Classifying Daily Patterns In Long Duration Network Traces

A thesis

submitted in fulfilment

of the requirements for the degree

of

Master of Science

at the

University of Waikato

by

Brendon Jones

Department of Computer Science

Hamilton, New Zealand

March 15, 2006

Abstract

Until recently there has not been a set of network traces spanning more than a few months available for researchers to use, which has perhaps limited understanding of long term trends and behaviours within computer networks. With the recent completion of the Waikato dataset by the WAND Network Research Group, such a trace archive now exists. This thesis is an investigation into the aspects of traffic and methods of describing it that can be used to identify and represent discrete "types" of days within these traces, and what differences in what quantities are important to distinguish between them. It is also the first in depth look at the long term data present in the Waikato dataset. The information gained is aimed towards use in simulation such that a generic day of a certain pattern may be simulated without being explicitly based on one particular day, or overfitted because of some chance observation in a single day. A number of traffic models are configured using parameters derived from different types and are evaluated using the network simulator *ns-2* and the same methodologies that created the initial types.

Acknowledgements

I would like to acknowledge the members of the WAND group for their support during this project. Perry Lorier deserves special mention for making available the new traffic generator that is employed herein and his assistance with making it work within *ns-2*.

Extra thanks should go to my supervisor Richard Nelson for his guidance and patience throughout the last year.

Contents

List of Figures

Chapter 1

Introduction

There are a lot of aspects of network traffic that can be measured; commonly data rate, packet rate, packet interarrival time and the number of flows[4] in a time interval. These can vary depending on the location of the measurement point and the time of day the measurements are taken. If either the location or the time change, then it is expected that the traffic will be different – the network traffic of a large organisation is very unlikely to ever look like like that of a home user on a dial-up connection, and there is usually less traffic overnight than there is during the day. However, it is not as clear what happens if *nothing* is changed and measurements are taken at the same place and same time, over a long period. Is one day very much like the next, or do they bear no relation to one another? The situation is not always the same because users have different schedules from day to day, and events such as holidays will change usage patterns, but these are made much easier to investigate without having to deal with the problems caused by changes in the way that measurements are collected.

The exciting thing about being able to examine this data that has been collected consistently in the same place at the same time over a long period is that it can be used to make statements about what the characteristics of an “average” day are, how this relates to the actual observed days, and if it is even possible to claim that any one day (or composite day) can be representative of what is seen at that measurement point.

The aim of this project is to characterise the traffic that is seen each day in such a way that allows for comparison between different groupings and gives a clear reason as to why each day belongs to the group it does – there needs to be knowledge of what constitutes any of the "typical" types of days, which can then be simulated using the factors that describe them. Breaking data into similar groups that are representative of the different ways that traffic is observed would be useful for simulation. The signature of a desired type of day can be used without being too specific to any particular day within that designation.

Using traces from the new Waikato dataset means that nearly two years worth of contiguous data is available for inspection. Common patterns of network usage at the University of Waikato where the data was captured should stand out and be supported by many examples, which should ease finding attributes that are strong predictors of the pattern type. Here the discussion is limited to the subset of traffic in the trace files that is usually known as Hyper Text Transfer Protocol (HTTP) or "web". In practical terms, the programs used to examine traffic were restricted to packets with a source or destination TCP port of 80 (the Internet Assigned Numbers Authority assigned port for HTTP). Besides peer-to-peer and streaming media, HTTP makes up most of the volume of traffic on Internet links[7], and on the University of Waikato uplink it is the single largest traffic type. An understanding of HTTP and good models of it is important for any simulation that intends to represent parts of the Internet, and vitally important for any that intend to represent the University. HTTP is arguably one of the more simple protocols seen on the Internet, and once the problem is solved for it then the lessons learnt can be applied to other traffic types, and to combinations of traffic types. The methods of describing and predicting HTTP will provide a useful starting point from which to investigate the remainder of the traffic seen. The attributes used to describe the traffic have a strong effect on what is actually produced[10].

Being able to extract the important features that define traffic over the course of a day

has other applications beyond generating and validating simulations. By recognising patterns of attributes that represent normal and unusual behaviour, attacks on network attached hosts can be detected and someone alerted to deal with the problem. Broken devices or other changes within the network that can negatively impact traffic can also be detected due to their deviations from usual patterns or falling into known bad ones. Large datasets can be helpful in providing baseline data with which to compare new data against and from which a sense of normalcy can be established and patterns built.

Chapter 2

Background

It is not until recently that network traces covering long periods of time have been available, and therefore there is not a lot of previous work to build on in investigating how networks perform and change over time. The idea of comparing two different distributions of measured network traffic has been discussed by groups such as the IP Performance Metrics Working Group[9] of the Internet Engineering Task Force for purposes of comparing different implementations of their metrics[11]. They need to show that two implementations recording network statistics for the same links give results that agree with each other in order to show they are interoperable. Other fields make use of comparisons between some sort of historical or expected data and what is actually observed in order to detect unusual network events that could be of interest to a network operator. These are often part of Intrusion Detection Systems (IDS) and can operate online using live data, or offline using stored network traces.

2.1 Network Trace Files

Network traces are records of the packets that traversed a network at a point where a monitoring machine is installed. Capturing these traces can, depending on the amount of each packet stored, show the types of traffic that users are doing, who they are com-

municating with, and what they are communicating. Looking at long term trends and variation requires data covering a long time period. Where previously, network trace files available to researchers have been limited to a small number of consecutive days (up to 45 in the case of the longest Auckland dataset from WITS[26], and a few months for the venerable Bellcore Ethernet traces[14]), or a stratified sample taken at different times of day, the *Waikato 1* Dataset[18] is a continuous capture of all network traffic entering and leaving the University of Waikato from December 6 2003 to August 17 2005. The WAND Network Research Group[27] at the University was responsible for the design and installation of the capture point.

A dedicated machine on the edge of the University network performed the capture using an Endace DAG3.5E[6][8] capture card. The resulting trace files contain a timestamp indicating the capture time of each packet, the IP header and any further protocol headers that the packet had such as ICMP, UDP, or TCP. Privacy issues mean that the addresses in the IP header (and fields that relate to them) are anonymised, but the encryption algorithm used ensures a one to one mapping between raw and anonymised addresses – the endpoints of packet flows will remain constant, and machines that do high volumes of traffic remain visible as important sources/sinks even after anonymisation. None of the packet payload is stored. This allows detailed examination of protocol mixes, network stack interactions, and host/application behaviour without compromising the privacy of network users.

The compressed headers for more than 600 days worth of data take up nearly one and a half terrabytes of space on disk and contain an incredible number of diverse packets, so WAND has developed tools to facilitate easier parsing of and access to data. They are released as part of a packet processing library called *libtrace*[15] that abstracts away a lot of network specific information and gives a unified interface to the important parts of the packet headers. *Libtrace* can interface with traces captured from a DAG card, or the more traditional *pcap* format used by programs such as *tcpdump*[23].

2.2 Event Detection

Automated detection of unusual network events is very important when dealing with large quantities of data as it is infeasible to have a network operator examine all of it manually. In order to determine if a network is behaving differently to expected there has to be a concept of normality to compare it to, and there have been a number of attempts made at this. Sometimes there are characteristics of the network that are known (or at least expected) to be true during normal operation and any deviation from these could be considered interesting or important, but often this is not the case and some baseline for comparison needs to be calculated. This could involve keeping track of what has happened in the past and comparing it with the current state (perhaps smoothing values and giving more weighting to those that are more recent), or even be just a set of arbitrary values decided upon that represent the range of what is believed to be sane behaviour.

2.2.1 Intrusion Detection Systems

Intrusion detection systems need to detect unusual changes in the traffic flow to pick up events such as denial of service attacks. Attacks consisting of small numbers of packets or disguised as part of an otherwise legitimate connection are usually better picked up by systems such as Bro[3] or Snort[22] that examine packet structure for patterns that are known to be exploit attempts. In the event of some attack that changes the observable characteristics of the traffic (such as greatly inflating it in order to deny room to other traffic), it needs to be distinguishable from what might be just a large volume of otherwise normal traffic so that correct action can be taken.

Data mining and machine learning techniques have been applied to this field and been moderately successful[30] at classifying traffic and spotting packet flows that do not behave as expected. Patterns can be applied over flows as a whole to match deviant

behaviours, at higher levels of aggregated traffic, or over individual packets. Many of these techniques are agnostic as to the source of their data – they only need to know what the numbers are and not what they represent, and so can be used with any sort of information that can be extracted directly or derived from the observed network traffic.

2.2.2 Network Performance

As well as detecting active attempts to attack a network, it is useful to monitor for unusual activity that has a less sinister cause. If a link or machine goes down, routes change, or if there is a lot of traffic causing overloading and packet loss, the network operator wants to be aware of this so that they can take steps to resolve the problem (or if it is outside of their control they are at least better informed). All of these things impact the traffic on a network and can be detected by deviation from the norm for that time and location. In many situations this only results in a degradation of service that is corrected without any active involvement from the network operator, in which case knowledge of the event allows them to plan for it happening again in the future and to notify customers if required. If the event is within the operators network, knowing about it as soon as possible means that identification of any faults that may have occurred can begin sooner, and the problem can be resolved sooner.

High Performance Connection (HPC) networks in the United States as well as selected sites worldwide take advantage of this with the help of the NLANR[17] Active Measurement Project[19]. Regular measurements of latency, loss and routes are taken between all sites in the mesh. They are one of the few groups to have experimented and documented their work on network event detection[16], perhaps because the large volume of information precludes doing it any other way (being a full mesh means that each site performs measurements to *every* other site). Of the statistics they collect, event detection is performed only on the round trip times and packet loss between hosts - routing changes can be inferred from changes in the base round trip time observed, or checked

for explicitly from the routing data that is collected.

When determining if there has been a significant change in the base RTT, a number of observations in the immediate past are treated as “normal” and kept in a buffer. Any new observations are compared to the mean and variance of the buffered packets and should they exceed the mean by more than an amount based on the variance they are added to both the buffer of historical data, and a separate buffer of anomalous ones. As more observations are seen that are considered normal, this second buffer empties but should it contain more than a certain number of observations it is considered enough of a change in RTT to be noteworthy. The distance from the mean allowable and the number of observations to track are all user configurable values that have recommended values based on empirical data.

This system is of interest as it assumes that future observations are likely to be similar or the same as past observations, and considers anything that breaks this assumption to be of interest. This works well in the situation it is applied to here, as links in the HPC network are rarely congested and typically have very stable round trip times. Their normal state is experienced most of the time and changes from it are events that should be noticed.

The amount of sensitivity – just how similar to the historical data current data should be – is still an open question here that is up to the individual operator to control.

Chapter 3

Overview of the Data

The data used here is a subset of the large *Waikato 1* dataset, and consists of the three hours from 1pm to 4pm for every day recorded. The reasoning behind this is that it is a period of consistent traffic that is unlikely to fluctuate greatly within each individual measurement period (there are no issues with diurnal periods, or boundaries between work- and after-hours such as around 9am and 5pm). Also, as previously stated, it is restricted to packets destined to or from TCP port 80 (web traffic). It is possible that other traffic sharing the link could impact the measurements, though this is unlikely as web traffic makes up approximately 70–80% of the total, and should another traffic type actually be significant enough to change the behaviour of the HTTP traffic then this is probably worth noting as unusual anyway.

The reduced dataset consists of 620 measurement sets, one for each day starting from December 7, 2003. Each set itself contains 10,800 measurements of one second duration each, for each statistic applied to each of the areas that are commonly identified as being key to describing aggregated traffic – the number of packets and bytes passing the measurement point, the number of flows active in that one second and the average packet interarrival time over that second.

The statistics were chosen in order to describe the shape and distribution of the data, and to measure how the data was related over time. The aim was to look at the data as

raw values as well as using the temporal information that can be gained from knowing that it is time-series, and using this as the basis for a number of comparisons.

Firstly, the summary statistics of arithmetic mean, median and standard deviation were applied to the raw data, and later after taking first differences. The Hurst parameter was used as a measure of the self similarity of the traffic, and when combined with the Kwiatkowski-Phillips-Schmidt-Shin[12] (KPSS) test for stationarity and an exponential smoothing model gives a view on the long term trends present. Direct comparisons are also made between each day and a number of "control" days using Student's t-test and the Kolmogorov-Smirnov test.

Machine learning techniques were later performed on these statistics to determine which were useful predictors when comparing days for similarity, which is described later in Chapter 5.2.

3.1 Collecting the Data

While all the trace files that make up the *Waikato 1* dataset are stored on disk and are easily accessible, they are raw packet traces and still require processing to extract the necessary data. Using the framework provided by *libtrace*, the important details are extracted from each packet, and filtered to ensure that only HTTP packets are observed. Results are collated and output once every second. Each packet seen increments a simple tally of packets serviced before the Internet Protocol (IP) header is examined for the length of the packet, which goes towards the count of bytes transferred in the measurement period. The IP header for these packets also contains a header for the Transmission Control Protocol (TCP); the source and destination addresses and ports from these headers are used to form a unique identifier that is tracked to count the number of active flows. Each packet also has associated with it a highly accurate timestamp as recorded by the DAG capture card upon receipt of the packet. Within each measurement period,

the difference between consecutive packets arrival time is noted and averaged in order to generate the mean interarrival time.

Each value is reset after the results of that measurement period have been written to disk so that the next one may begin with a clean slate. In addition, daylight saving time needed to be noted so that the same three hour period was recorded for all days throughout the year.

3.2 Statistical Measures

The usual summary statistics were the first investigated, and the easiest to calculate directly from the output of the collection process. For each of packet count, byte count, flow count and interarrival time, the mean, median and standard deviation were taken for each day, across all measurements for that day. The 10800 seconds worth of measurements per day were condensed to a single number per statistic, the resulting dataset consisting of 620 measurements for each one.

To help alleviate any problems of scale caused by basing information off of only the raw data, each of the daily runs was differenced and the mean, median and standard deviation were taken of these also. This highlights the amount of variation between consecutive measurements rather than their absolute values – for example a high mean means that there are many large changes, while a smaller one would indicate that they are more gradual and happen over a longer period of time.

Self similarity and autocorrelation within network traffic has been identified as an important part of describing and understanding its behaviour[13], and the Hurst parameter is often a good indicator of this. In this case, the Hurst parameter is derived from the slope of the Auto Correlation Function (ACF) plot after taking the logarithm of both axes (this is the method employed by the *fBasics* package from *Rmetrics*[21]). Any scores below 0.5 are considered not to show any long range dependence. The

Holt-Winters filter for time series data was also applied to investigate how much of a relationship there was between adjacent measurements and how much emphasis should be placed on previous values in order to predict future ones.

To make some direct comparisons between days, Student's t-test was used on both the raw data and the differenced data. This was rather unsuccessful in the first instance as the volume of data meant that two days must be near identical to get a result that says they are probably similar in distribution, and no pair of days came close. The differenced data had the opposite problem – all pairs of days were considered to be very similar as they had all been adjusted to essentially a zero mean and a much lower standard deviation. Because every day was considered possibly similar, this was not a useful test to distinguish between days, and so both of these tests were dropped. The Kolmogorov-Smirnov test was investigated as an alternative and performed slightly better than the t-test on unmodified data but still very few days crossed the threshold required to be considered similar. Using the differenced data had the same problem as did the t-test in that everything became too similar to tell apart. Not assuming a normal distribution helped the test somewhat in the first case, but again neither the test on the raw data or the test on differenced data seemed to distinguish well enough between days to be a useful indicator of similarity.

The Kwiatkowski-Phillips-Schmidt-Shin test was used to test the data for stationarity. The non-stationary nature of network traffic is well documented [25] and the results of applying this test to the data here agrees with this. In a number of unusual cases however, the data is actually stationary over the measurement period. In most cases the results of this test do not provide any useful information, but in these few cases it is quite an important thing to be aware of, and to classify it accordingly. Using this test on the differenced data gives even less information, as after differencing all the data becomes stationary.

3.3 Exploring the Dataset

Simple descriptive statistics are explored to see the shape of the data, and to illustrate some of the factors that influence the data. Unsurprisingly, packet, byte and flow rates are all strongly positively correlated and looking at one describes all the others well, apart from the differences in scale. Almost all the statistics measured reflect the mean rate for that particular attribute which makes the mean of any one measure a good estimator for the behaviour of any of the others.

3.3.1 Packets Per Second

The number of packets crossing the measurement point per second is a simple indicator of how busy the link is. Even on its own it should be indicative of how many users there are, and how much traffic they are doing, and most other measures are related directly or indirectly to the packet rate. The University calendar is actually a very good predictor of the level of traffic, and because of the large variation in number of users the University has, the number of packets changes accordingly.

The total measurement period covers nearly two years worth of Internet traffic at the University, and the progression from term-time to student holidays is quite evident in the graph of daily mean HTTP packet rates in Figure 3.1. The first 30 days cover the quiet period at the end of the year and the Christmas and New Years holidays, and as is to be expected packet rates are low compared to other times of the year. From there up to day 85 is Summer School which is only attended by a small number of students. The first semester of the year starts at day 86 and goes through to day 204 (including examinations) and is a time of high usage apart from the mid-semester and study breaks. The second semester follows the same pattern, and usage tails off from the end of it through to the Christmas holidays again, where the cycle begins anew.

As well as the variation over the course of the measurements, there are two clear

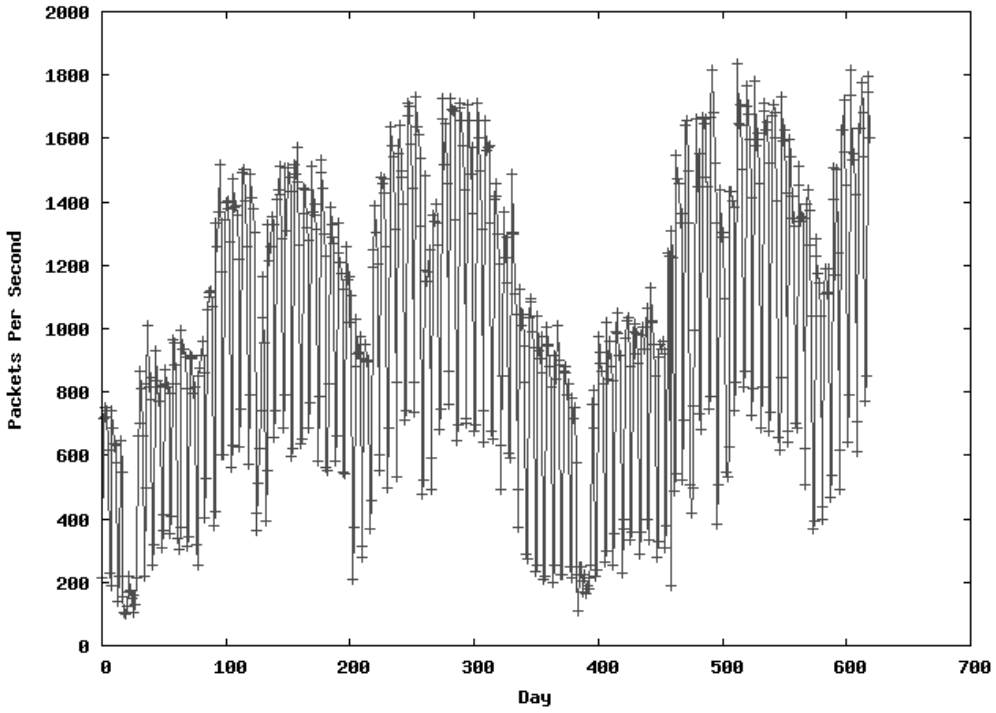
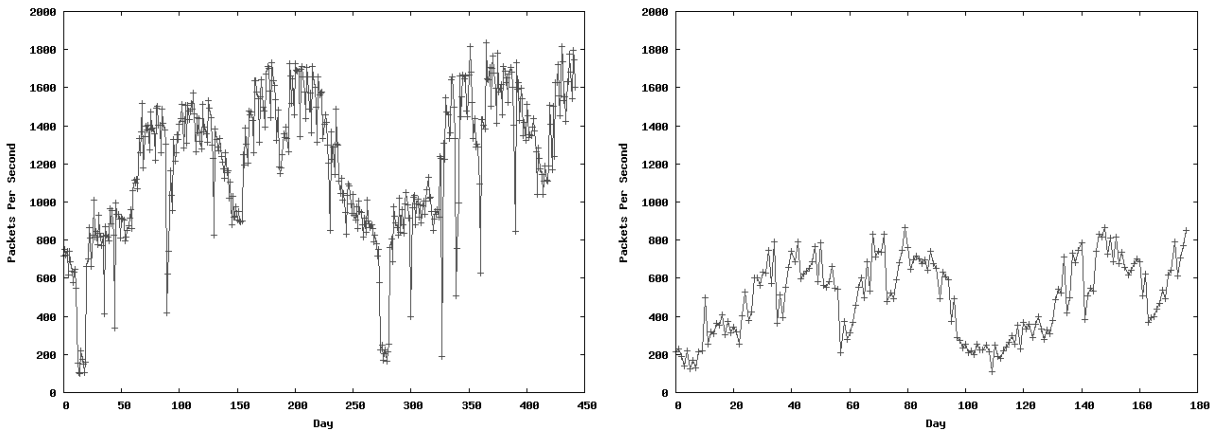


Figure 3.1: Daily Mean HTTP Packets Per Second

levels in Figure 3.1 that are mostly independent of the time of year. On average, two out of every seven days are significantly lower than the other five, and these two days are Saturday and Sunday. The amount of day to day variability can be reduced greatly by considering weekdays and weekends as separate datasets like in Figure 3.2. Each measurement is now much closer to the ones surrounding it, with the majority of differences being caused by the time of year. Perhaps of interest is that the weekend data in Figure 3.2 (b) also follows the same overall pattern of peaks and troughs which suggests that a good proportion of users are active during times when it is expected that they are away from the University. Some of this could be accounted for by automated systems such as webcrawlers and regularly downloaded software updates, but because the differences between the highs and lows are still quite pronounced, this is probably only a minority of traffic.



(a) (b)

Figure 3.2: Daily Mean HTTP Packets Per Second during (a) Weekdays, and (b) Weekends

The Hurst parameter averages 0.83 across all days measured and has a standard deviation of 0.06 – it is very consistent. The Hurst parameter is slightly lower when there are less users but there is no significant change when there are more. On the other hand, values of alpha for the exponential smoothing algorithm vary quite a lot in comparison, and appear to follow the seasonal curve. Higher values are observed in times when there are fewer students, and lower ones when there are more.

3.3.2 Bytes Per Second

Without looking at the specific data, it is sensible to assume that the number of bytes per second is positively correlated with the number of packets seen in that same time, due to constraints such as maximum and minimum packet sizes that the network will accept. Figure 3.3 plots the mean packets per second for each day against the mean bytes per second for the same day. The very clear linear trend would suggest that this assumption holds true. The average number of bytes per packet over all days is 483 with a standard deviation of 30 bytes, which is interesting when the packet sizes for the common

TCP control packets is taken into account. The three packets involved in connection establishment and a large proportion of acknowledgement packets contain no data and are therefore the minimum allowable packet size of 40 bytes (or slightly larger if there are options involved). HTTP connections tend to be very one-sided, where one side is sending the majority of the data, and so one side is likely to be sending packets as large as it possibly can while the other is sending packets as small as it possibly can in return. Assuming TCP delayed acknowledgements, then every second packet is acknowledged and the average packet size in a bulk transfer tends towards an average packet size of approximately 1013 as it increases in duration. Without delayed acknowledgements the same situation tends towards an average packet size of approximately 770 bytes. In order to bring the overall average down to 483 then there must be a lot of flows consisting only of a single response that does not fill an entire packet, or a lot of SYN packets without responses in enough quantity to counter the number packets part of bulk transfers. The number of bulk transfers is perhaps lower here than in many cases due to the University policy of charging students based on data usage – users of the network may be unwilling to bear the cost of downloading large files and restrict themselves only to web browsing.

Events that cause a disproportionate number of very small packets or very large packets could skew this somewhat but in general it holds true. Any day that has an event like this *should* appear different to other days and stand out as different, so this is not really a problem. In most cases here however, the byte rate follows exactly the same pattern as the packet rate does. It peaks and troughs at the same points throughout the year, as less packets means less bytes unless new uses to which HTTP is put changes the nature of the traffic changes drastically. The Hurst parameter and other statistics all follow the same trends as those for packet rate as well.

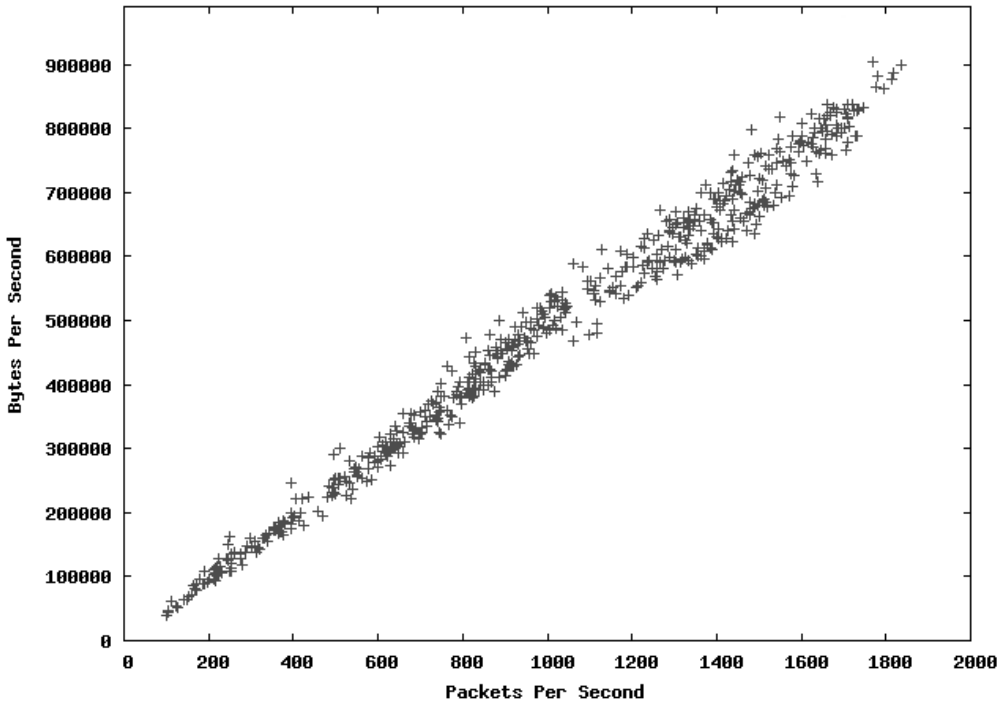


Figure 3.3: Daily Mean HTTP Packets Per Second vs Daily Mean HTTP Bytes Per Second

3.3.3 Flows Per Second

A flow is uniquely identified here by the 4-tuple of source and destination IP addresses, and source and destination ports. Because the traces are filtered to contain only HTTP packets, the protocol will always be TCP and may be ignored, and one of the ports will always be port 80. Any flow for which a packet is observed in a measurement period is said to be active during that time. The number of flows active in any one second very closely follows the same pattern with respect to time of year as do bytes and packets, and so the individual graphs have been omitted. Figure 3.4 compares the number of active flows with the number of packets observed in the same period. There is a very definite positive relationship between the two as expected. The mean packets per flow is 5.79 with a standard deviation of 0.55 packets. This is nearly enough packets to perform

full connection setup and tear down as well as make an HTTP request and receive a response, but that is ignoring bulk transfers which are likely to transfer many times that number of packets. The low average number of packets is likely explained by single packet flows sent to hosts that were not responding, or as part of hostile scans looking for webservers.

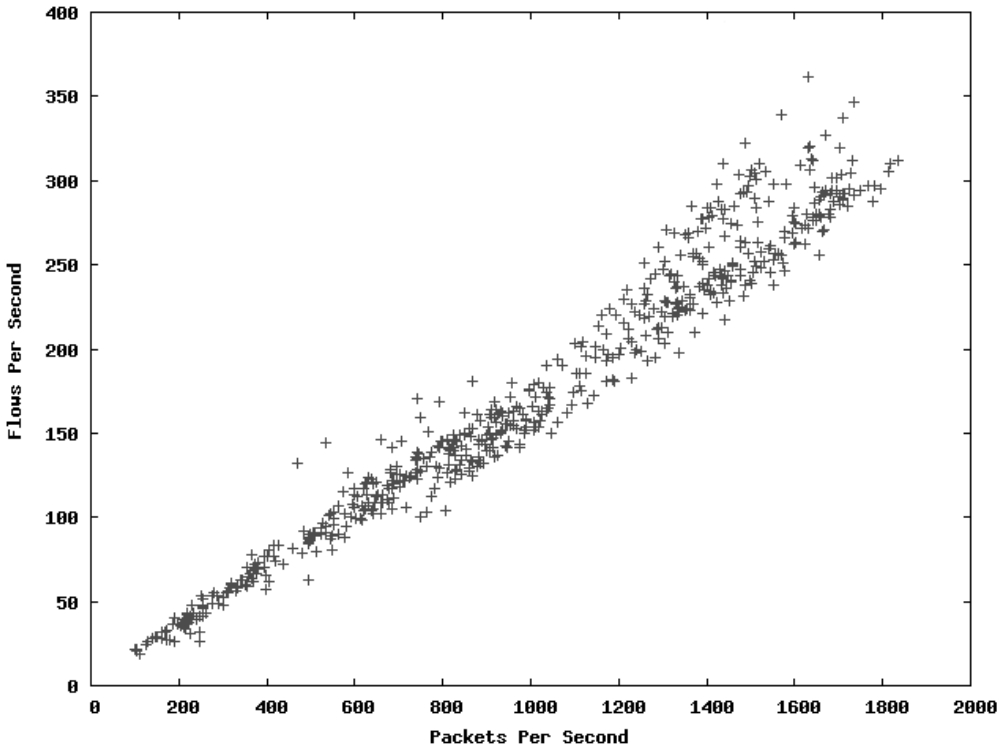


Figure 3.4: Daily Mean HTTP Packets Per Second vs Daily Mean HTTP Flows Per Second

3.3.4 Packet Interarrival Time

The time between one packet and the next one to arrive is known as the interarrival time. The data here is generated by averaging over each second the interarrival times of all packets that arrive during that time. This should be related to the packet rates, if more packets are arriving then the gap between them must be smaller to accommodate

them in the same time frame. Figure 3.5 agrees with this and shows the periods of large interarrival time corresponding with the dips in packet rate shown in Figure 3.1. Note that these graphs use a logarithmic scale for the y-axis due to the broad range of values observed. Breaking this up into weekdays and weekends as well shows similar traits – the weekends have consistently higher interarrival times, apart from where weekdays are holidays.

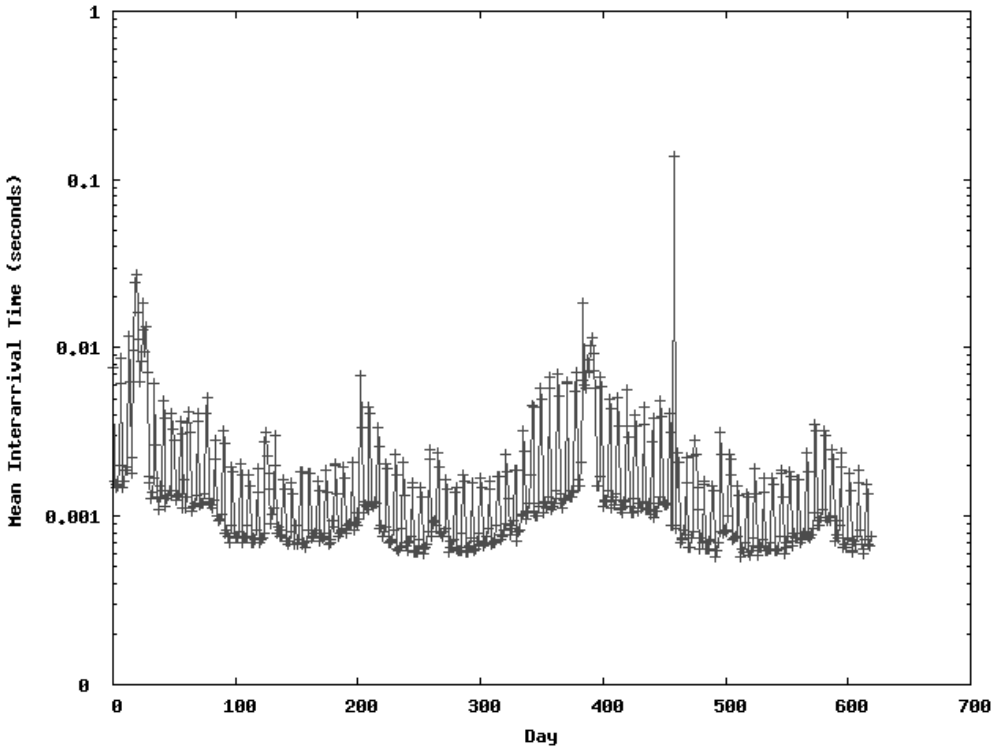
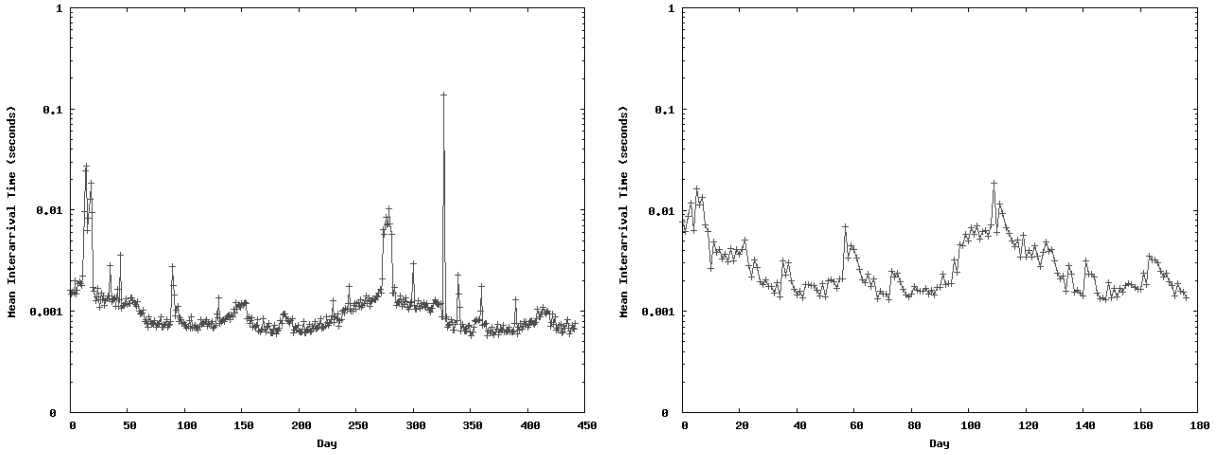


Figure 3.5: Daily Mean HTTP Packet Interarrival Time

The majority of the values that are higher than the baseline are during the weekend and are shown in Figure 3.6b. The peaks during weekdays shown in Figure 3.6a are times when the University is closed such as the Christmas and New Years holidays. The extreme peak about weekday 325 is known to have been caused by a network outage lasting most of the day. No packets were able to be received or sent during this time, but many users and applications continued to make attempts.



(a) (b)

Figure 3.6: Daily Mean HTTP Packet Interarrival Time during (a) Weekdays, and (b) Weekends

Chapter 4

Finding Similar Days

The first real analysis of the dataset was a direct day to day comparison intended to explore how much variation there was in core statistics. Standard methods of visualising comparisons were undertaken and are presented here – differencing the data to measure variability between adjacent days, and quantile-quantile plots comparing the distributions.

4.1 Differencing Data

A simple method of discovering how much difference there is between consecutive days is through differencing. Taking the first difference shows the amount of change that occurred from one day to the next, while taking the second difference shows how much variability there is in these changes.

Figure 4.1 shows the results of differencing the mean packets per second for each day. The thick central band is made up almost entirely of transitions from a weekday to another weekday or a Saturday to a Sunday, while the upper and lower bands are transitions from a Friday to a Saturday, or from a Sunday to a Monday. The difference day to day is very consistent, and in most cases is not large. The other statistics follow

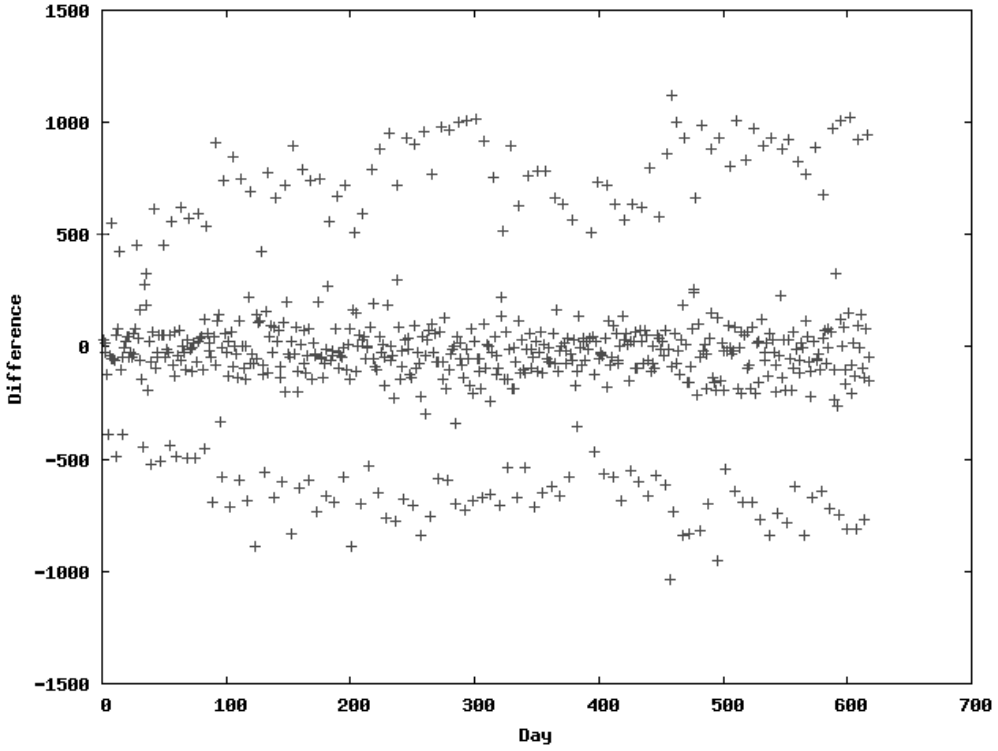


Figure 4.1: Differenced Mean HTTP Packets Per Second

the same pattern again – very little change from one day to another of a similar type, large changes between types.

4.2 Quantile-Quantile Plots

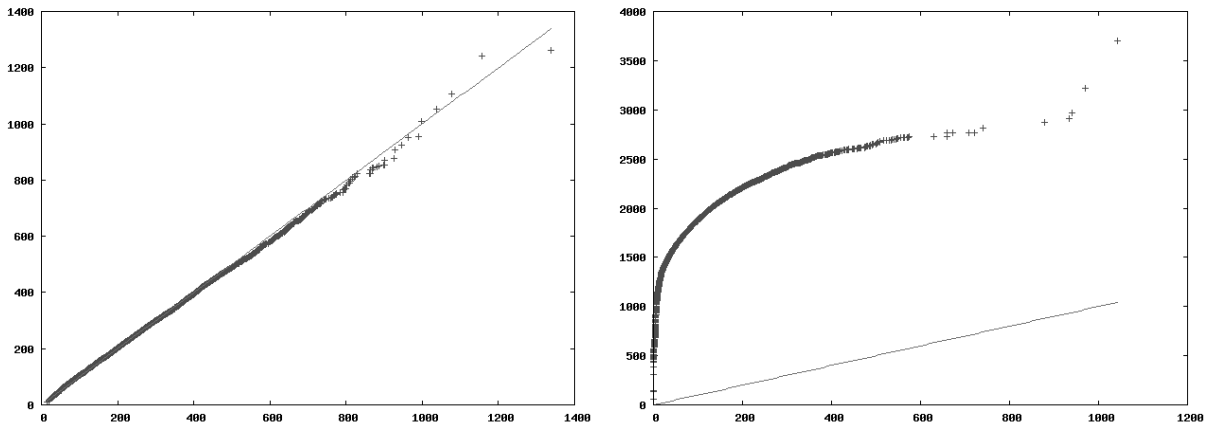
Quantile-quantile plots are made by plotting the cumulative density functions of two sets of data, with one as the X axis and the other as the Y axis. If the distributions of the two are very similar (or very different) then it is quite obvious from a visual inspection of the graph - a line of points that closely follow a reference line that intercepts the Y-axis at the origin and has a slope of 1 indicates a good match. A different Y-intercept but the same slope is indicative of a difference in location between two otherwise similar distributions. To try to compare the data to find how similar days could be, plots were

made of the mean values for the simple statistics for every day against every other day.

620 days worth of traces each compared against every other day gives more than 190,000 unique combinations and so required a method of determining which graphs were of possible interest. Using the property of quantile-quantile plots that results in a straight line following the reference line if there is a good match between the distributions led to the creation of two heuristic scoring function based on least-squares regression. Calculating the square root of the sum of the squared residual values after performing least-squares regression on each pair of days gives the absolute difference between the fitted and observed values – this is how close to a straight line (of any slope) the plot is. Doing the same but using the reference line as the line of best fit and taking residual differences between the observed values and the reference line shows how close the two are. For both tests, higher values should represent days that are spectacularly different, values closer to zero should represent days that are very similar. With this ranking system in place, it was possible to sort all the values to find the pairs that were likely to be the most similar and different, and to get a broad overview of just how much difference there was likely to be between any two days.

Figure 4.2 shows two examples of how the quantile-quantile plots can look based on their score for the heuristic based on their distance from the reference line. Figure 4.2 (a) compares Sunday 22 February 2004 with Sunday 21 November 2004 and shows that the two Sundays outside of term-time have similar traffic characteristics. (b) compares Thursday 1 January 2004 with Tuesday 12 April 2005 and gets very different results - one of the days with the least traffic volume (New Years Day) is very different to a weekday in the middle of term-time.

Figure 4.3 shows the highest and lowest scoring days for the heuristic based off the residuals from comparing the observed data with the line of best fit (drawn). Section (a) compares Tuesday 30 December 2003 with Friday 2 January 2004 and (b) compares Saturday 13 December 2003 with Monday 6 September 2004. The two very similar



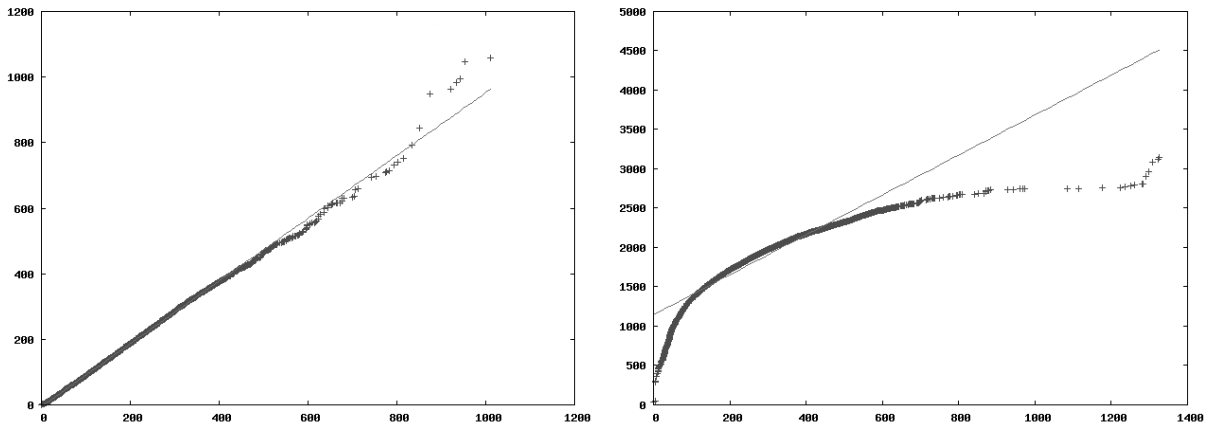
(a) (b)

Figure 4.2: Quantile-Quantile Plots of packet rates for (a) two days with similar distributions, (b) two days with different distributions

days are both during the statutory holidays that surround the new year, and the slope of the line of best fit is actually very close to that of the reference line.

While this method is useful for bringing some order to the large amount of data here, it has a number of shortcomings. Having a low difference when compared to the reference line gives no indication as to how consistently similar the two days involved are. The quantile-quantile plot could oscillate around the reference line and so spend a lot of time near it that cancels out most of the time spent further away, resulting in a low score, but the distributions of the days would not be similar. Two days that have very similar distributions and follow the reference line closely apart from a few observations that deviate greatly will also have a low score (and perhaps deservedly so), but some sort of event may have occurred to cause this and it should stand out as being different. A quantile-quantile plot that follows the slope of the reference line perfectly but is offset by some amount will have a large score as every observation is a long way from the reference line, when in fact the two distributions are similar and differ only in location.

Comparing the residual value of the quantile-quantile plot to a line of best fit found



(a) (b)

Figure 4.3: Quantile-Quantile Plots of packet rates for (a) two days with low scores for the linear regression heuristic, (b) two days with large scores for the same heuristic

through linear regression will determine how straight it is, but gives no information about the angle present. It is possible for a low scoring quantile-quantile plot to have a slope that does not even approximate that of the ideal reference line, though a high scoring one shows that there is a definite difference between the distributions. Even when all these are combined to give a measure of how similar two days are, the values do not have any real meaning - there is an implied ordering for each metric (a lower score is closer to a straight line and therefore more similar) that allows statements to be made about comparative similarity, but at what point in the scale are two days no longer considered to be similar? Manual observation could perhaps offer a cut-off point, but something further is required for this to be automated.

If the slope is taken into account as a fully important part of the heuristic then it makes the results a lot more clear cut, but the problem of knowing at what point to draw the line between being similar and different is still unsolved.

4.3 Changes Over Time

Also worth investigating is the matter of growth and change in the way the network is used over time. If the information is going to be used as part of a simulation framework, then it is important that it is still relevant after it has been measured and not made obsolete. If there is a general trend towards increasing traffic amongst days of a certain type, then if this is known it can be factored into future simulations.

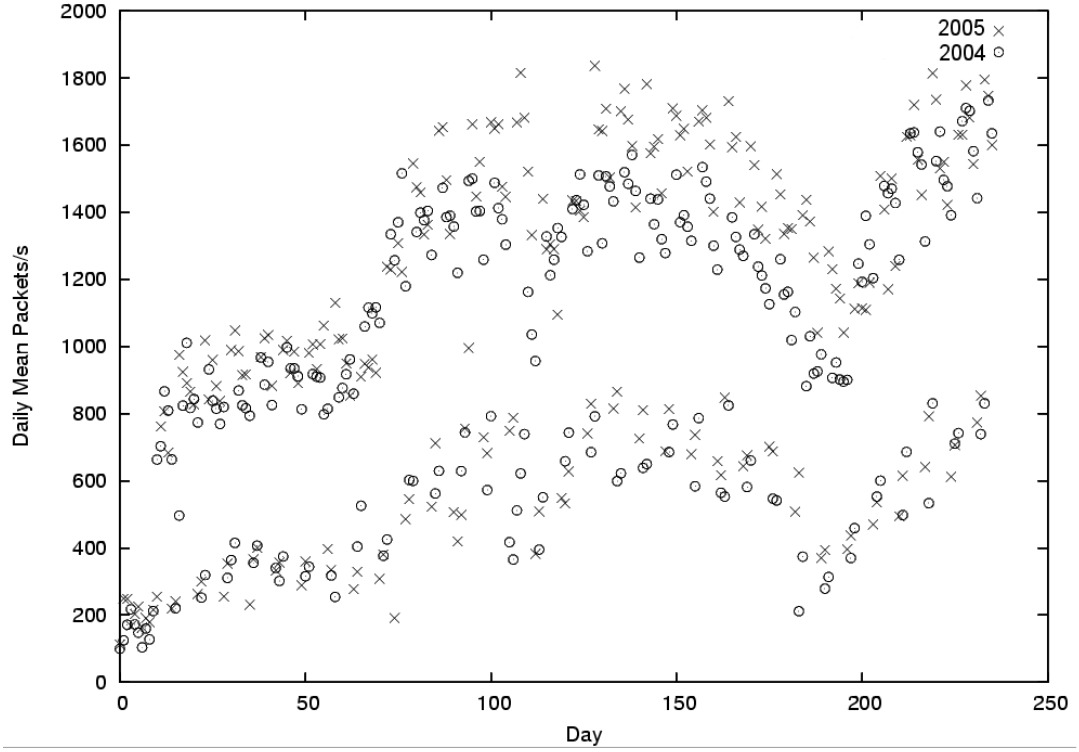
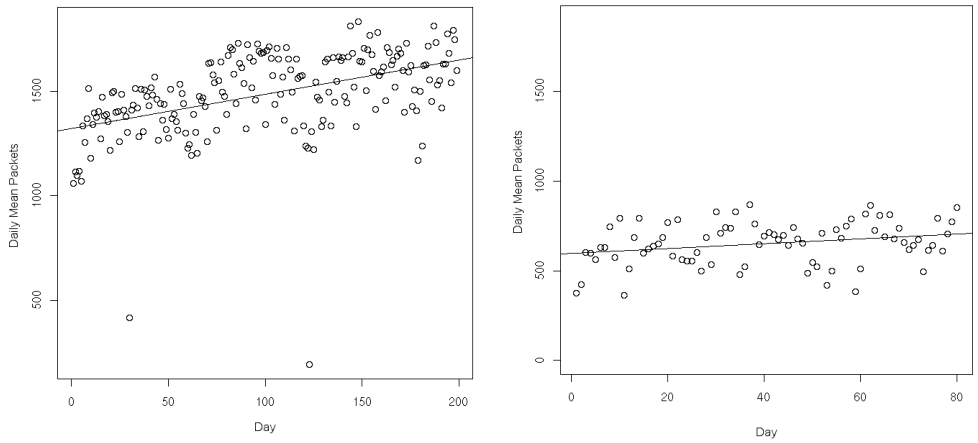


Figure 4.4: One to one comparison of packet rate of days in 2004 with those in 2005

Figure 4.4 overlays the mean packets per second for each day of the year for which data is present for both 2004 and 2005. Using New Years Day as a starting point, each day is compared with the one 365 days after it. Not every day is directly comparable as holidays fall at different times and University term dates change slightly, but it is still useful to see. Remembering from Chapter 3.3.1 that the upper band is made up mostly

of weekdays and the lower band mostly weekends, it can be seen that the weekdays in 2005 are consistently higher by 50 to 200 packets per second. Weekends are very similar in both years, with only slight variation on the whole. Only one third of the pairs recorded a lower value in 2005, and of them the majority are only a small change. The differences appear bigger during busy times but as a percentage they are very similar throughout the year.

An alternative approach using least squares regression also agrees with what is seen in Figure 4.4. Over the whole dataset, the regression line has a slight positive slope showing a general increase in the quantity of traffic traversing the University link, though with the seasonal variation and the data beginning at a time of year with low traffic and ending at a time with high traffic this is not necessarily to be trusted. Splitting it into its composite parts tells the same story however – traffic on weekdays during semester increased by nearly 25% from start to finish, and weekends during semester by 18% (Figure 4.5a and Figure 4.5b).



(a) (b)

Figure 4.5: Linear regression lines drawn with (a) mean packet rate for weekdays during termtime, (b) mean packet rate for weekends during termtime

4.4 Comparison Using Empirical Distributions

The large amount of data available makes it possible to use the distributions of data itself to determine similarity. Any two days that share similar statistics are more likely to be the same than two that are at opposite ends of the distribution. A simple program to compare the locality of two days within the whole range of days was written and is discussed here as a method used to directly compare them.

4.4.1 How It Works

Each day has a number of statistics associated with it and each of these statistics has its own distribution of values. Some of these are distributed normally, some exponentially, some uniformly, each day contributing a single point. In order to compare the values of two statistics belonging to different days, both points are plotted as part of the distribution to which they belong, and the relative distance between them in relation to other values is measured. In the example shown in Figure 4.6, the statistic examined is normally distributed, and a comparison is being made between points *A* and *B*. The area of interest between the two points is shown shaded in the diagram and is the part of the graph that falls between the curve, the x-axis and the two vertical lines at *A* and *B*.

In the actual cases, the distribution curve for each statistic is built directly from the observed data, and is unlikely to be such a smooth curve as in Figure 4.6. This does not prevent the use of integration to calculate the area under the curve between two points. Finding this area as a percentage of the total area shows how many other observations separate these two – few observations and they are likely more similar, many observations and the two days are likely more different. Because of this it works independent of the actual difference in values of the two observations.

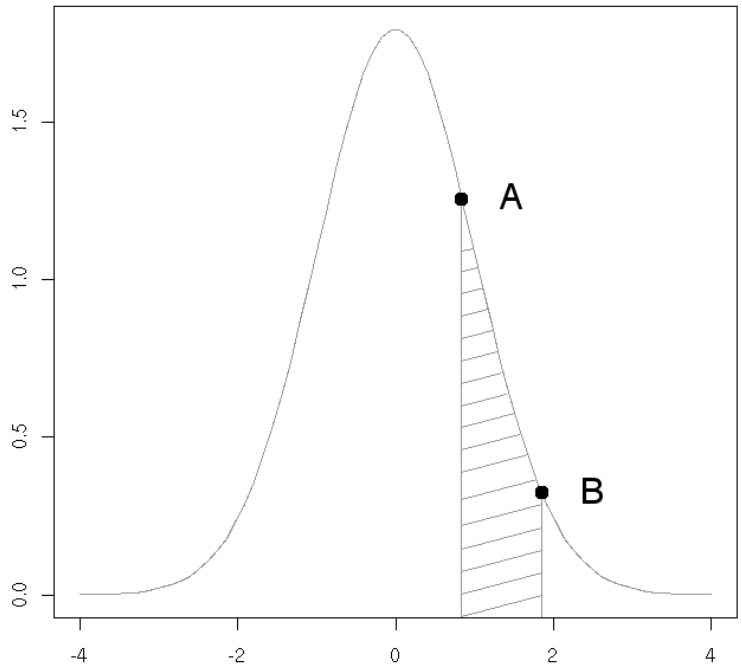


Figure 4.6: Example of a comparison between two points

4.4.2 Disadvantages

This method suffers from some of the same problems as others do - determining how close two days should be before they can be considered the same. If there are a lot of observations within a small area, then small changes in value are actually a large change in the placement of the observation within the distribution. It is not obvious if this is good or bad behaviour - these observations could all be from days that have similar values because they *are* similar and should be counted as such, or because the amount of variability is so low then the small changes are quite significant and the days are different even though there is not a lot of difference in their values.

Some of these problems are investigated further in Section 5.4.2 by combining this method with the results of using machine learning to cluster the data. The difference between all the members of each cluster can be used to see how much variability should

be allowed between any two days before they are considered different.

Chapter 5

Grouping Data Using Machine Learning

With the large amount of data it was important that only the more useful attributes were considered in order to speed execution time and to make human understanding of the attributes easier. To this end, the machine learning tool Weka[29] was used for its clustering and selection algorithms. Also of use is the flexible visualisation functionality that makes it simple to compare the results of clustering with the individual attributes used.

What this aimed to do was divide the dataset into a number of groupings, with each grouping following a separate pattern. Each of these patterns (*clusters* in machine learning terminology) would describe days that were quite similar to each other and shared common characteristics. The values for the defining characteristics of each group can then be used to describe the cluster for further use in such tasks as simulation. The attributes used to decide on cluster membership are discussed in Chapter 3 and of these the most important are investigated below in Section 5.2.

5.1 Clustering the Data

Weka requires the data to be in a particular format known as the Attribute-Relation File Format (ARFF). It is a simple text based format that describes the attributes that are being considered, the type of data they contain, and a list of observations. Figure 5.1 shows the beginning of an ARFF file that was used to cluster data based on various aspects of the measured packet rate. All of the attributes that were independent of any other day were given as possibly important, as well as giving the day of the week, and the part of the university calendar that the data was captured during (mostly based around the number of students present and what they were meant to be doing, for example attending classes or sitting exams). Statistics that had been investigated but relied on a comparison with another day were initially used but very soon dropped as they were considered to be misrepresenting the data when there was no proper method for determining which day(s) they should be compared with.

The Weka software contains an implementation of the EM (Expectation Maximisation) clustering algorithm that is commonly used to put observations into groups that have the maximum amount of similarity within and the maximum amount of dissimilarity without.

The EM algorithm was given freedom to choose the number of clusters that were appropriate to the dataset given. Observations of packets per second, bytes per second, active flows per second, mean interarrival times, and a combination of all of them were used, and in most cases between 6 and 13 clusters were made. However, between each run the membership of many clusters was consistent, with most clusters corresponding to a particular part of the University calendar. This is positive in that it agrees with the earlier observations made of the raw data that each attribute of the traffic measured is strongly related to one another. The rest of the clusters formed had small memberships (sometimes only one or two observations in size) and appeared to have been formed because one aspect or another of the traffic was not quite similar enough to gain entry to

```
@relation packets-indep

@attribute mean real
@attribute median real
@attribute sd real
@attribute diff_mean real
@attribute diff_median real
@attribute diff_sd real
@attribute alpha real
@attribute hurst real
@attribute kpss_pvalue real
@attribute diff_kpss_pvalue real
@attribute day {Mon,Tue,Wed,Thu,Fri,Sat,Sun}
@attribute type {NoStudents,Closed,Stat,Summer,SummerExams,Enrollment,A,B,Recess,Study,Exams}

@data
214.7935 181 145.2776 0.01000093 0 101.4248 0.5456751 0.8535248 0.01 0.1 Sun NoStudents
715.1884 693 247.2832 -0.01518659 0 213.6805 0.4637392 0.746715 0.01 0.1 Mon NoStudents
750.0874 729 237.4157 0.01083434 -1 206.6448 0.4116410 0.8059564 0.01 0.1 Tue NoStudents
723.7181 698 248.3084 -0.01472359 -2 203.0334 0.3860067 0.8548072 0.01 0.1 Wed NoStudents
743.4346 727 245.3017 0.01277896 1 205.9213 0.393638 0.8294088 0.01 0.1 Thu NoStudents
...
```

Figure 5.1: Example ARFF File for Packet Rate Data

one of the major clusters. These are the days where everything was normal apart from there being an unusually light load (statutory holidays), or where something went very wrong with the network itself (such as the complete loss of the uplink made obvious by the peak in Figure 3.5).

5.1.1 Example - Packets Per Second

The initial run used only packet rate data to test the clustering tools and to make sure that the output made sense given what is known about the link measured. Figure 5.2 shows the output from Weka after running EM over the data and visualising the clusters in relation to the day of the week on which the data was collected. Almost every cluster is clearly defined as representing a type of weekday or weekend, apart from a number of very small clusters that cover both. Clusters 3, 9 and 12 cover most of the weekdays, with the majority being cluster 9. Cross-referencing this with Figure 5.3 shows they each belong to mostly separate parts of the year - cluster 9 covering times when there are students around, with 3 and 12 covering quieter parts of the year. Cluster 2 looks to

correspond with busier weekends and 7 to those weekends where there are less students.

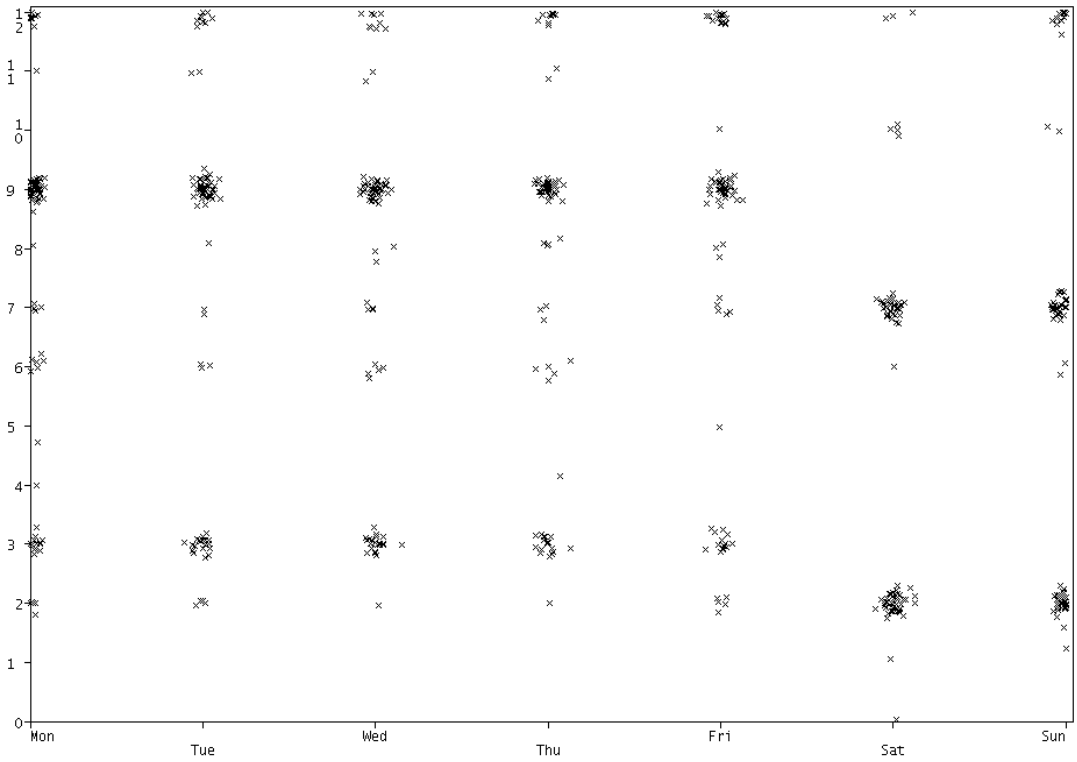


Figure 5.2: Clustered Mean Daily Packets per Second vs Day of the Week

The smaller clusters such as 0, 1, 4, 5, 6, 8, 10 and 11 tend to appear in only one part of the University year and are unusual. Many of the clusters have a membership of less than ten days out of the 620. Chapter 5.2 accounts for most of these clusters as the result of over-fitting the data after investigating which attributes were used during the selection process.

Treating each aspect of the measured traffic separately gave very similar results that are not shown here as they would add nothing new. They are also made redundant by later developments discussed in Section 5.3. The final ARFF file was a combination of all the data involving packet, byte, flow counts, and interarrival times. As expected due to the similarity of the four components making up this combined dataset, the results of putting them all together are very similar to the individual results. The major clusters

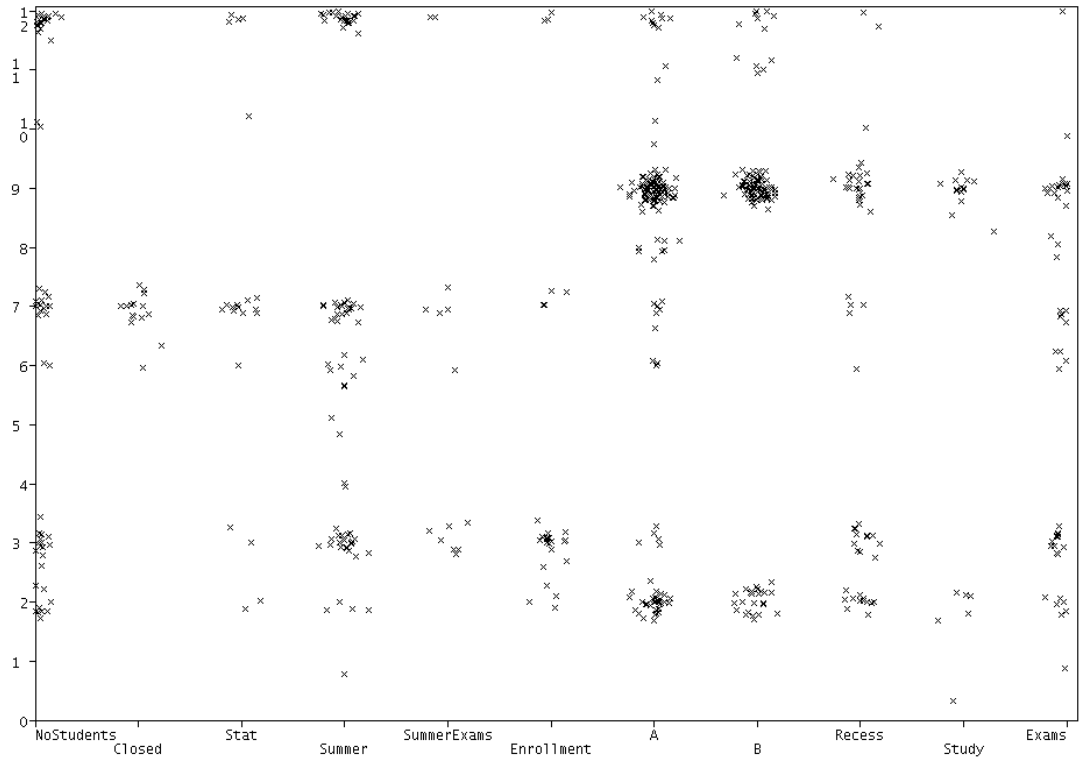


Figure 5.3: Clustered Mean Daily Packets per Second vs University Calendar

are still built around the University term dates and the minor clusters still exist to fill niche situations. Everything discussed with respect to the clustering based solely on packet rates is still applicable to these clusters.

5.2 Attribute Selection

Because the clustering algorithm gives no information as to what attributes were used to make decisions, a supervised learner needs to be run over the output of the clusterer. Assuming that each day belongs to the cluster that was shown to be the most likely candidate, the cluster membership can be appended to the record in the ARFF file and used as a class for a supervised learner. Using a decision tree instead of a method such as a neural network has the advantage that the output is in a human readable format and

it makes it explicit which attributes and values were used to make a decision. Tree and rule learners must be able to express their results in a way that shows what attributes were considered useful, while neural nets and other more complex systems do not always show this clearly. Weka also contains a module specifically for determining which attributes are the most effective at predicting class, and this was used to check against the decision trees.

5.2.1 Decision Trees As Attribute Indicators

The initial decision tree created by the J48[20] algorithm using packet rate data is shown in Figure 5.4. The highest nodes in the tree are those that branch on the median, the mean and the p-value of the KPSS test for stationarity (shown as *kpss_pvalue* in the figure) which agrees very strongly with the results of the dedicated attribute selection. Of interest however are the nodes branching on the result of the KPSS test - they are comparing p-values that are so similar their is really little difference between them. Most of the values used in the pivot points are significant at the 0.05 level and it does not really make sense to examine the data in too much more detail. They could be treated as different degrees of significant, but the main reason behind performing the test was to check the data for stationarity. It is only really important as to whether the data is likely to be stationary or not, and a low p-value allows us to reject the null hypothesis of stationarity. So, the data was reformatted to convert the p-value into a nominal with two possible values, *stationary* and *non-stationary*, with any score over 0.05 being considered a failure to reject the null hypothesis and marked as *stationary*. This change caused the KPSS test to drop out of the decision tree and attribute selection altogether and left the mean, median and standard deviation as the most important attributes.

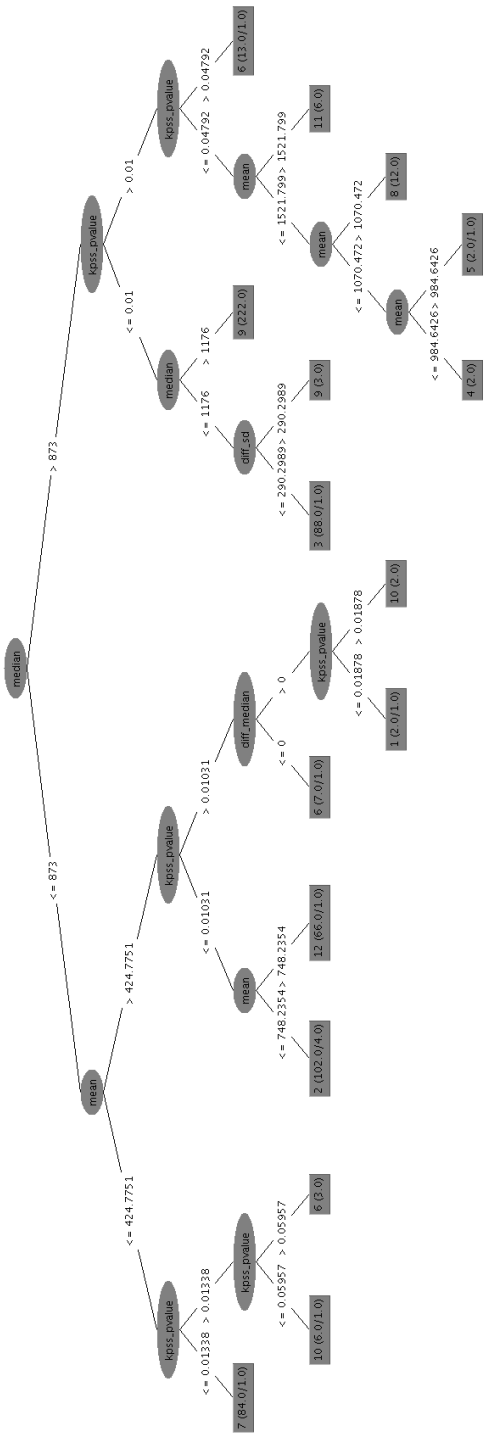


Figure 5.4: J48 Decision Tree Based on Packet Clusters

5.2.2 Weka Attribute Selection

Weka includes an attribute selection algorithm that looks for attributes that correlate strongly with the class and have low correlation between themselves. Those attributes that are the strongest predictors of class are chosen. Running this over the same data gave very similar results to what was indicated by the decision trees. Over the combined data set the following attributes were determined to be the best predictors of class -

- Packets per second
  - Mean
  - Median
  - Standard deviation
  - Standard deviation of differenced data
  - Alpha value
- Bytes per second
  - Mean
  - Median
  - Standard deviation
  - Standard deviation of differenced data
- Flows per second
  - Mean
  - Median of the differenced data
  - Standard deviation of the differenced data
  - Alpha value

- Packet interarrival time
  - Mean
  - Median
  - Standard deviation
  - Hurst parameter
- Position in the University calendar

Though these are the attributes that are the strongest predictors of class according to Weka, not all of these are actually as distinguishing as they might appear. For example, though the Hurst parameter has been indicated to be important (both here and in other literature) when examining their values across all days and all clusters there is no distinguishable pattern. No cluster stands out as being predisposed towards having Hurst parameters of a certain value, though its presence in decision trees (after the changes to the way the KPSS test was treated) and being chosen by the attribute selection algorithms means it must have some predictive value. Chapter 5.4.2 explores this particular case in more detail.

These attributes listed are not treated equally either. Though the attribute selection algorithm does not assign a value to show their importance, combining this information with the decision tree and examining the results of clustering can give some clues. In the new decision tree after changing the way the results of the KPSS test are treated, packet based statistics are represented in larger numbers than those based on other aspects of the traffic. At the root node, the mean number of packets per second is used to essentially split days based on whether or not they are weekdays or weekends. Other attributes then fine tune the selection.

5.3 Re-clustering The Data

The new set of clusters that came about after changing the KPSS test to produce a nominal value is shown in Figure 5.5. In general it is not too dissimilar to the first in that there are clear distinctions between dates during termtime and outside of it, and weekdays and weekends. What is different however is that there is a distinct change in the behaviour of termtime weekdays at approximately day number 255 – halfway through the second semester in 2004 – and then again at the start of the A semester in 2005. This is observable in Figure 5.5 in that cluster 3 is solely present during the B semester that is this change period, and cluster 5 following it is lacking in B semester dates, as the observations end part way through it in 2005. Cluster 3 is a transition period where the mean volume of termtime weekday traffic increases from the stable and consistent level it had been up to that point, to the larger and more variable level it exhibits in 2005.

The same change is observable in the individual packet and byte count data, and also happens to a smaller degree for the flow data. Interarrival times of packets does not show any such obvious change. Suggested reasons for this phenomenon include an increase in the amount of bandwidth available to the University or an improvement in the way the service is supplied by the ISP as these would both make it possible for the volume of traffic to go up. It seems almost certain that the explanation for this is technical – the consistency of the maximum exhibited in clusters 2 and 8 has to be indicative of an artificial limit, and during the time of cluster 3 this is increased slightly, but still limited. Decreasing traffic costs to users could increase the amount of usage, as could availability of networking somewhere it was previously unavailable, but neither of these possibilities fits well with the extremely consistent maximum rates observed here.

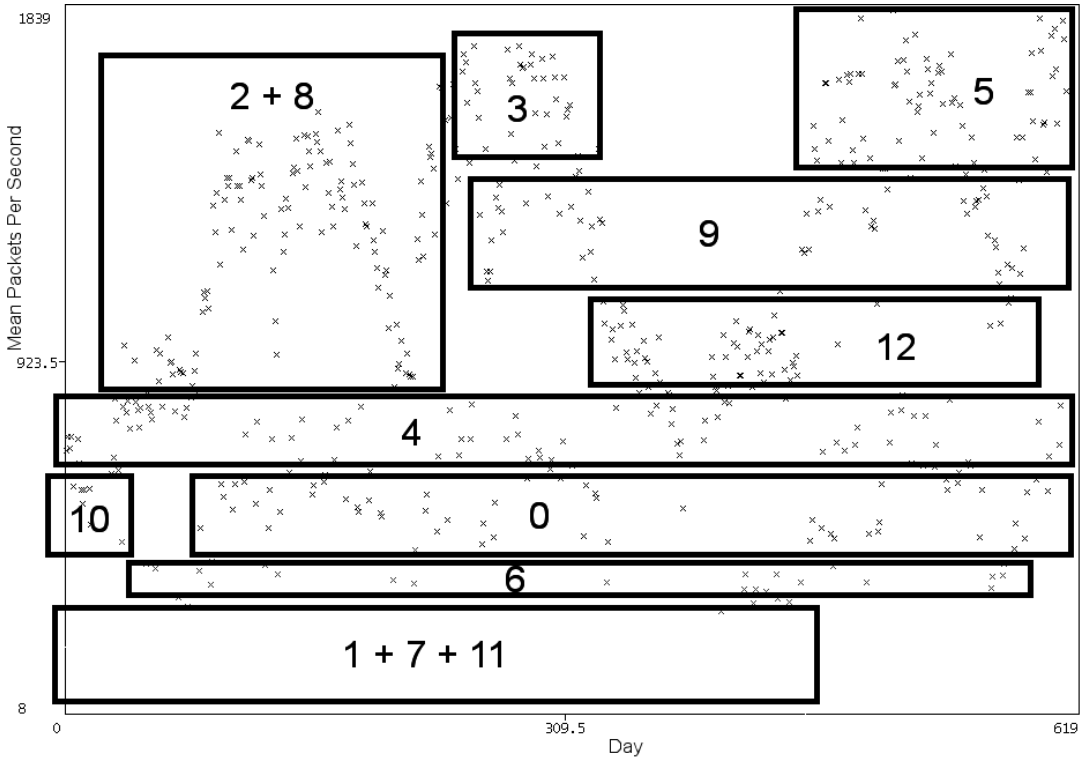


Figure 5.5: Daily Mean Packets Per Second with Cluster Memberships From Combined Data

5.4 Cluster Analysis

Now that there is some sort of indicator as to which days are similar to other days, comparisons can be made within each cluster. Using the system described in Chapter 4.4, the days that make up each cluster were plotted on the distribution curve representing all the data. Each day in the cluster was compared with each other day in the cluster and the distance between them in the distribution recorded.

Plotting only the days belonging to a single cluster results in a curve approximating normal for each statistic, which is not unexpected. Comparing all the members of the cluster with each other but inside the entire distribution shows the extent that each cluster is spread, and in most cases this is actually quite large, sometimes spanning the entire distribution.

5.4.1 What Are the Clusters?

As described in Chapter 3.3, even looking at just the simple descriptive statistics shows clearly that even though most days are not the same, many of them share similar characteristics. Weekdays are a quite distinct group compared to the weekends, but that is a simplistic division based on only a few aspects of the data. Chapter 5 added more information and used machine learning to create more complex groups, though they were still in line with what was noted earlier.

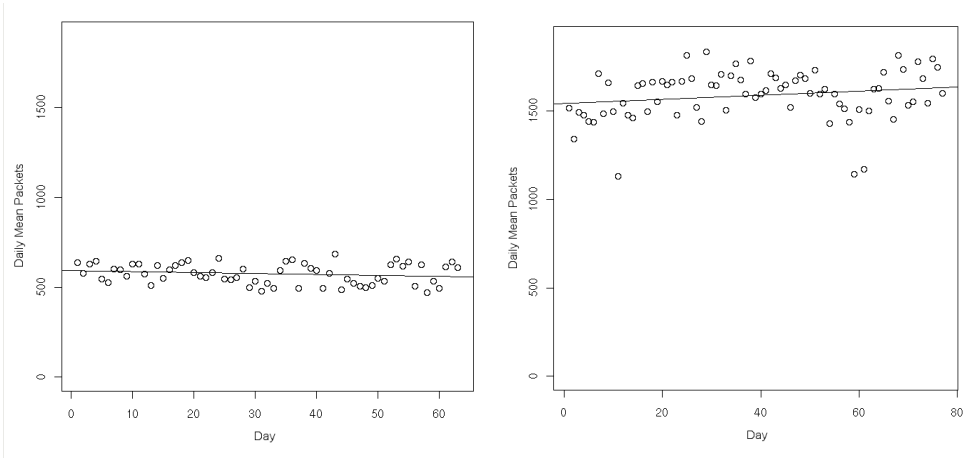
The clusters are very strongly tied to the number of users that are present at the time the measurements were taken. This can be observed in the way that cluster membership follows the University calendar and the working week, and are made up mostly of only one period within it. As was shown in Figure 5.5, when the clusters are overlaid onto a graph of the mean packet rate the groups are clearly defined and only minimally intermixed. Weekdays have higher packet rates than weekends, and days during termtime have higher rates than their counterparts outside of it, and so the days with the highest rates are weekdays during termtime. These days form the basis for approximately a third of the clusters, with variation due to (what are likely to be) changes to the amount of available bandwidth. Weekends outside of termtime have the lowest rates, and unusual days such as statutory holidays fall into the same category. In terms of cluster membership, these are a lot more stable over time and fall into a number of clusters depending on exactly how busy it is at the time.

Changes Over Time

Lower rate clusters involving weekends and dates outside of termtime have much less change over the measurement period. During times when there are no students studying, the number of users (i.e. University staff) is likely to remain relatively constant.

Figure 5.6a shows that the mean HTTP packets per second during termtime weekends (cluster 0) is relatively stable, though decreasing slightly. The number of people

who are in the University during the weekends apparently changes little, and a good portion of this traffic is likely generated externally such as people visiting the University website from off campus. This is different to the result shown for all term time weekends in Figure 4.5. Another set of days at the same time must have higher packet counts and belong to another group – cluster 0 is the norm and most common type of termtime weekend, but particularly busy or light days do exist and belong to other groups.



(a) (b)

Figure 5.6: Linear regression lines drawn with (a) mean packet rate for days in cluster 0, (b) mean packet rate for days in cluster 5

The mostly termtime weekdays represented by cluster 5 are shown in Figure 5.6. They are much more similar to the total set of termtime weekdays in that they are increasing over time, but not by as much. A major reason for the increase shown in the total set is due to the assumed increase in bandwidth available yet cluster 5 consists entirely of days that fall after this point and still shows an upward trend.

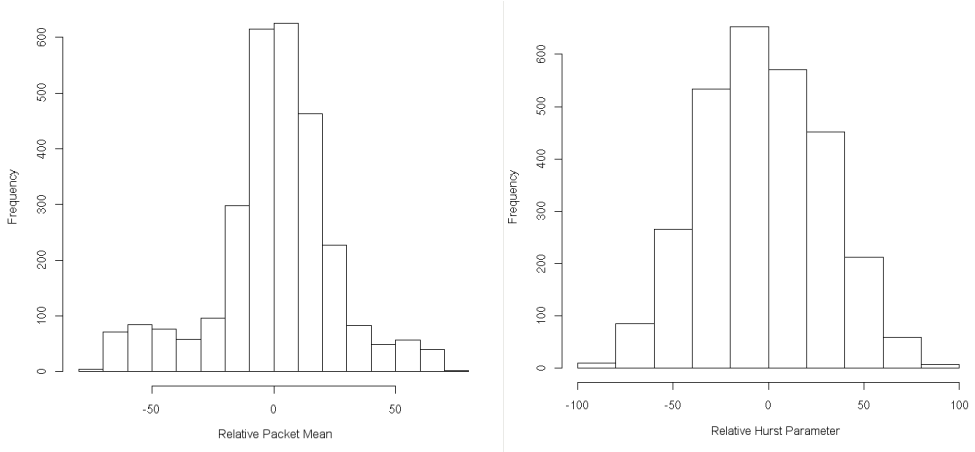
5.4.2 Distribution Of Statistics

Taking the method described in Section 4.4 and applying it to the results of clustering can give an indication as to how similar days are. If two days fall into the same cluster

then they already share some traits, but just from knowing that it is not clear to what degree this extends. Comparing the locations of all the statistics for each day within a cluster with all other days in the same cluster will show just how similar that statistic is. If all the days have a very similar value, then there will be very little spread throughout the distribution. The values for a number of statistics in two of the major clusters are discussed below.

Distribution of Days Within Cluster 5

The placement of these days within the complete distribution is mostly limited to a small portion, though individual days can be found at all points throughout. Figure 5.7a shows the distribution of packet means and is representative of most statistics measured here – the majority of the values are tightly focused around a single point, and deviate only slightly for the most part. These attributes are also the ones that were noted as being most useful for distinguishing between different types of day because all days of the same type have values for these attributes that are grouped very closely together.



(a) (b)

Figure 5.7: Cluster 5 Distributions of (a) Mean Packet Rate, (b) Hurst Parameter

Contrast this with the histogram in Figure 5.7b where the Hurst parameters for days

in cluster 5 are spread all throughout the entire range of values observed. A particular value of the Hurst Parameter is not going to be a good indication of what type of day is being examined because there is no co-locality of values in the same grouping.

Other statistics fall somewhere in the middle. None are in such a strong grouping as the packet mean, but many are close, and none are as widespread as the Hurst parameter. The amount of spread is not large, but nor is it narrow enough that any decision can be made about what sort of day is being observed just from the position of a few values within the distribution.

Distribution of Days Within Cluster 0

Cluster 0 behaves very similar to cluster 5 in all respects, though it is slightly more variable. Figure 5.8 is the distribution of packet means and is the most tightly grouped statistic for this cluster also. There is very little difference between the shapes of the graphs compared with what was just seen in cluster 5, and none of them are really grouped tight enough that they can be trusted enough to base decisions off of.

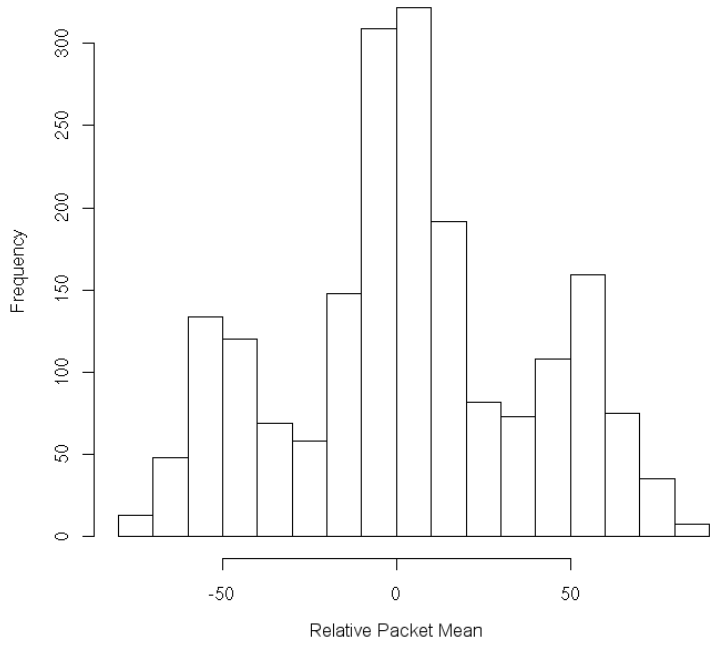


Figure 5.8: Distribution of Packet Means within Cluster 0

Chapter 6

Simulator Evaluation

In order to evaluate the effectiveness of the techniques described here, three different workload generators capable of producing HTTP traffic were tested. All work with the network simulator *ns*[24]. A number of different simulations were run covering some of the different types of days discussed in Section 5.4.1. The techniques above and the manual observation of the resulting data can be used to assess both how accurate the traffic models are, as well as how accurate the classification of similar days is. The simulation setup is detailed here, and the results of these simulations can be found in Chapter 6.3.

6.1 Simulation Setup

Each simulation used the same topology as shown in Figure 6.1, which is based on the commonly used dumbbell. The left hand side represents machines within the University and the right machines on the Internet. Each of the links has a serialisation speed of 100Mb and a delay ranging from less than a millisecond to a few that have in excess of a second. These delays were chosen by observing the round trip times of all hosts sending or receiving HTTP packets during the three hour measurement period for the day 25 May 2004. This day was used here as being representative of a weekday during termtime

that belongs to cluster 8 from the combined dataset, and because previous work had been done to extract a lot of information from it already. Parameters for each simulation were given as the mean values over all days that fell within the desired designation. The specific days mentioned above were only used to determine the simulation topology.

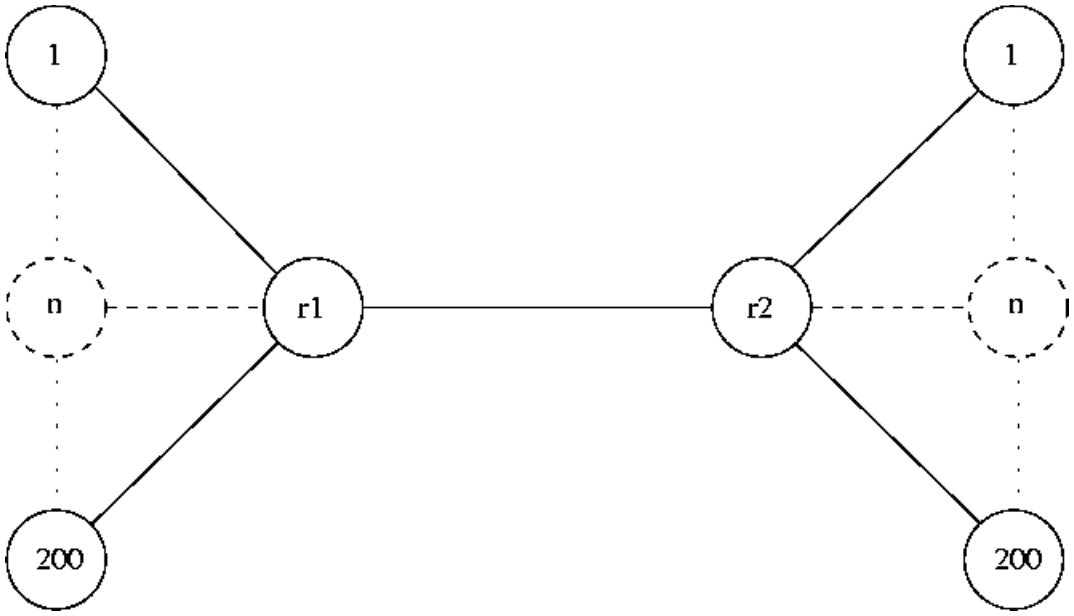


Figure 6.1: Simulation Topology

The round trip time calculations were made by tracking data packets on a per flow basis as they passed the capture point, and waiting for an acknowledgement packet in the opposite direction. Assuming that the time between a host receiving a data packet and sending an acknowledgement in response is negligible then the the time a packet takes to travel from the measurement point to that host can be calculated. Doing this in both directions will give an estimate of the round trip time. However, most network stacks make use of delayed acknowledgements, so only the timing of the most recently sent data packet that is covered by the acknowledgement can be used. If other packets were to be used, then the estimate of the round trip time would be much larger than the actual one, as the acknowledgement may be delayed by up to 500ms[1] (though often much less in practice). This could still cause problems should any acknowledgement

be delayed for the length of the timer, but averaging all the observations will limit the effect of this as more packets around the true RTT are seen.

The average round trip time to each host (over 14600 external and 200 internal hosts) were then sampled with replacement using the number of packets sent to/from each host as the probability of it being chosen. Using replacement meant that the round trip times to nodes in the simulation topology would be of a similar distribution to what was observed on the network. Nodes in simulation all have an equal chance of being selected as an endpoint so the distribution of delays on the links needs to match that of the real data.

6.2 Traffic Generators

The SURGE[2] HTTP workload generator is considered a seminal work in the field of traffic generation. There have been models derived from SURGE for use within *ns* but these have a strong emphasis on the characteristics of the webpages and webservers involved and try to model per user behaviour. It is possible to recreate the original file sizes from the network traces and use such a model, but the main focus here is towards aggregate flows. The first two models investigated generate data based on a description of the traffic directly, though the third is more closely related to the SURGE method – it uses a model of packet sizes and user responses built into a state machine.

The traffic generators take a number of parameters that are used to describe what should be generated. Some of them are quite detailed and require a lot of information in order to tailor the traffic to what is expected, while others are very minimal and make use of built in values. The three traffic generators presented here were chosen to represent different levels of configuration and reliance on knowing about the properties of the traffic to be generated.

	Cluster 5	Cluster 0
Rate of New Connections	49/s	33/s
Duration	10800 seconds	10800 seconds

Figure 6.2: PackMime Parameters

6.2.1 PackMime

The PackMime HTTP traffic generator[5][28] attempts to model traffic as a series of connections between hosts. It does not take a lot of parameters that describe the specific qualities of the traffic to be generated. Much of the detail is built into the model based on the decisions that were made when it was written, and so the only configurable traffic option that it is possible to set is the rate of new connections. Most of the network settings PackMime makes available are overwritten with the standard topology described in Chapter 6.1 and so are not used. Though the configuration is rather limited, this did not appear to be too much of an issue because the number of flows is an important attribute in determining cluster membership, as well as packet and byte rates with are deterministic on the number of flows.

6.2.2 SupFPR

SupFPR[31] is also a traffic generator for *ns*. It creates self similar traffic using fractals, based on a number of user configurable parameters. It was chosen because the parameters were good match with those that had already been identified as important to describing traffic, and it was one of the few that allowed the degree of autocorrelation to be specified explicitly. Although it has not rated highly in the attribute selection tests here, it has been shown that self similarity and a degree of autocorrelation are integral to describing network traffic and so it is important that they are represented here.

Table 6.3 lists the parameters used in the simulation of two clusters. These were generated from the mean values of all days that belong to the given cluster to get settings

	Cluster 5	Cluster 0
Mean Sessions	269	102.2884
Mean Session Duration	62 seconds	33 seconds
Mean Data Rate of Sessions	22.47Kbit/s	21.18Kbit/s
Mean Packet Size	487 bytes	481 bytes
Hurst Parameter	0.836	0.8097542
Duration	10800 seconds	10800 seconds

Figure 6.3: SupFPR Parameters

that

6.2.3 Traffic Generation Using State Machines

The methods described above were also used to evaluate a traffic generator that is currently in development within the WAND Network Research Group at the University of Waikato. Using a saved trace file, it places groups of packets into *objects* – a single part of a transaction between two hosts, such as a webpage or a file. These are grouped according to flow, along with their size and the direction they were travelling in, then machine learning techniques are used to generate possible states for each object. Links between the states are created and merged together iteratively as they are found to be similar until no more states can be merged and the state machine is of a minimum size.

Running this over the trace file containing the three hours from 1pm to 4pm on Tuesday 25 May 2004 as an example of a weekday within termtime produced the state machine graph shown in Figure 6.4. The machine is traversed from the *initial* to the *final* state, with the probability of any particular link being taken indicated beside the link. Again, this day was chosen because it is a relatively busy day containing many HTTP flows to build a model from, and because it has been used for previous experimentation there is a lot of information pre-calculated from it. Unfortunately this day belongs to cluster 8 whereas the other simulations are based on days in cluster 5, but at this stage of development the process of building a model for this simulator is long and complex.

This day had already been built and made available for simulation.

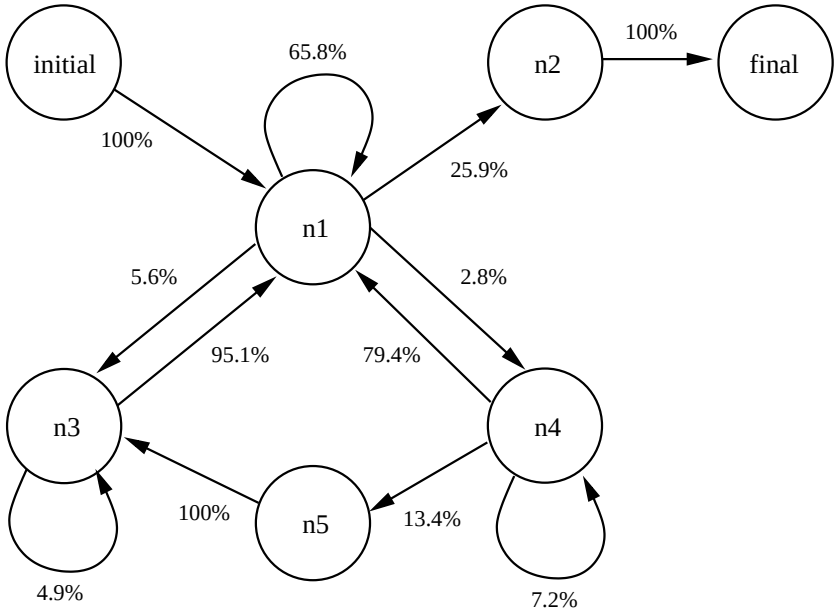


Figure 6.4: Generated HTTP State Machine

Examination of the packet size distributions and the number of connections that traversed each link gives some indication of what parts of an HTTP connection the state machine represents.

A large number of connections are very short and consist of a small request and small response – they pass through the state machine in the quickest way possible. More have a small request and quite a large response (these are the connections that take the path through the machine looping back solely on node *n1*) before finishing with a small packet. The paths involving nodes *n3*, *n4* and *n5* are much rarer and represent more "chatty" uses of HTTP. Some of these involve swapping smaller packets, and others a combination of the two, in different directions. HTTP is used for a wide variety of tasks, and it seems right that the state machine should reflect that with shorter smaller connections and larger longer ones. Other types of HTTP traffic can be constructed through a combination of the two major types. This method of traffic generation is still a work in progress and is yet to undergo more formal validation than what is presented

in Chapter 6.3 where its output in a small number of situations is discussed.

Because it is still in the early stages of development, this traffic generator uses empirically observed timings for the creation of flows. The start time of each flow in the original trace is replicated for the simulation, which could bias the results more accurate than perhaps they should be, however the duration of the flows and the packets transmitted are based off the state machine above. If this state machine is not accurate, then having exactly correct flow start times is not going to be enough to make the simulation as a whole accurate.

6.3 Results

Applying the same statistical analysis used on the real data to the results of simulation gave results that tended not to match as well as they could have. The results of 10 simulated runs were averaged to find the values reported in the tables below, except for the simulation using state machines due to the complexity involved.

6.3.1 PackMime

The lack of configuration options appears to have prevented PackMime from accurately representing the HTTP traffic done by the University in both busy times and quiet times. Though the average number of connections per second can be specified, the duration of flows and the amount of data they transfer are all based off a different set of observations. While perhaps useful for generating generic HTTP traffic, this means that it is not likely to produce traffic that is similar to that observed at any other observation point. The average results after ten simulations using PackMime are presented in Figure 6.5.

Despite most of the raw values being very different to what was expected, and the single configuration option, it appears that a lot of the underlying properties are similar to the University link. The average number of bytes per packet is consistent between

	Cluster 5	Cluster 5 Sim	Cluster 0	Cluster 0 Sim
Mean Packets/sec	1588.507	885.805	575.027	602.2271
Packets/sec SD	327.4611	453.8062	217.373	367.8936
Hurst Parameter	0.84	0.80	0.81	0.81
Mean Bytes/sec	773381.6	437302.8	276852.4	296095.2
Bytes/sec SD	184265.9	248321.1	138450.7	208252.4
Mean Flows/sec	268.9532	130.9401	102.2884	87.48644
Flows/sec SD	53.21577	65.66118	29.48343	49.58348
Mean Interarrival	0.0006833	0.0014988	0.0020294	0.0024332
Interarrival SD	0.0005435	0.0009446	0.0009247	0.0018435

Figure 6.5: PackMime Results

the observed and simulated data sets for both weekdays and weekends, and the Hurst parameter of the generated traffic is very similar to that expected.

Classification of Results Using Weka

The average results from the simulations were converted into an ARFF record and the models that generated from the original data were used to classify them. The simulation of a weekday during termtime in the year 2005 (cluster 5) was classified as belonging to cluster 3, which though different, is still a cluster containing mostly weekdays during termtime. It covers the interesting change-period that happens during B semester in 2004, and ends when the cluster that it was intended to match begins. There is not a lot of difference between the two, except that cluster 5 generally has a larger volume of traffic that is more variable across the measured time period than cluster 3. However, the simulated day is still not a good fit into this cluster and only really fits within it due to the simplicity of the model used, and the fact that it must belong to an existing class. Based mostly on the mean packet and byte rates and with no lower limits, this simulated day is classified rather than being declared as being outside of any traffic grouping.

The raw statistics of the simulation of a weekend during termtime are a lot closer to the observed days, but they still are not close enough in order to place the day into the

right group. Though the parameters were based on days from cluster 0, the averaged results of simulation were also classified into cluster 3 using the same model used above, for the same reasons.

Classification Using Other Methods

As an example of how the simulation compares with other days, Figure 6.6 shows a quantile-quantile plot of the mean packet rates from the weekday simulation with the mean packet rates for Tuesday 5 May 2005. This day was the base for the topology used and is in a similar time within the University year. It is not a good match with the results of simulation however. The majority of the one second measurements made in simulation fall well below the values from the real day, apart from the highest values seen which are in excess of those actually observed.

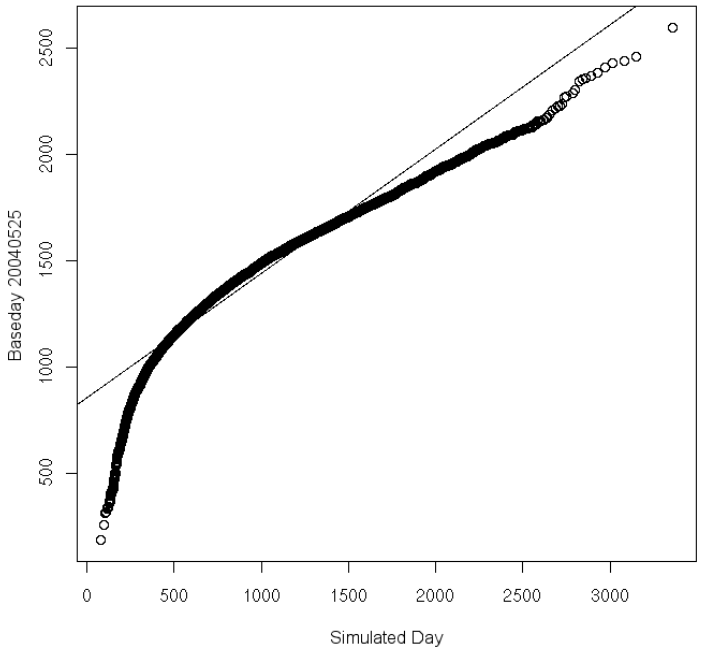


Figure 6.6: Quantile-Quantile Plot Between Mean Packet Rates of PackMime Simulated Cluster 5 Day and 20040525

	Cluster 5	Cluster 5 Sim	Cluster 0	Cluster 0 Sim
Mean Packets/sec	1588.507	1540.362	575.027	554.630
Packets/sec SD	327.4611	461.3585	217.373	250.541
Hurst Parameter	0.84	$<$0.5	0.81	$<$0.5
Mean Bytes/sec	773381.6	436759.7	276852.4	155637.5
Bytes/sec SD	184265.9	130949.4	138450.7	70374.54
Mean Flows/sec	268.9532	26.44581	102.2884	12.660
Flows/sec SD	53.21577	5.113293	29.48343	3.471119
Mean Interarrival	0.0006833	0.0007118	0.0020294	0.0021829
Interarrival SD	0.0005435	0.0003506	0.0009247	0.0015625

Figure 6.7: SupFPR Results

6.3.2 SupFPR

The traffic generated by SupFPR looks superficially similar to the data that it was based on. The number of packets per second in simulation is comparable to that observed in each cluster, but the number of bytes and flows observed is a lot lower than expected. Interarrival times are similar to what they should be, understandable because of the similar packet rate.

The higher level properties of the traffic are completely different however, and do not match the target traffic. Neither do they match the parameters originally given to the simulator, for example in no case was the Hurst parameter for packet arrivals remotely close to that which was specified. The analysis of the SupFPR generator in [31] includes discussion of the Hurst parameter and its effect on the autocorrelation function over a short simulation lasting approximately 65 seconds. In their analysis, the ACF decays much more rapidly than that of the observed data on the Waikato link, but still exhibits quite strong long range dependence (their figure is reproduced here as Figure 6.8a). The simulations performed here using SupFPR however have almost no long range dependence (Figure 6.8b) and the ACF decays to zero much more rapidly than it has been shown to, and much more rapidly than it should if it were accurately representing the desired traffic (Figure 6.8c). Based on their analysis it was expected that the value

given for the Hurst parameter would be approximated in the output, but this is obviously not the case here.

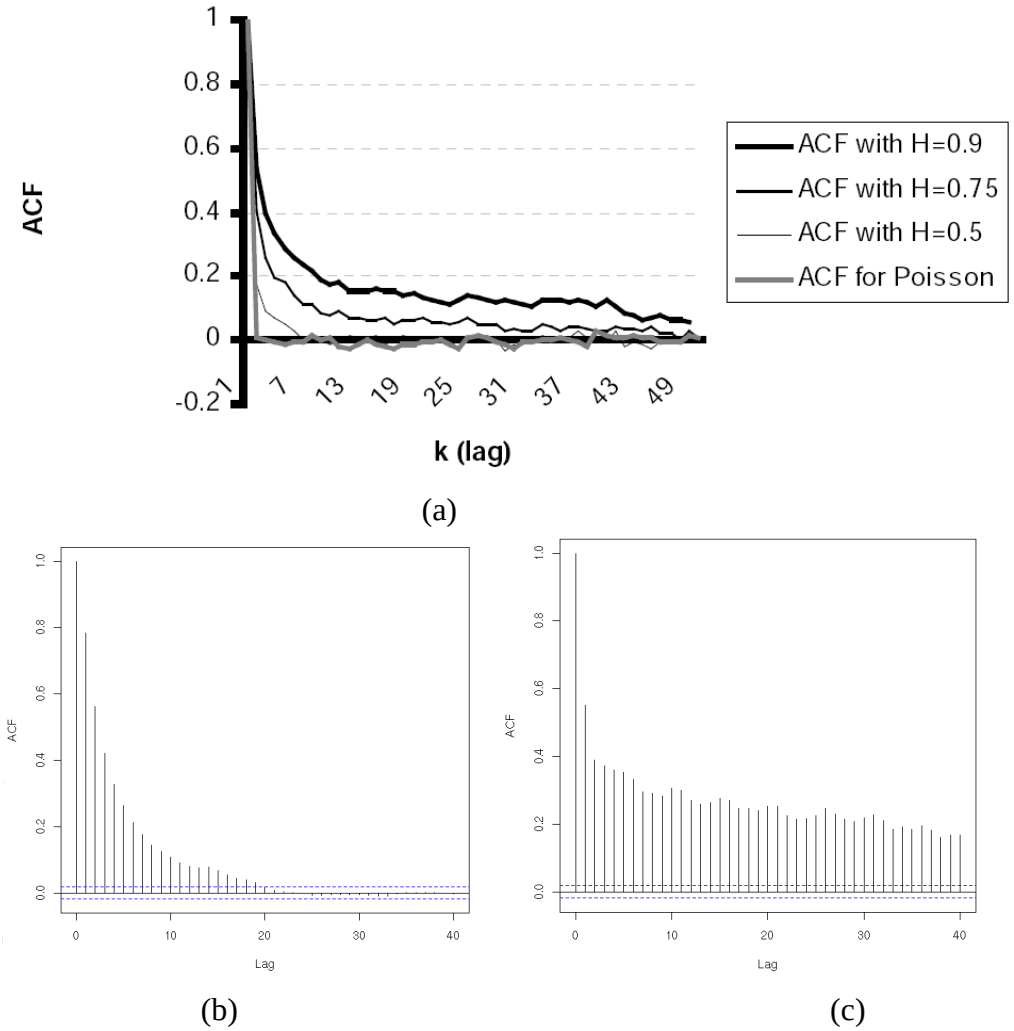


Figure 6.8: ACF graphs showing (a) Analysis of SupFPR from [31], (b) a simulated day in Cluster 5 and (c) an actual day in Cluster 5

The mean packet rate is one of the few measured statistics that is accurate to the data the simulation parameters were based on, which leads to the interarrival times of packets being the other accurate statistic. The mean packet size is only three fifths of what it

should be, which brings down the byte rate considerably. The much lower number of flows per second with the packet rate being what it is means that each flow must involve a lot more packets. It is unusual that the average data transfer rate of each flow is about 129k/s but the average packet size is 284 bytes – this is a lot of very small packets.

Such a small number of flows with a small average packet size could perhaps be indicative of a web proxy aggregating connections, but that should only be evident at one end of the connection. A large number of users coming from a single host would still be hitting a broad range of addresses on the remote end. Small packets could be the result of instant messaging style applications using HTTP to gain access past restrictive firewalls or such, but SupFPR is meant to generate traffic that is representative of normal browsing traffic, not exclusively instant messaging. Bulk transfers or webpages with a lot of content would likely be sent in MTU sized packets and should raise the average somewhat, but they have very little representation. In any case, it fails to match with the parameters it was given, even those that were made explicit. It is unusual that so much of the traffic generated should be so different to the specifications that its was given, but repeated runs of the simulation with different random seeds and a variety of other parameters still gives results that are quite dissimilar to those expected.

Classification of Results Using Weka

The results of the simulations were classified in Weka using the same models used to classify the original data and to classify the results of the PackMime simulations in Section 6.3.1. The simulation based on the weekdays during termtime gets classified into the same grouping as the observations that generated the parameters for it. This is promising in that it matches the intended group, but a lot of the important statistics are still very different. Much like with the PackMime simulations, it seems more an accident that it happens to be classified correctly because of the strong emphasis on mean packet rates and the broad ranges of acceptable values. Given the parameters for

cluster 5, the average results from simulation are classified as belonging to cluster 5 also.

The simulation of a weekend during termtime bears the same resemblance to the original data as the simulation of the weekday does to its originals. Packet count is about right though a little low, and most other measures are also low which sees it fall into cluster 1, which contains weekends during times when there are less students, outside of termtime. Matching mean packet rates places it in a grouping that is similar to where it should be, despite the rest of the statistics.

Classification Using Other Methods

The quantile-quantile plot in Figure 6.9 is a closer match than that for the PackMime simulations, but it still is not a good fit. The packet rates from simulation are on the whole slightly larger than those observed on Tuesday 5 May 2004. At the lower rates there is a lot of variance, but at higher rates both sets of data share similar distributions. Most of the differences here are a matter of scale.

6.3.3 Simulation Using State Machines

The complexity of constructing a simulation using this method and the long run times meant that it was only possible to simulate a single set of circumstances, in this case a weekday during termtime. Despite being based very heavily on a particular day, Figure 6.10 shows that this traffic generator matched the summary statistics least well. In most cases the simulated values were well below what was expected, though it did generate traffic with a Hurst parameter approaching that of the observed days. The shape of the traffic was a good match for the original though this is mostly due to the heavy reliance it had on using the actual flow start times. Compare the two graphs in Figure 6.11 – on the left is the observed packet rate measured each second from original day, the 25 of March 2005, that was used to generate the state machine and flows for this simulation,

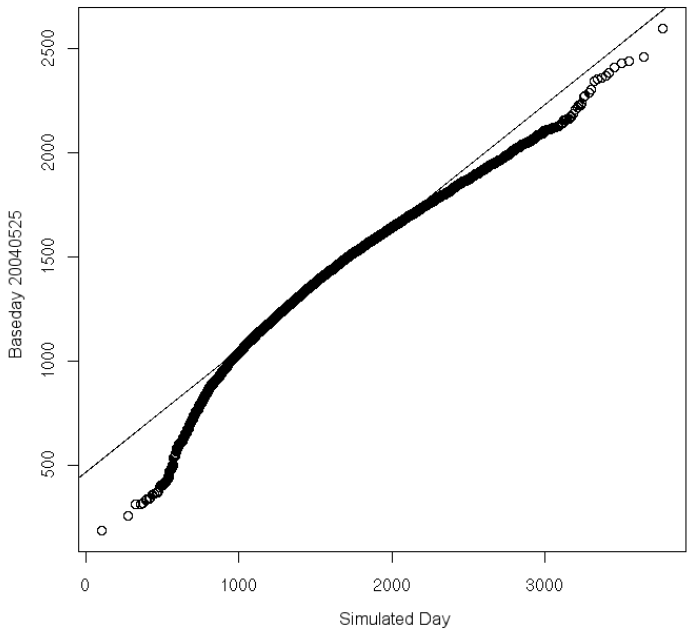


Figure 6.9: Quantile-Quantile Plot Between Mean Packet Rates of SupFPR Simulated Cluster 5 Day and 20040525

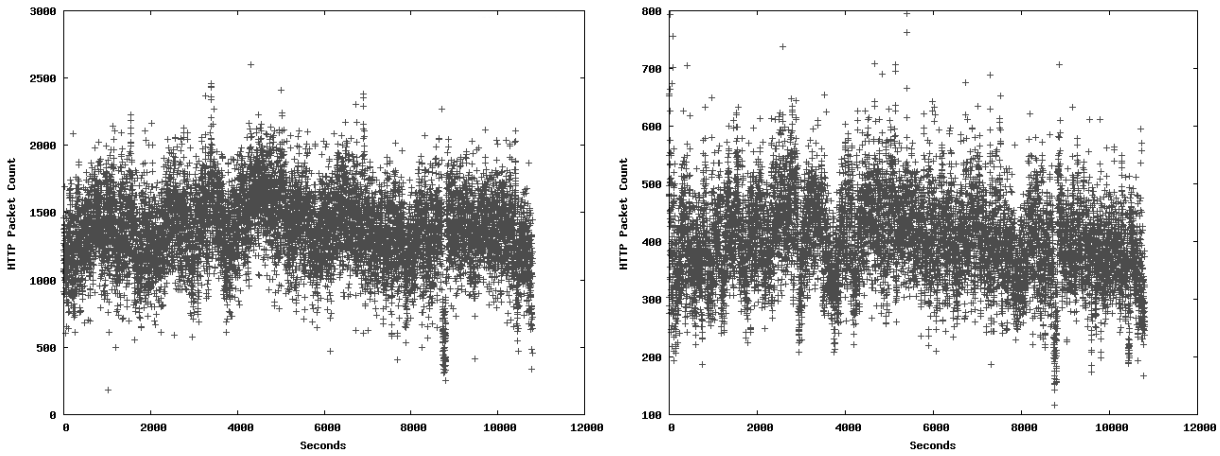
and on the right is the observed packet rate from the simulation. By basing it off only a single day, the simulation is over-fitted in that it includes most of the nuances, the unusual peaks and troughs that are not present on other days that belong to the same cluster and are unique to this one. Because the two graphs are on different scales, their similarity could lead the casual observer to believe them to be the same though they are not. It is hoped that as it is developed, this model will be able to maintain this high degree of similarity but on the same scale as all the source days.

There are a few known issues with this method of traffic generation as it currently stands that may have contributed to some of the poor results. In order to lower memory requirements, agents within *ns* are reused after a timeout period of 60 seconds in which there is no data transfer. At this stage in the development of the traffic generator there is no facility in place to close these reused connections, nor to re-open them. This means that any reused connections will not go through proper TCP connection establishment,

	Cluster 8	Cluster 8 Simulated
Mean Packets/sec	1375.798	400.8929
Packets/sec SD	257.7751	74.41457
Hurst Parameter	0.87	0.78
Mean Bytes/sec	625110.8	21454.96
Bytes/sec SD	133176.1	4098.210
Mean Flows/sec	263.6097	55.20657
Flows/sec SD	49.725	8.9129
Mean Interarrival	0.0007725	0.0025800
Interarrival SD	0.0003166	0.0005262

Figure 6.10: State Machine Results

the 3 way handshake of SYN, SYN/ACK, ACK does not take place, nor are FIN packets sent. Assuming that there is no packet loss each flow is potentially missing up to seven packets. If they were present this would bolster packet numbers somewhat but likely still not to the degree required.



(a) (b)

Figure 6.11: HTTP Packets Per Second for (a) Tuesday 25 May 2004 (b) Simulation Based on Tuesday 25 May 2004

Classification of Results Using Weka

Classifying the results of this simulation using the models previously built using Weka places it in cluster 1 – a weekend during a time when there are few students present. It should be in cluster 8, representing a weekday during termtime early in 2004 but the low mean packet rate causes it not to fit where it should. Again this is caused by the large importance placed on the mean packet rate in determining which cluster a day should belong to.

Classification Using Other Methods

Besides the packet rate of the simulation being much lower than that of Tuesday 5 May 2004, the actual distribution of packet rates is very similar. There is a small amount of difference at the higher and lower rates, but overall they are a good match. This is good in that it shows this method of simulation is promising and could eventually return good results.

Comparing the two distributions using the Kolmogorov-Smirnov test also gives a poor result and shows that the two distributions are in no way related. Again, this is likely caused by the difference in scales, which creates a large difference in the cumulative density functions even though they are very similarly shaped.

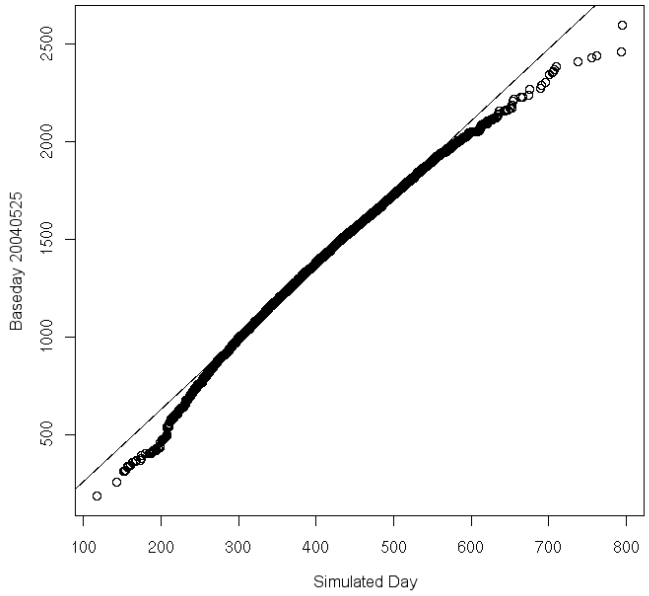


Figure 6.12: Quantile-Quantile Plot Between Mean Packet Rates of State Machine Simulated Cluster 8 Day and 20040525

Chapter 7

Conclusions

620 consecutive days from the *Waikato 1* dataset were examined in order to determine what made them similar. Observable properties of the traffic such as packet and byte counts were measured and used to describe each day so that comparisons could be made between it and other days. Initially these took the form of direct comparisons between values and their distributions which was useful but not accurate enough to properly determine if any two days fell into the same category or not. Taking this further and using an Expectation Maximisation algorithm to estimate cluster membership based on these values produced sensible groups of days, and once these were known the reasons behind the choices could be investigated. Decision trees and other methods were used to determine the attributes important to classifying the data. Of course these techniques are only as good as the data they are given, and though the statistics used are those that have been identified as being important to describing network traffic, it is not clear that they are the best ones for spotting the important differences between days.

It is clear that the most basic statistics such as the mean packet rate are strong indicators of what sort of day each one is, at least in this setting. In part this is likely due to the University network and the limitation of the study to HTTP traffic – the monetary costs that most users must bear undoubtedly restrict the way in which the network is used. Short transactions such as fetching webpages are the norm, with few bulk trans-

fers or any other sort of longer lived flow. With only really a single use of HTTP, all traffic looks very much the same apart from the quantity of it which leads to the mean packet/byte rates standing out as so important. These are directly affected by the number of users present, which is in turn based on the University calendar and the day of the week. The clustering algorithm picked up on this and so most of the clusters correspond to different periods during the year as students arrive and leave.

If another organisation was to host a measurement point they might find that there are better ways to characterise their traffic because they have more varied uses of HTTP and more stable user numbers. Assuming a five day working week, weekends would likely still stand out as having little traffic but weekdays could be categorised based on the different types of task that were performed during each day rather than solely on the amount of traffic.

In any case, it appears that HTTP models for traffic generation have a way to go before they can accurately model specific parameters such as might represent a class of day. Most of the models investigated are approximately correct, but are not close enough to be considered the same. Decisions made in the construction of the models may mean that it is modelling a different style or use of HTTP, or perhaps the interaction with the simulated network prevented them from behaving quite as they should. They could be made to match the observed traffic by overestimating the values of parameters though this is a work-around, not a solution.

7.1 Future Work

The decision to limit investigation to only HTTP packets could have been part of the reason why user numbers have been found to be so important. The limited number of tasks for which HTTP appears to be used within the University means that there is no real variation in observed traffic, just different scales. It should be examined whether

this property is true across other traffic types that are seen on the University link, and also if it holds when all the traffic is aggregated. Most types of traffic are still going to be influenced by the number of users present that are creating traffic, but some kinds of traffic are more likely to be generated externally or according to other patterns. It would also be useful to measure the behaviour of HTTP at other locations to see if the traffic maintains the sameness about it that is seen at the University.

The measurements should be continued into the future in order to expand the data set available, providing more information upon which to compare days. Watching this with knowledge of changes occurring in the network would be beneficial to spotting and understanding the reasoning behind cluster membership that goes beyond the number of users present.

Being able to compare the traffic of different days has other applications beyond building and validating simulations. Detecting unusual events within a network is a difficult area of research and some of the techniques described here could be useful. If a model of what a day is expected to look like can be built, and acceptable ranges for the distribution of its attributes decided on, then comparisons are simple. It would also be simple to point out the aspects of the traffic that are in violation of the given constraints.

Bibliography

[1] Mark Allman, Vern Paxson, and W. Richard Stevens. TCP Congestion Control. RFC-2581, April 1999.

[2] Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 151–160, 1998.

[3] Bro Intrusion Detection System. http://www.bro-ids.org/, Accessed 2005.

[4] Ramon Caceres, Peter B. Danzig, Sugih Jamin, and Danny J. Mitzel. Characteristics of wide-area tcp/ip conversations. In *Proceedings of ACM SIGCOMM*, pages 101–112, September 1991.

[5] Jin Cao, William S. Cleveland, Yuan Gao, Kevin Jeffay, F. Donelson Smith, and Michele C. Weigle. Stochastic models for generating synthetic http source traffic. In *Proceedings of IEEE INFOCOM*, 2004.

[6] Endace dag 3.5e network monitoring interface card. http://www.endace.com/dag3.5E.htm, Accessed 2005.

[7] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot. Packet-level traffic measurements from the Sprint IP backbone. *IEEE Network Magazine*, 17(6):6–16, 2003.

[8] Ian Graham, Murray Pearson, Jed Martens, and Stephen Donnelly. Dag - a cell capture board for ATM measurement systems. Technical report, Department of Computer Science, University of Waikato, 1997.

[9] IP Performance Metrics Working Group. http://www.ietf.org/html.charters/ippm-charter.html, Accessed 2005.

[10] Y. Joo, V. Ribeiro, A. Feldmann, A. Gilbert, and W. Willinger. Tcp/ip traffic dynamics and network performance: A lesson in workload modeling, flow control, and trace-driven simulations. *SIGCOMM Computer Communication Review*, 31(2):25–37, April 2001.

[11] Sunil Kalidindi, Henk Uijterwaal, René Wilhelm, and Matt Zekaukas. Comparing Two Implementations of the Delay and Loss Metrics (ANS and RIPE-NCC). Presentation given to the IPPM Working Group at the Forty-Fourth IETF meeting - http://www3.ietf.org/proceedings/99mar/slides/ippm-impl-99mar.pdf, March 1999.

[12] D. Kwiatkowski, P. C. B. Phillips, P. Schmidt, and Y. Shin. Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54:159–178, 1992.

[13] Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[14] Will E. Leland and Daniel V. Wilson. High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection. In *Proceedings of IEEE INFOCOM'91*, pages 1360–1366, April 1991.

[15] Libtrace. http://research.wand.net.nz/software/libtrace.php, Access 2005.

[16] Tony McGregor and Hans-Werner Braun. Automated event detection for active measurement systems. In *Proceedings of the PAM2001 workshop on Passive and Active Measurements*, pages 23–32, April 2001.

[17] National Laboratory for Applied Network Research (NLANR). http://www.nlanr.net/, Accessed 2005.

[18] Richard Nelson, Daniel Lawson, and Perry Lorier. Analysis of long duration traces. *SIGCOMM Comput. Commun. Rev.*, 35(1):45–52, 2005.

[19] NLANR Active Measurement Project. http://amp.nlanr.net/, Accessed 2005.

[20] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[21] Rmetrics. http://www.itp.phys.ethz.ch/econophysics/R/index.html, Accessed 2005.

[22] Snort. http://www.snort.org/, Accessed 2005.

[23] tcpdump. http://www.tcpdump.org, Accessed 2005.

[24] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/, Accessed 2004.

[25] Steve Uhlig. Non-stationarity and high order scaling in tcp flow arrivals: a methodological analysis. *ACM SIGCOMM Computer Communication Review*, 34(2):9–24, 2004.

[26] Waikato Internet Trace Storage (WITS). http://wand.cs.waikato.ac.nz/wand/wits/, Accessed 2005.

[27] WAND Network Research Group. http://www.wand.net.nz, Accessed 2005.

[28] Web Traffic Generation in NS-2 with PackMime-HTTP. http://dirt.cs.unc.edu/packmime/, Accessed 2005.

[29] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, June 2005.

[30] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. Profiling internet backbone traffic: behavior models and applications. *SIGCOMM Computer Communication Review*, 35(4):169–180, 2005.

[31] M. Yuksel, B. Sikdar, K. S. Vastola, and B. Szymanski. Workload generation for simulations of wide area networks and the internet. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 93–98, 2000.