We select the time instants when a single connection contributes more than 50% of the traffic and check if they pick out the bursts. Figure 3(b) illustrates that this indeed is the case. The selected time instances are circled in the trace, and we observe that most of the bursts are captured.

3.3 Separation into Alpha and Beta traffic

The above analysis of bursts motivates a separation of connections into two classes: one that dominates other active connections and cause bursts, and the other that comprise the remaining connections. The connections that cause bursts can be aggregated together to give the Alpha component of traffic. The remaining traffic is grouped together to give the Beta component. We show that the Beta component is Gaussian if there is sufficient amount of traffic. There are several ways in which the Alpha and Beta components can be separated. We discuss four schemes below.

3.3.1 Threshold-based separation

In this scheme we consider all time instances where the number of bytes exceeds the threshold of mean rate plus three standard deviations. For Auck-2 data, this threshold is 191,320 bytes. At each of the time instances where the

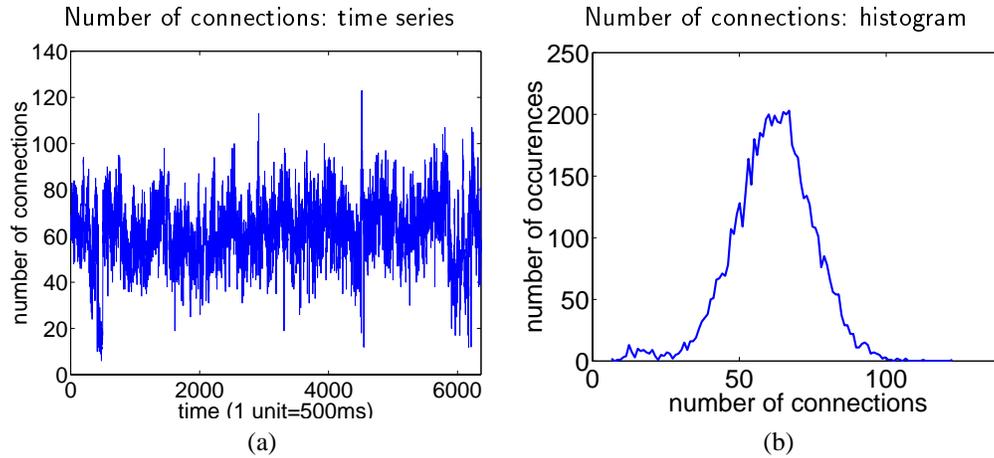


Figure 2: Number of connections active in intervals of 500ms duration for the Auck-2 trace: (a) time series and (b) histogram.

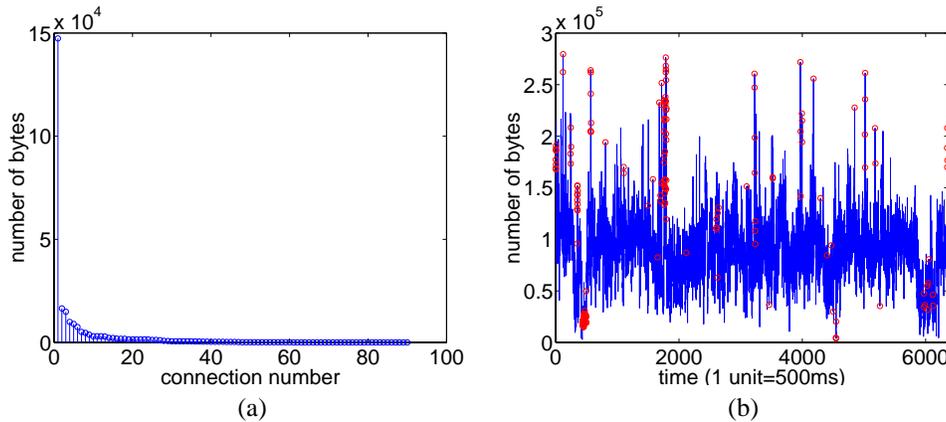


Figure 3: (a) Connection strengths at time bin 568 at 500ms resolution and (b) time instances when a single connection contributes more than 50% of the traffic.

data rate exceeds the threshold, we *subtract* the contribution of the largest connection. If the ON/OFF model were accurate, subtracting one connection at a peak should produce a traffic trace that looks similar to the original bytes per trace plot. This is because the model assumes that peaks are caused due to a large number of equally strong active connections, and as a result removing one component will not reduce the peak significantly. On the other hand, if the hypothesis that bursts are caused due to the presence of a single dominant connection is true, we should expect to see a considerably smoother traffic trace. Indeed, this is the case observed, as shown in Figure 4(a). The resulting trace (Beta component) looks significantly less spiky, and is reflected in the low kurtosis value of 3.36, which is very nearly Gaussian. The resulting trace can thus be well modeled by Gaussian models like fGn, WIG or ON/OFF. On the other hand, the kurtosis of the Alpha component (Alpha component) is 112.06, which suggests that the non-Gaussianity is captured in this component.

An important point to note is that the total number of connections that contribute to the Alpha component is just 64, out of a total of 64087 connections that constituted the overall traffic. Thus burstiness is caused due to a very small fraction of the total number of connections.

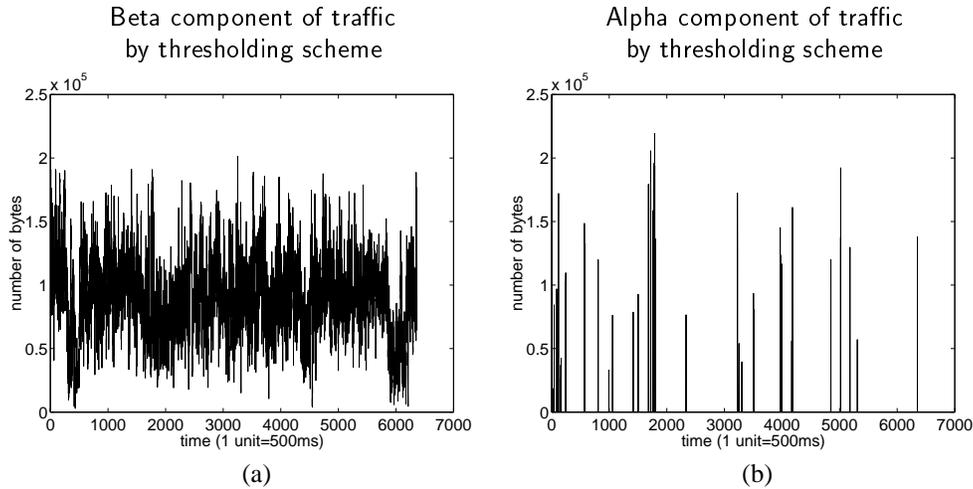


Figure 4: (a) Bytes-per-time arrival process at 500ms aggregation level for the Beta component of the traffic using thresholding scheme on Auck-2. Note its Gaussian character. (b) Similar Alpha component. Note its bursty character.

3.3.2 Uniform separation

The second method to extract Alpha and Beta components is to remove the maximum connection component uniformly at *all points* in the trace. Figure 5 shows the resulting traces when this scheme is adopted. The kurtosis of the Beta component is 3.31 and that of the Alpha component is 35.91. The advantage of this scheme is its simplicity, requiring no thresholding. The disadvantage is that a large part of the Alpha traffic extracted in this scheme is smooth and not bursty. This is because we removed the maximum connections even at non-bursty parts of traffic. It is desirable to aggregate the smooth part with the Beta component. Also, the maximum connection component must be computed at each time bin, and this may require some computations.

3.3.3 Connection based separation

In this method, we compute the maximum rate at which *each connection* sends data in 500ms time-bins. We select a threshold of mean plus twice the standard deviation of the aggregate trace and divide the connections into two groups: one in which the maximum connection rate exceeds the threshold and the second in which the maximum rate is less than the threshold. For Auck-2 data, the number of connections whose rate exceeded the threshold is just 40 out of the total 64087 connections that constitutes the aggregate traffic. These 40 traces add up to give a highly bursty traffic (Alpha component) with kurtosis 37.78. The remaining connections add up to give a Gaussian trace (Beta component) with kurtosis 3.38. The resulting traces from this scheme are shown in Figure 6. Once again, we find that the number of connections that contribute to the bursts is significantly smaller than the total number of connections. The advantage of the connection based separation scheme is that we identify only those connections that are capable of causing bursts. Hence this is a good way to get the *smallest* set of connections that cause bursts. However, this scheme has some drawbacks. First, the computation time is high, since the maximum rate for *each* connection is required. Secondly, a connection need not send data at high rates throughout its lifetime. The burst causing connections may send data at low rates during its lifetime, and it is desirable to group this part with the Beta component.

3.3.4 Separation using wavelet denoising

In the last scheme to extract the Alpha and Beta components, we use only the aggregate traffic data and discard the connection level information. This scheme is useful when connection level data of the traffic is not available. This scheme is based on the fact that we can treat the Beta component as “noise” and the Alpha component as the “signal”, and use well known denoising techniques to separate the two. We use Wavelet based denoising techniques, with coefficient thresholding. For colored denoising, we use different thresholds for wavelet coefficients at different scales.

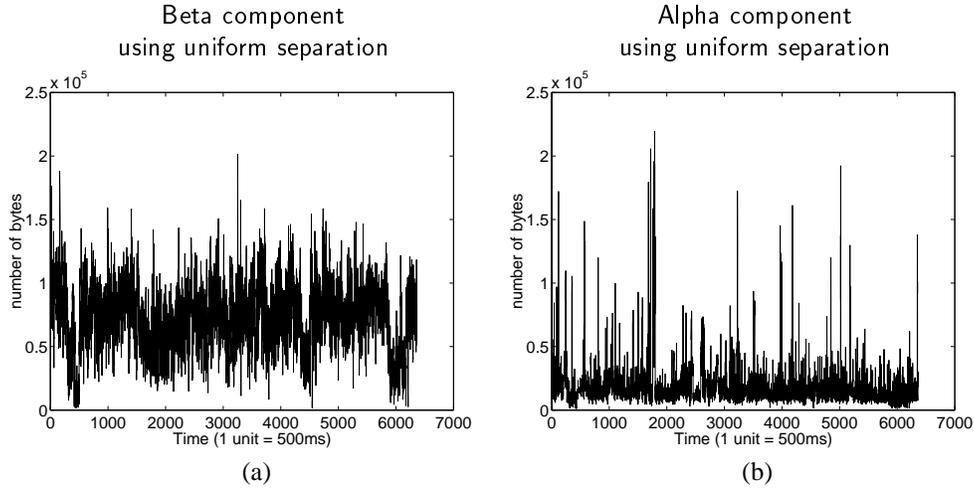


Figure 5: (a) Bytes-per-time arrival process at 500ms aggregation level for Beta component using uniform separation scheme on Auck-2 and (b) Alpha component

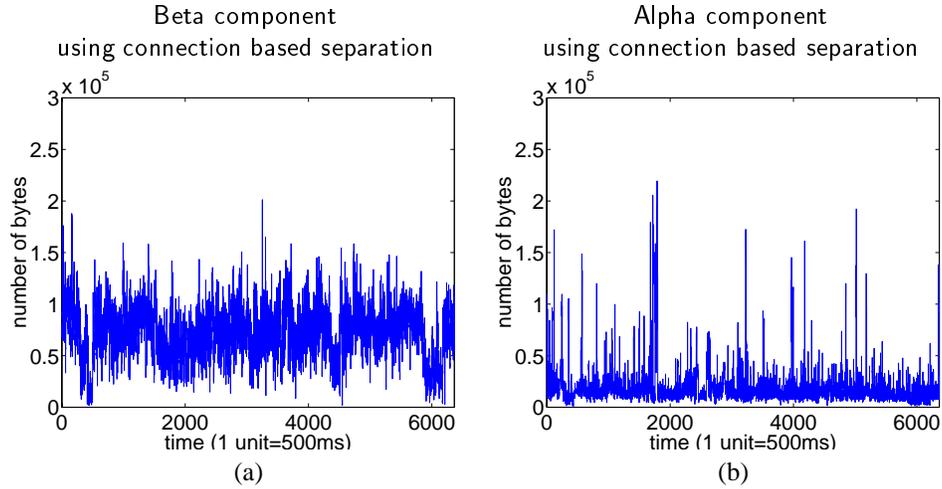


Figure 6: (a) Bytes-per-time arrival process at 500ms aggregation level for Beta component using connection based separation on Auck-2 and (b) Alpha component

Kaplan and Kuo [14] have shown that for Haar wavelet, the variance progression of the wavelet transform of fGn with Hurst parameter H satisfies

$$\text{var}(W_{j,k}) \propto 2^{-j(2H-1)} \quad (8)$$

where j is the resolution with increasing j denoting finer resolutions. In colored denoising scheme, the threshold at each scale is made proportional to the expected standard deviation of the wavelet coefficients at that scale. Thus, knowing the Hurst parameter, we can fix the threshold at each scale using equation (8) Johnstone et al [13] have shown that this thresholding scheme is optimal for colored denoising.

For the Auck-2 trace, the Hurst parameter is estimated as 0.87 using variance time plot technique (see next section). Figure 7 shows the Beta and Alpha components obtained by employing this colored denoising scheme. We use only the first 4096 data points in Auck-2 trace at 500ms resolution to facilitate wavelet decomposition. The kurtosis of this truncated data set is 6.10. Note that the Beta component is zero mean, with the mean contributing to the Alpha

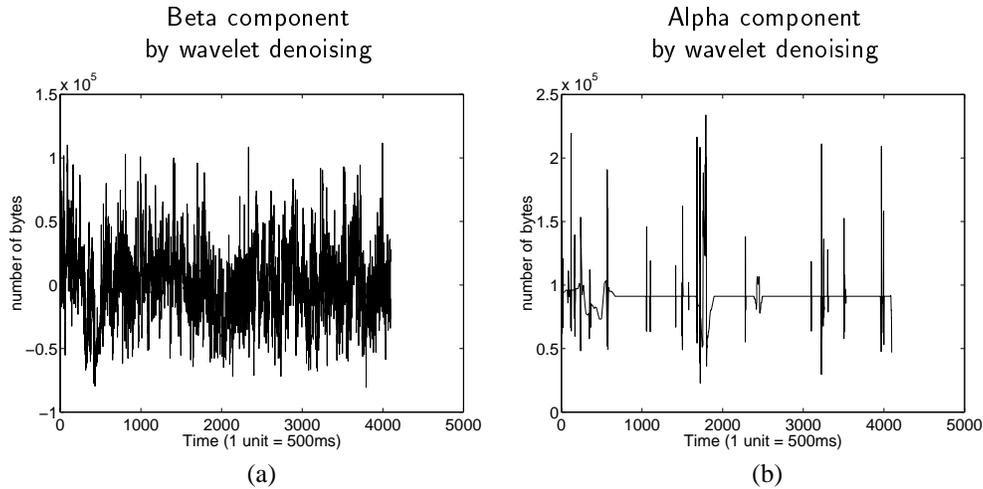


Figure 7: (a) Bytes-per-time arrival process at 500ms aggregation level for Beta component using colored denoising on Auck-2 and (b) Alpha component

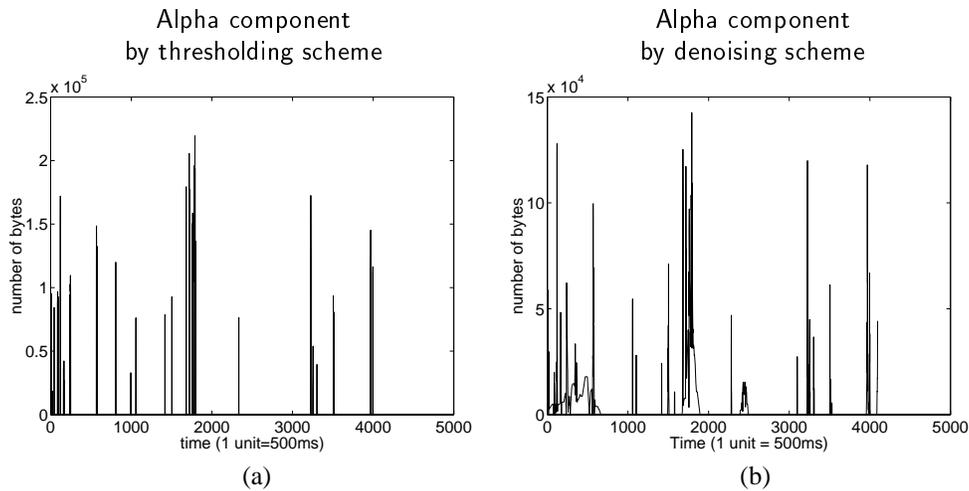


Figure 8: For a comparison of thresholding and denoising schemes we plot on the left a slight zoom into Figure 4(b) to display the same time range as in Figure 7(b). This plot is redone on the right, but where the mean is subtracted and the negative values are discarded. This allows to compare the ability of the schemes to detect burst.

component. The kurtosis for the Beta component is 3.42. To judge the effectiveness of the denoising scheme, we plot the Alpha component computed by thresholding scheme and denoising scheme alongside in Figure 8. The mean of Alpha component obtained from colored denoising is removed and only the positive values are plotted. We observe that the locations and intensities of the bursts in denoising scheme compares well with the those computed using the thresholding scheme.

3.4 Analysis of Beta traffic: Cause of LRD

We now argue that the LRD properties of the aggregate traffic comes from the Beta component. To see this, we use the variance-time plot to estimate the Hurst parameter for the three traces in Figure 9. The Hurst parameter value and the variance for the aggregate trace and the Beta trace are found to be nearly equal, with values of 0.87 and 0.88. Thus

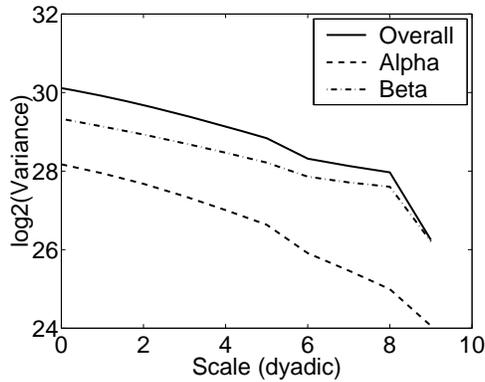


Figure 9: Variance-time plot of Auck-2 trace.

Table 3: Analysis of Auck-1 data.

Time scale (ms)	Kurtosis aggregate	Skew aggregate	Kurtosis Beta	Skew Beta	Kurtosis Alpha
50	4.778	1.009	2.994	0.626	165.829
500	4.754	0.737	2.821	0.299	200.992
5000	3.283	0.284	2.763	0.116	298.591

the Beta traffic carries the same fractal scaling exponent as the aggregate traffic.

3.5 Results of traffic trace analysis

We summarize the connection level analysis of the different traffic traces in Tables 3 to 8. In all the analysis, the thresholding scheme of component separation is used. For Auck-1, Auck-2 and DEC traces, the removal of burst causing connections results in a traffic trace with kurtosis very close to 3, indicating that it is very close to Gaussian. However, for the Auck-3 and LBL traces, the Beta component has high kurtosis. This is due to the small amount of data transfer, justified by the utilization for Auck-3 and number of connections in LBL traces. In scenarios with low link utilization or small number of connections, we do not have enough flows to add up in order to give rise to Gaussianity. To illustrate this point further, Figure 10 shows the histograms of the traffic traces at 500ms resolution. We observe that for Auck-3 and LBL traces, the histogram is not bell shaped. This is because we do not have enough flows that aggregate to give rise to Gaussian marginals. In contrast, the histograms of Auck-1, Auck-2 and DEC traces have a well defined bell-shape, which is captured well by the Beta component.

3.6 Dependence of burstiness on load

In this section we study the variation of burstiness with link utilization. Datasets Auck-4 and Auck-5 are used, both of which are 24 hour long traces. The traces are the forward and reverse flows recorded at the same link simultaneously. The traffic trace over the 24 hour duration is non-stationary. To circumvent the problem of non-stationarity, we split the traces into 24 sub-traces each of 1 hour duration. We assume that the traffic is stationary within the sub-traces. This assumption is found valid in the study of Paxson et al [27]. The main finding in our study is that the traffic is more non-Gaussian when the link utilization is small, and vice versa. Also, the Beta component of the traffic has kurtosis closer to 3 at high utilization.

Figures 12(a) and 12(d) give the utilization of the link on an hourly basis. This resembles the hourly variation in traffic rates studied previously in [27]. The start time of the traces is 19:25:48 hours on December 1, 1999 Auckland local time. The utilization is high during working hours with 60 to 90 percent in Auck-4 and 25 to 42 percent in Auck-

Table 4: Analysis of Auck-2 data.

Time scale (ms)	Kurtosis aggregate	Skew aggregate	Kurtosis Beta	Skew Beta	Kurtosis Alpha
50	4.777	0.892	2.976	0.455	143.057
500	5.808	0.803	3.359	0.112	112.062
5000	5.901	0.358	3.513	-0.296	151.470

Table 5: Analysis of Auck-3 data.

Time scale (ms)	Kurtosis aggregate	Skew aggregate	Kurtosis Beta	Skew Beta	Kurtosis Alpha
50	8.680	2.098	4.498	1.355	66.275
500	14.718	2.730	5.062	1.264	67.111
5000	15.098	2.640	4.201	1.044	76.372

Table 6: Analysis of DEC-PKT-3 data.

Time scale (ms)	Kurtosis aggregate	Skew aggregate	Kurtosis Beta	Skew Beta	Kurtosis Alpha
50	4.632	1.003	3.149	0.659	164.082
500	5.198	0.990	3.366	0.607	209.242
5000	4.701	0.780	2.977	0.389	166.420

Table 7: Analysis of LBL-PKT-4 data.

Time scale (ms)	Kurtosis aggregate	Skew aggregate	Kurtosis Beta	Skew Beta	Kurtosis Alpha
50	18.065	3.019	6.199	1.699	65.006
500	13.045	2.675	6.136	1.621	47.179
5000	7.485	1.961	5.051	1.416	48.778

Table 8: Analysis of LBL-PKT-5 data.

Time scale (ms)	Kurtosis aggregate	Skew aggregate	Kurtosis Beta	Skew Beta	Kurtosis Alpha
50	16.642	2.878	5.090	1.638	87.453
500	17.699	2.740	4.683	1.166	78.037
5000	12.567	2.001	4.555	0.860	91.379

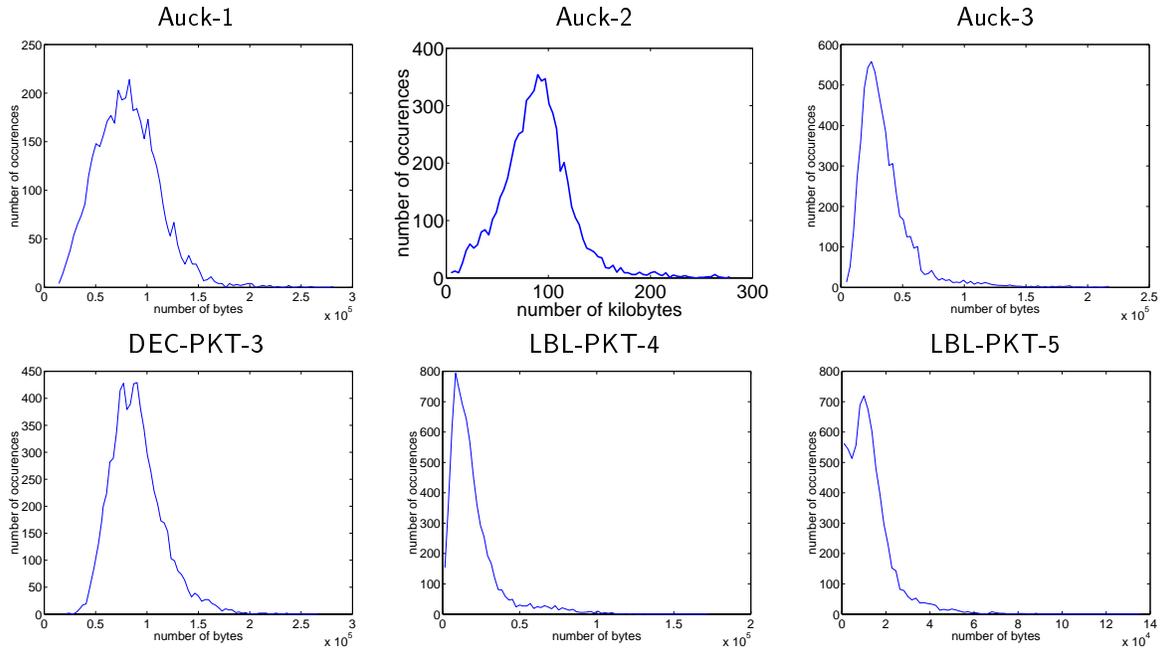


Figure 10: Histograms of aggregate traffic at 500ms for all traces

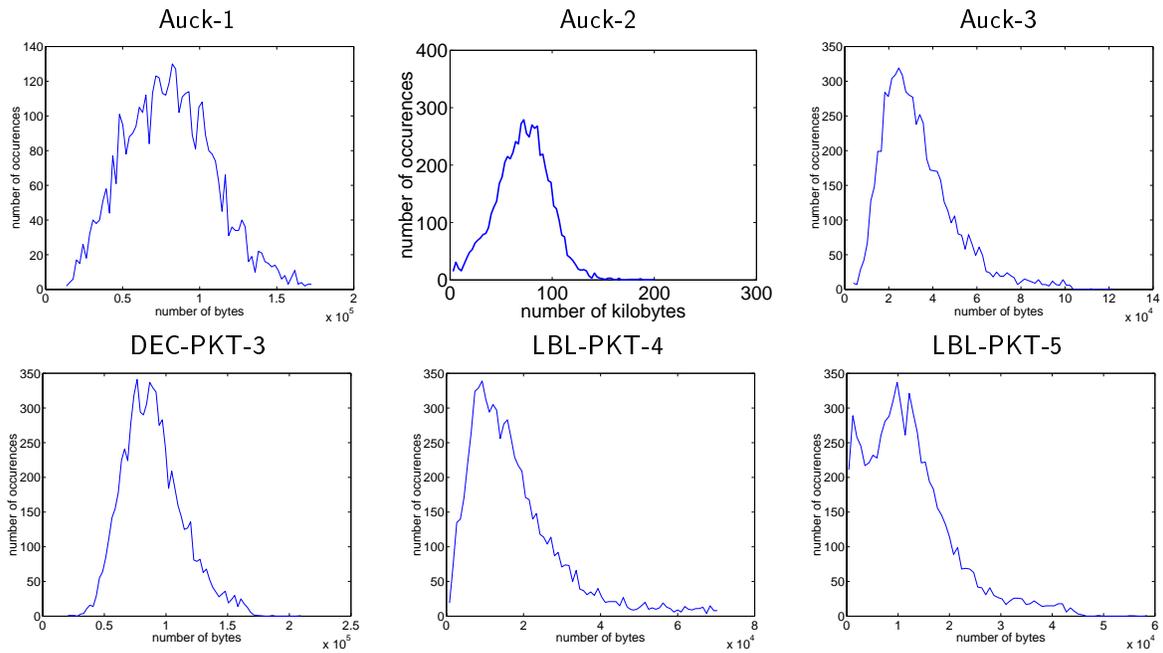


Figure 11: Histograms of Beta component of traffic at 500ms for all traces

5. At other times, the link utilization is low. Since both traces record the flows in the forward and reverse directions, we infer that there is more data flow in one direction than in the other.

In Figure 12(b) and Figure 12(e), we plot the kurtosis of the overall traffic and that of the Beta component, on

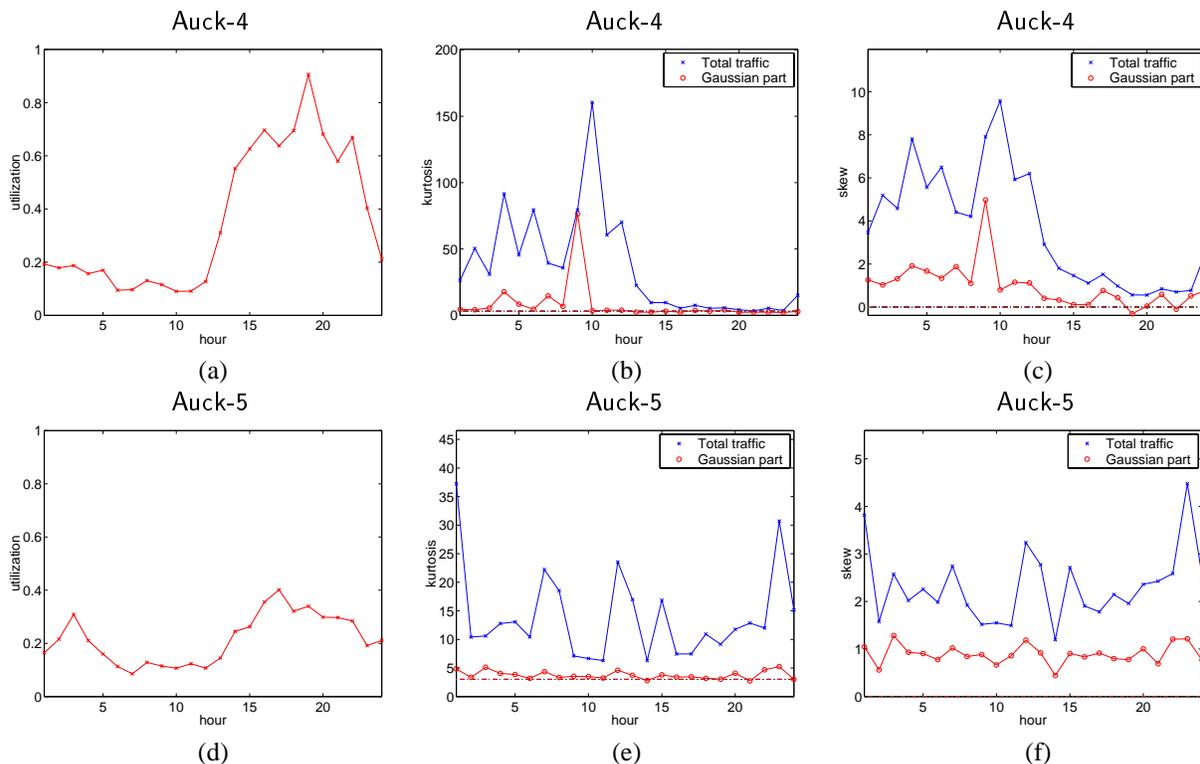


Figure 12: Hourly variation of burstiness. Hour 0 corresponds to Auckland local time 19:25:48, on December 1, 1999.

an hourly basis. The traces are studied at 500ms resolution. There is a clear relation between kurtosis of the overall traffic and the link utilization. With more link utilization, the overall traffic tends towards Gaussian. Also, the Beta component of the traffic has kurtosis closer to 3 when at high utilization. Figure 12(c) and Figure 12(f) show the skew variation during the course of 24 hours. Note that for the most part the skew is positive. This could be due to the fact that the traffic trace is positive, and hence the tail on the positive side of the distribution is more than that on the negative side.

4 Origin of Alpha traffic

In this section, we investigate the possible causes of burst causing connections. We list four reasons why we could observe a burst causing connection. A simple test reveals that all the burst causing connections in the real traces can be attributed to only one cause, namely *heterogeneity in bottleneck bandwidths*.

4.1 Potential causes of bursts

Burst causing connections can arise due to several reasons. An exhaustive list of such reasons is given below:

- **Transient response to re-routing:** This could happen if packets from a connection are re-routed from a high-bandwidth end-to-end path to a low-bandwidth one. TCP, which probes for the available bandwidth by adjusting its window size will find that the optimal window size to use in the new route is far less than the current window size, which was suitable in the old route. Since feedback to TCP takes at least an RTT, we could expect a transient bursty behavior.
- **Transient response to start/stop of connections:** Whenever other connections sharing the same link with a particular connection terminates, it causes freeing up of available bandwidth which can be used by the competing

connections. TCP will sense the increase in available bandwidth and try to grab its share. This might potentially lead to bursty connections, especially for those connections where the phase of TCP is conducive to sudden increase in window size, like slow-start.

- TCP slow-start peculiarities: In this case, some connections can get “lucky” during slow start in that it encounters no packet drops for unusually long time. This can happen when packet drops at congested routers happen only for the competing connections, albeit with very low probability.
- Heterogeneity in bottleneck bandwidths: In this scenario, the connections active in the measured link can be bottleneck-ed elsewhere. The bottlenecks need not be the same link, hence need not be similar. We can imagine a wide range of bottleneck connections, ranging from slow modem lines to fast ones such as DSL and Ethernet. When we have a large pool of connections, we would expect that the high bottleneck connections will dominate over the low-bottleneck connections and could potentially cause bursts. This scenario assumes that there are only a few connections that have high bottleneck bandwidth.

We will argue that the last scenario, i.e., *the heterogeneity in bottleneck bandwidths* is solely responsible for the bursts observed in the traces we analyzed.

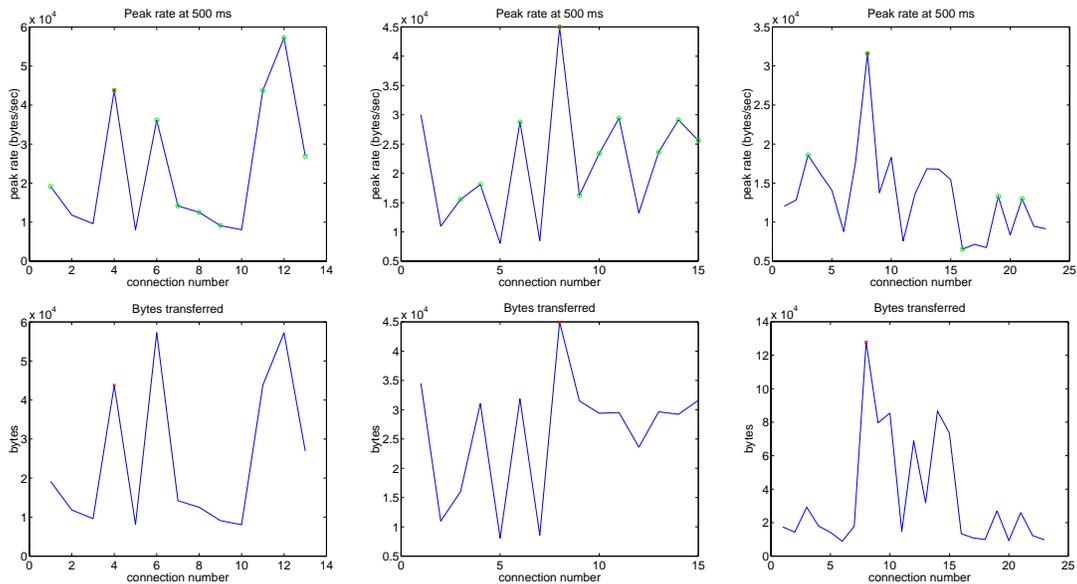


Figure 13: Top: Peak rates for connections that share the same source and destination hosts. Three such groups are shown. Bottom: Amount of data transferred in these connections

4.2 Testing for the origin of burstiness

To support our claim that *the heterogeneity in bottleneck bandwidths* is solely responsible for the observed bursts we propose a test to identify whether the cause of burstiness is related to the end-to-end path. Among the four possible causes of burstiness listed above, only the last one depends on the end-to-end path. For the traces analyzed here, the tests were positive, the burst causing connections are due to end-to-end properties, and thus we can attribute the burst phenomenon to the heterogeneity in bottleneck bandwidths.

The test involves selecting a burst causing connection and identify all connections that share the same source and destination hosts. We filter out those connections that do not transmit more than a given amount of data, since these connections cannot potentially create a burst. For each of the connections selected, we determine the maximum data rate for the time period under study by sliding a moving window along the connection duration. We plot the peak rate for all the connections selected. If burstiness is caused due to an end-to-end phenomena (like heterogeneity in

bottleneck bandwidths), we would expect all the selected connections to be high rate, since they share the same end-to-end path with a burst causing connection. On the other hand, if the burstiness is due to other reasons (like transient responses, etc.) then we would expect the initial bursty connection to be the only one among the selected connections to be high rate.

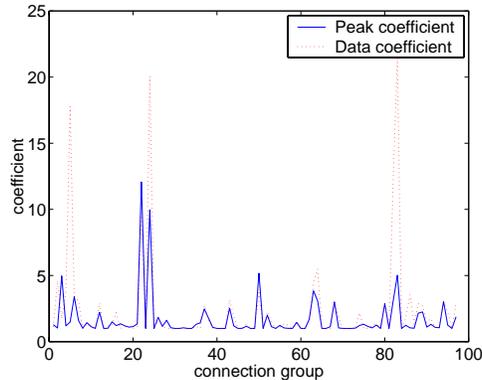


Figure 14: Peak rate and data rate coefficients

Figure 13 shows the results for a few groups of connections in the Auck-2 trace. We observe that all connections within a group are of more or less equal strengths. We also plot the total bytes that were transferred in the connection. Note that the peak-rate plot and the total-data plot are highly correlated. This suggests that many of the connections did not utilize the full available bandwidth, because the entire file transfer was over by the time TCP could probe for the optimal bandwidth.

If a burst causing connection is not due to heterogeneity in bottleneck bandwidths, we would see a high peak rate for this connection compared to other connections sharing the same end-to-end hosts. This motivates a measure that can pick out such connections. We define peak-rate-coefficient (PRC) as the ratio between the largest and second largest peaks in an end-to-end connection group. PRC would thus be much larger than 1 if the cause is not heterogeneity in bandwidths. However, we can also expect large PRC when we have different file sizes. In other words, the other connections in the connection group can have low peak rates due to small data transfers in those connections. To tackle this problem, we define another measure, data-rate-coefficient (DRC), as the ratio of the bytes transferred in the corresponding connections. If the increase in PRC was due to an increase in DRC, we can deduce that burstiness is due to an end-to-end phenomena, namely heterogeneity in bottlenecks. Indeed, this was the case observed, as shown in Figure 14. Thus we conclude that all burst causing connections are due to bottleneck heterogeneity.

5 Connection-level traffic model

We propose a network model to explain the origins of burst causing connections. The model is validated using simulations in *ns*, a network simulator. We have seen in the previous chapter that the existence of bursts in real traces cannot be explained in the framework of classical ON/OFF model. The burst causing connections, which send data at rates considerably higher than other connections, form a small fraction of the total connections. This motivates a model for the connections where the majority of TCP connections have similar bottleneck bandwidths in their end-to-end path, while a small number of connections have high bottleneck bandwidths. The first kind of connections aggregate to a fGn process plus a mean (the Beta component), and the second kind of connections add up to give the burst component of the traffic. We model the connection durations as taken from heavy tailed Pareto distribution [26], to impart LRD to traffic. Thus we argue that the heterogeneity in both bottleneck bandwidths and file-sizes gives rise to the conspicuous features of realistic traffic namely burstiness and LRD. We simulate the above scenario in the well known network simulator *ns* and show that it generates realistic traffic that is both LRD and non-Gaussian. *ns* [7, 23] is LBNL's event driven simulator derived from S. Keshav's REAL network simulator.

5.1 Connection classes

The heterogeneity in file-sizes and bottleneck bandwidths motivates a classification of TCP connections into four classes:

- Class 1: Low bottleneck rate and small connection load. We define connection load as the amount of data transferred in that connection. This class includes data transfers of a few packets, like SMTP (email without attachments), web traffic with small number of packets transferred, etc. We can also include telnet sessions and ACK streams in this class, since they do not compete for bandwidth.
- Class 2: Low bottleneck rate and large connection load. Connections in this class are usually long HTTP connections, SMTP (email with attachments), FTP, etc that are bottleneck-ed by a slow link in the end-to-end path.
- Class 3: High bottleneck rate and small connection load. Here, the individual connections are short and spiky.
- Class 4: High bottleneck rate and large connection load. These connections are large file transfers similar to case 2 but where the bottleneck bandwidth is considerably higher than those of the other connections. The bottleneck for these flows could be the link where the measurements are taken. These connections are the candidates that can cause burstiness.

5.2 Network model to explain non-Gaussian LRD traffic

Based on our careful analysis of several real traffic traces presented in Chapter 3 and 4, we propose a network model that gives rise to non-Gaussian LRD traffic as observed in the real world. The model introduces heterogeneity in the bottleneck bandwidths of the TCP connections. This hypothesis agrees with the variations in link capacity in real world networks, from analog modems to T1 dedicated phone lines. The LRD property is brought about by the heavy tailed distribution of file sizes. This model is verified in section 5.3 using ns simulations.

We model the different bottleneck constraints as follows. The majority of the connections have low to medium bottleneck, whereas very few connections have high bottleneck capacities. The connections from small bottlenecks aggregate to give a Gaussian traffic, and can be well modeled as fGn. The high bottleneck connections, on the other hand, arrive infrequently and dominate the instantaneous arrival rates, causing bursts.

The proposed model captures both the local-topology information as well as the user behavior. Local-topology information is used to determine the values of the bottleneck bandwidths. The user behavior is captured by the inter-arrival of the bursts and the Hurst parameter of the Beta component of traffic. Adjusting these parameters give rise to simulated traffic to match any realistic traffic trace.

5.3 Simulations to validate the network model

The objective in this section is two fold: to further validate the proposed network model using ns simulations, and to have a simulation setup that produces realistic non-Gaussian LRD traffic.

The arguments in the previous section motivates a simulation setup in which majority of the TCP connections have similar bottleneck bandwidths, and a small number of connections having much higher bandwidths. We adopt the topology in Figure 15. The network has 40 servers and a large number of clients. The number of clients is chosen based on the intensity of traffic that we wish to simulate. The links $D - P_i$ serve as the bottlenecks for the end-to-end path. Each of the link $D - P_1$ to $D - P_n$ has bandwidth taken from a uniform distribution. The link $D - P_0$ has much higher bandwidth, of about 100 to 1000 times those of $D - P_i$, $1 \leq i \leq n$. The TCP connections that use this link as the bottleneck will serve to create burstiness in the aggregate traffic observed at link $A - B$.

The servers establish TCP connections with a client on the other side that is chosen at random. The connection times and the idle times are taken from Pareto distribution with parameter $\alpha = 1.2$ (except for servers connected to the large bottleneck link $D - P_0$). The relationship between the expected Hurst parameter of the aggregate traffic and α is given by [26]

$$H = \frac{(3 - \alpha)}{2} \quad (9)$$

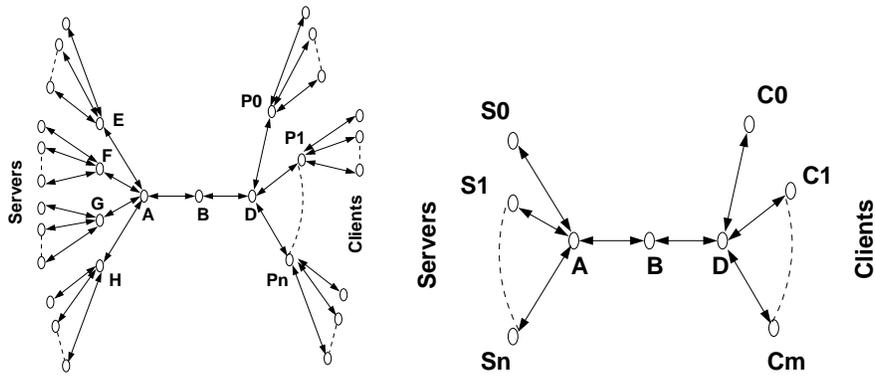


Figure 15: Left: *ns* topology to generate realistic traffic. Right: *ns* topology commonly used

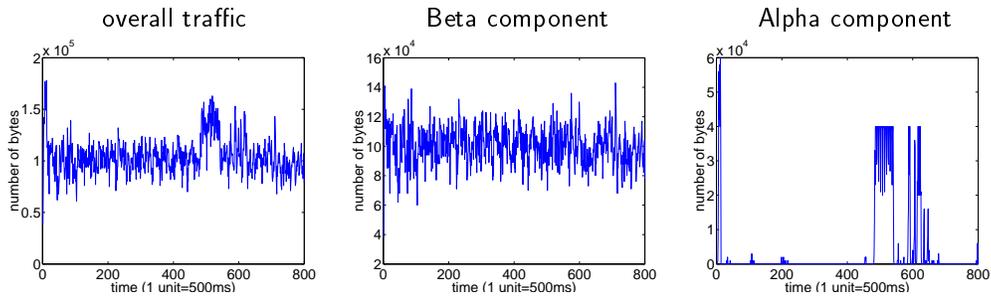


Figure 16: Bytes-per-time arrival process at 500ms aggregation level for (a) Aggregate network traffic (Auck-2) (b) The Beta component and (c) Alpha component. The Alpha component comprise of bytes from a single connection at each time bin.

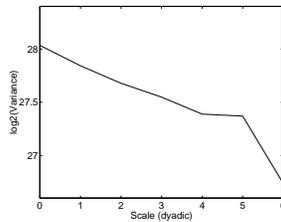


Figure 17: Variance time plot for *ns* simulation. The Hurst parameter is 0.92.

Overall traffic that flows through link $A - B$ is measured and aggregated over time-bins of 50ms and 500ms. Figure 16 gives the bytes-per-time plot for the series obtained from simulation, with parameters taken from Table 10.

We note that the visual appearance of the traffic resembles realistic traffic. The kurtosis for this trace is 4.80 and the Hurst parameter 0.92, close to the value of 0.9 predicted by equation 9. If the burst causing connections are removed from the trace, the kurtosis reduces to 3.06. This is consistent with the previous observations that the bursts are caused due to single connections.

We mention that the often used *ns* topology, shown in Figure 15 does not give traffic with kurtosis significantly greater than 3. Thus we conclude that heterogeneity in bottleneck links are essential for producing realistic bursty traffic.

Table 9: Link parameters used in the ns simulation

Link	Bandwidth (Kbps)	Latency (ms)
$D - P_0$	12000	20
$D - P_1$ to $D - P_n$	Unif(50,120)	20
$P_i - \text{clients}$	5000	10
$A - B$	20000	20
$B - D$	20000	20
$A - E$ to $A - H$	10000	20
$E - \text{servers}$ to $H - \text{servers}$	10000	Unif(10,100)

Table 10: Network topology parameters for heavy traffic

Number of nodes P_i	15
Number of clients per P_i	6
Mean ON time (sec)	10
Mean OFF time (sec)	10
Pareto parameter α	1.2

6 Impact and applications of model

6.1 Impact on queueing

The importance of queueing analysis in network design and control cannot be overemphasized. Buffer dimensioning in routers and call admission control are but two of the many crucial areas in networking research that rely on an accurate characterization of the queueing behavior of data traffic. We begin by showing that the Alpha component of network traffic significantly affects the tail queue probability, it contributes only a small fraction of the aggregate traffic load. The Alpha component dictates the tail queue behavior for large queue sizes, whereas the Beta component controls the tail queue behavior at small queue sizes.

We perform queueing experiments with the real data to study the impact of the Alpha and Beta components of the traffic separately. The traffic is fed to an infinite queue with constant service rate. We then determine the probability that the queue size exceeds a given value. We consider only the Auck traces, since the link capacity for these traces is known, and is equal to 2Mbps. First we feed the queue with the overall traffic and study its tail queue behavior. Then we feed the queue with the Alpha component *plus* the mean of the Beta component, to study the impact of burstiness. The mean of Beta component is added in order to keep the link utilization equal. Similarly, to study the impact of the Beta component on queueing, the queue is fed with the Beta component plus the mean of the Alpha component.

The impact of the Alpha component of the real data is considerable, as we demonstrate in Figure 18. We plot the queue size versus tail-queue probability for the Auck traces. The plots reveal that the Alpha component significantly affects the queueing behavior, especially for large queue sizes. This is remarkable, considering that the Alpha component constitutes only a small fraction of the total load. The Beta component determines the queueing behavior at small queue sizes. Note that for the Auck-3 trace, the Beta component of the traffic does not fill up the queue at all, although it constitutes 94 percent of the overall traffic. This is due to low link utilization (as we saw in Chapter 3).

6.2 Multifractal spectra of traffic

In this section, we study the multifractal spectra of network traffic and its components. In very rough terms, the multifractal spectrum $f(a)$ of a function $A(t)$ describes the singularity structure, i.e., the burstiness of A in a compact way (see Figure 19): the parameter a measures the strength of a burst, while $f(a)$ measures how frequency it occurs (see [29, 28]).

The spectrum f is often estimated through the Legendre path which seems the only practical way but which,

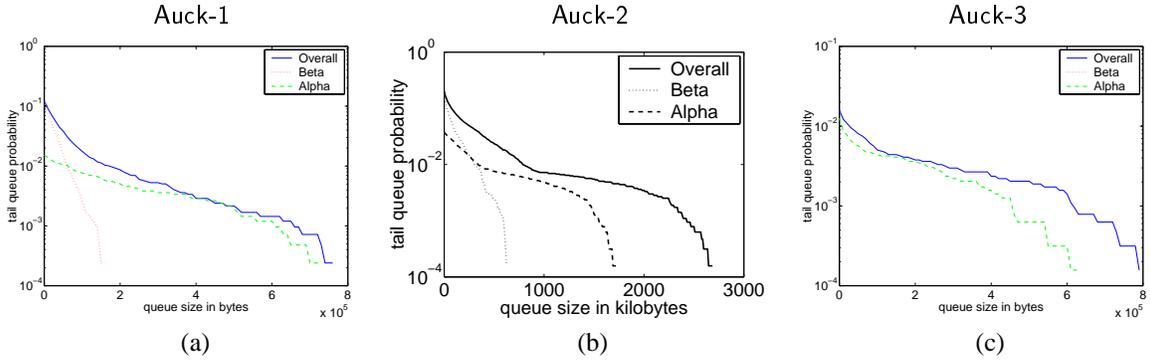


Figure 18: Queueing behavior of traffic.

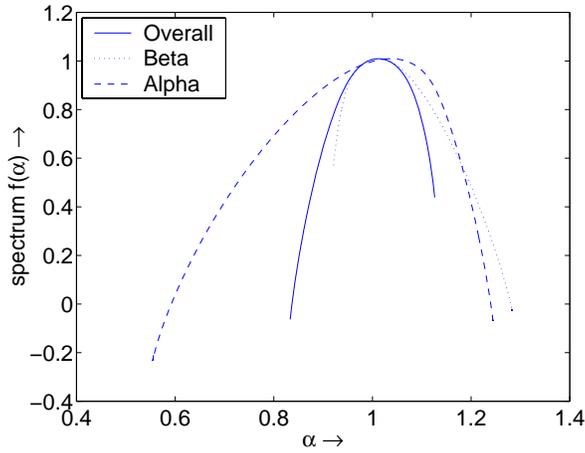


Figure 19: Multifractal spectra of traffic

unfortunately, increases the estimation error and adds bias. In particular, the Legendre estimate of f is always very smooth, while the true f might not be.

Let us first comment on the shape of the spectra of the components. The Beta component shows a rather narrow spectrum, which could roughly be approximated by two line segments, one joining $(9.9, .5)$ and $(1, 1)$, the other $(1, 1)$ with $(1.3, 0)$. Such estimates are typically observed with fGn, thus confirming well the Gaussianity of this component. Since other processes might show similar spectra, the shape of the spectrum should not be seen as a proof of the Gaussianity. Nevertheless, we may conclude that fGn is an accurate model for the Beta component not only in what concerns marginals and LRD, but also small scale properties.

The spectrum of the Alpha component, on the other hand, is broad, and shows in particular considerably large $f(a)$ values for a below 1, this reflects the bursty behavior of this component. When comparing with the spectrum of the overall traffic it is immediately evident that the bursty exponents ($a < 1$) of the overall traffic stem from the Alpha connections. Theory says (see [30, Section 4]) that the maximum of the spectra of the components, i.e. $\max(f_\alpha(a), f_\beta(a))$ should have the same convex hull as $f(a)$, the spectrum of the overall traffic. However, since the Alpha traffic is so small in volume estimation errors prevent it from pushing up the left part of $f(a)$ as much as theory predicts. Nevertheless, theory confirms that the bursts of the traffic is to an overwhelming degree due to the Alpha component.

7 Discussion and Conclusions

7.1 Summary

In this report, we verified the following principle for burstiness in network traffic:

“In any series of elements to be controlled, a selected small fraction in terms of number of elements almost always accounts for a large fraction in terms of effect.” —*Pareto’s principle*

A connection level analysis of network traffic reveals that the burstiness is caused by a small number of connections that transmits data at unusually high rates, thereby verifying Pareto’s principle in the context of network traffic modeling.

Based on very careful analysis of several real traffic traces, we infer that Internet traffic can be separated into two components: the Beta component and the Alpha component. The Alpha component is due to the small number of connections that give rise to bursts. The Beta component, which is composed of the majority of the connections captures the LRD and is well modeled as fGn plus mean.

A physical model which captures heterogeneity in bottleneck bandwidths and size of files transferred was introduced to explain the Beta and Alpha components of the traffic. *ns* simulations were performed which support the model.

Queueing experiments were done to determine the impact of the two components. Whereas Beta component affects small queue sizes, the Alpha component controls the queueing behavior at large queue sizes.

7.2 Future work

The immediate goal is to estimate the bottleneck bandwidth of individual connections. This is a challenging problem because we need the connections to send a large number of packets in order to accurately estimate the maximum rate.

It would be interesting to analyze the Alpha component of traffic and study the inter-arrival times of the burst-causing connections. Preliminary investigations indicate that burst-causing connections arrive in a non-Poisson fashion. Efforts to generalize the process such as non-homogeneous Poisson process or Weibull is underway.

A novel technique to analyze and separate Gaussian and non-Gaussian components of data using Independent Component Analysis (ICA) [11] is planned.

An important item for future work is synthesis of network traffic using the fGn plus bursts model, trained on real data sets. The synthetic traffic must match the real traffic in terms of queueing behavior.

The analysis of burstiness can be used in Active Queue Management (AQM) policies. For example, in schemes such as Random Early Drop (RED), we can drop packets that belong to burst-causing connections instead of dropping packets randomly. In this scheme, only the burst-causing connection is penalized.

A Tests for Gaussianity using Kurtosis and Skew

A.1 Kurtosis

The classical measure of non-Gaussianity is *kurtosis*. The kurtosis of a random variable X is the fourth central moment divided by fourth power of the standard deviation.

$$K(X) = \frac{E(X - \bar{X})^4}{(E(X - \bar{X})^2)^2} \quad (10)$$

For a Gaussian X , the fourth central moment equals $3((E(X - \bar{X})^2)^2)$. Thus, the kurtosis $K(X) = 3$ for a Gaussian random variable. For most non-Gaussian random variables, the Kurtosis differs from 3.

Random variables that have Kurtosis less than 3 are called *sub-Gaussian*, and those with Kurtosis greater than 3 are called *super-Gaussian*. In statistical literature, the corresponding expressions *platykurtic* and *leptokurtic* are used. Super-Gaussian random variables have typically a “spiky” probability density function (pdf) with heavy tails, i.e., the pdf is relatively large at large values of the variable. Sub-Gaussian random variables, on the other hand, have typically a “flat” pdf, in which pdf is small for values far away from the mean when compared to a Gaussian random variable. Figure 20 illustrates the differences between super-Gaussian and sub-Gaussian random variables.

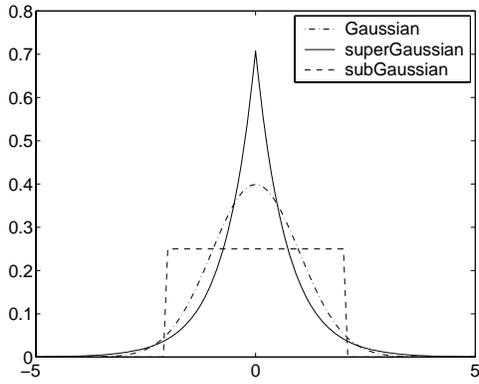


Figure 20: Examples of sub-Gaussian and super-Gaussian distributions

Typically non-Gaussianity is measured by the deviation of Kurtosis from 3 [11]. There are non-Gaussian random variables that have a Kurtosis of 3 but they can be considered as very rare. An example of a random variable with kurtosis 3 is the Weibull distribution with shape parameter 2.2 (approx.) [12].

The main reason for using Kurtosis as a measure of non-Gaussianity is its simplicity, both computational and theoretical. Computationally, Kurtosis can be estimated simply by using the fourth and second central moments of the sample data. Theoretical analysis is simplified because of the following linear property: If X_1 and X_2 are two independent random variables with unit variance, then the following relations hold:

$$(K(X_1 + X_2) - 3) = (K(X_1) - 3) + (K(X_2) - 3) \quad (11)$$

and

$$(K(\alpha X_1) - 3) = \alpha^4 (K(X_1) - 3) \quad (12)$$

where α is a scalar. These properties can be easily proven using the definition.

Kurtosis has also some drawbacks in practice, when its value has to be estimated from a measured sample. The main problem is that Kurtosis can be very sensitive to outliers [10]. Its value may depend only on a few observations in the tails of the distribution. In other words, Kurtosis may not be a robust measure of non-Gaussianity.

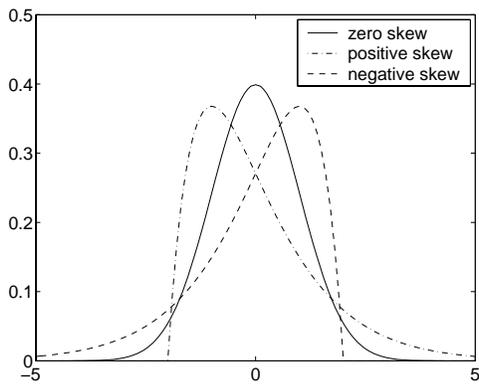


Figure 21: Distributions with positive and negative skew

A.2 Skew

Skew is defined as the third central moment divided by the cube of the standard deviation.

$$S(X) = \frac{E(X - \bar{X})^3}{(E(X - \bar{X})^2)^{\frac{3}{2}}} \quad (13)$$

A distribution is skewed if one of its tails is longer than the other. A skew value of zero indicates that the values are evenly distributed on both sides of the mode. A negative skew indicates an uneven distribution with a higher than normal distribution of values to the right of the mode, a positive value for the skew indicates a larger than normal distribution of values to the left of the mode. Figure 21 illustrates this with examples.

References

- [1] S. Bates and S. McLaughlin. An investigation of the gaussian assumption for self-similar teletraffic models. In *Proc. of IEEE Sig. Proc. Workshop on Higher Order Statistics*, pages 444–447, July 1997.
- [2] F. Brichet, J. Roberts, A. Simonian, and D. Veitch. Heavy traffic analysis of a fluid queue fed by a superposition of ON/OFF sources. *COST*, 242, 1994.
- [3] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic. Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5:835–846, December 1997.
- [4] A. Erramilli, O. Narayan, A. Neidhard, and I. Sanjeev. Performance impacts of multi-scaling in wide area tcp/ip traffic. In *Proceedings of the IEEE INFOCOMM'2000, Tel Aviv*, 2000.
- [5] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent traffic. *IEEE/ACM Trans. Networking*, pages 209–223, April 1996.
- [6] Ashok Erramilli, Onuttom Narayan, and Walter Willinger. Experimental queueing analysis with long-range dependent traffic. *IEEE/ACM Trans. Networking*, pages 209–223, April 1996.
- [7] K. Fall and K. Varadhan. ns notes and documentation. <http://www-mash.cs.berkeley.edu/ns>, 2000.
- [8] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. *Proc. ACM/Sigcomm 98*, 28:42–55, 1998.
- [9] P. Flandrin. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Trans. Inform. Theory*, 38(2):910–916, Mar. 1992.
- [10] P. J. Huber. Projection pursuit. *The Annals of Statistics*, pages 435–475, 1985.
- [11] A. Hyvarinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.
- [12] N. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1-2. John Wiley & Sons, New York, 1994.
- [13] I. M. Johnstone and B. W. Silverman. Wavelet threshold estimators for data with correlated noise. *J. Royal Stat. Soc. B*, (59):319–351, 1997.
- [14] L.M. Kaplan and C.-C.J. Kuo. Fractal estimation from noisy data via discrete fractional Gaussian noise (DFGN) and the Haar basis. *IEEE Trans. Signal Proc.*, 41(12):3554–3562, Dec. 1993.
- [15] L.M. Kaplan and C.-C.J. Kuo. Extending self-similarity for fractional Brownian motion. *IEEE Trans. Signal Proc.*, 42(12):3526–3530, Dec. 1994.
- [16] LBL. Internet traffic archive. <http://ita.ee.lbl.gov/html/traces.html>.
- [17] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic. *IEEE/ACM Trans. Networking*, 2(1):1–15, Feb. 1994.
- [18] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Networking*, pages 1–15, 1994.
- [19] N. Likhanov, B. Tsybakov, and N. Georganas. Analysis of an ATM buffer with self-similar input traffic. *Proc. IEEE, Info com '95 (Boston 1995)*, pages 985–992, 1995.
- [20] S. Ma and C. Ji. Modeling video traffic in the wavelet domain. In *Proc. of 17th Annual IEEE Conf. on Comp. Comm., INFOCOM*, pages 201–208, Mar. 1998.
- [21] B. Mandelbrot and J. W. Van Ness. Fractional Brownian motions, fractional noises and applications. *SIAM Review*, 10(4):422–437, Oct. 1968.

- [22] B. B. Mandelbrot. Long-run linearity, locally gaussian processes, ii-spectra and infinite variances. *International Economic Review*, 10:82–113, 1969.
- [23] S. McCane and S. Floyd. ns- network simulator. <http://www-mash.cs.berkeley.edu/ns>.
- [24] NLANR. Auckland-II trace archive. <http://moat.nlanr.net/Traces/Kiwitraces/>, 2000.
- [25] I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.
- [26] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols*, pages 171–180, 1996.
- [27] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Trans. Networking*, 3(3):226–244, June 1995.
- [28] R. H. Riedi. Multifractal processes. in: “*Long range dependence : theory and applications*”, eds. Doukhan, Oppenheim and Taqqu, to appear 2001.
- [29] R. H. Riedi, M. S. Crouse, V. Ribiero, and R. G. Baraniuk. A multifractal wavelet model with application to TCP network traffic. *IEEE Trans. Inform. Theory*, 45(3):992–1018, April 1999.
- [30] R. H. Riedi and W. Willinger. *Self-similar Network Traffic and Performance Evaluation*, chapter Toward an Improved Understanding of Network Traffic Dynamics, pages 507–530. Wiley, 2000.
- [31] M. Taqqu and J. Levy. *Using renewal processes to generate LRD and high variability*. In: Progress in probability and statistics, E. Eberlein and M. Taqqu eds., volume 11. Birkhaeuser, Boston, 1986. pp 73–89.
- [32] M. Taqqu, V. Teverovsky, and W. Willinger. Estimators for long-range dependence: An empirical study. *Fractals.*, 3:785–798, 1995.
- [33] A. Tewfik and M. Kim. Correlation structure of the discrete wavelet coefficients of fraction Brownian motion. *IEEE Trans. Inform. Theory*, 38(2):904–909, Mar. 1992.
- [34] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Trans. Networking (Extended Version)*, 5(1):71–86, Feb. 1997.
- [35] G. W. Wornell. A Karhunen-Loève like expansion for $1/f$ processes via wavelets. *IEEE Trans. Inform. Theory*, 36(2):859–861, Mar. 1990.