LAN Traffic Locality: Characterization and Application

Neeraj Gulati
Carey L. Williamson
Richard B. Bunt
Department of Computational Science
University of Saskatchewan

January 11, 1993

Abstract

Network traffic locality is a special case of the locality phenomenon commonly seen in computer systems. The source and destination addresses seen in network packet traffic are observed to be non-uniformly distributed, both in time and space. Several recent research efforts have identified the presence of network traffic locality, and suggested ways to exploit the locality property in the design and operation of computer networks. While the intuitive notion of network traffic locality is fairly well understood, there is still a distinct lack of formal definitions or measures of network locality in the literature.

In this paper, we propose four new measures to characterize network traffic locality: persistence, concentration, address re-use, and reference density. These measures are motivated, in part, by locality characteristics identified in other domains (e.g., memory reference studies of program behaviour, and file referencing characteristics), but reflect properties important to network applications. Definitions are given for each measure, and measurement data collected from a local area network environment are used to quantify each measure. Finally, to demonstrate a potential application for the network traffic locality property, we present a brief analysis of how this property might be used to advantage in the design of a network interconnection device.

1 Introduction

Network traffic locality is a special case of the locality phenomenon commonly seen in computer systems. The source and destination addresses seen in network packet traffic are observed to be non-uniformly distributed, both in time and space. Successive packets are rarely independent of each other, and aggregate packet traffic is frequently dominated by a small number of transmitting hosts, even over significant intervals of time.

In packet-switched networks, the locality property is a natural artifact of the protocols and applications used for data transmission. Bulk data transfer activities, such as file transfer, generate bursts of packets between a particular pair of hosts. Interactive terminal traffic also generates highly correlated bidirectional traffic streams. Models such as the *packet train model* [8] or the *packet tandem model* [10] have been proposed to explain such behaviour. Such models have proven to be useful not only in the improvement of techniques for modelling workloads, but also in their application to network design.

Several recent research efforts have identified the presence of network traffic locality [8, 9, 11], and suggested ways to exploit the locality property in the design and operation of computer networks [2, 6, 14]. Exploiting locality, usually by caching recently referenced items, has been crucial in the success of virtual memory systems, multiprocessor systems, and distributed file systems, where the access time of the cache is an order of magnitude faster than access to main storage. It remains to be seen how important the notion of locality is to the design of future networks. Several recently proposed network architectures [13, 15] rely explicitly on network locality for efficient operation. Exploiting locality may become even more important as networks become larger and faster [9].

While the intuitive notion of network traffic locality is fairly well understood, there is a distinct lack of formal definitions or measures of network locality in the literature. The purpose of this paper, then, is to precisely characterize the salient aspects of network traffic locality in a local area network (LAN) environment (i.e., in the packet traffic seen by a network interconnection device between LANs). In particular, we propose four new measures of network traffic locality: persistence, concentration, address re-use, and reference density. These measures are based, in part, on locality characteristics identified in other domains, such as program behaviour [4], and file referencing behaviour [12]. Definitions are provided for each measure, and measurement data collected from a local area network environment is used to quantify each measure.

The remainder of this paper is organized as follows. Section 2 motivates and defines our four locality measures. Section 3 presents our measurement methodology, and demonstrates our locality measures on packet trace data collected in a local area network environment. This demonstrates the extent to which these properties exist in a typical (albeit small) network environment. Section 4 discusses a possible application of these locality properties to the design of network interconnection devices, by presenting a brief analysis of the implications of the observed locality properties on the performance of a network interconnection device. Finally, Section 5 presents our conclusions, and suggested directions for future research.

2 Locality Characteristics

Studies of memory and file referencing behaviour have found that such references typically display non-random characteristics [4, 12]. References are distributed non-uniformly in both time and space with small subsets of information favoured over relatively long time intervals. The term "locality of reference" has been used to describe this phenomenon.

The General Principle of Locality [5] has been described as:

1. During any interval of time the resource demands of the system are distributed non-uniformly over the total set of available resources.

2. The resource demand pattern for a given resource tends to change slowly with time.

3. The correlation between the immediate past and immediate future demands tends to be high; the correlation between disjoint resource demand patterns tends to zero as the elapsed time between them tends to infinity.

In the networking context, the "resource demand pattern" is the sequence of source and destination addresses seen in the network packets of an aggregate network traffic packet trace.

Two aspects of locality have been identified, referred to as *temporal locality* and *spatial locality*. Temporal locality means that the information currently in use is likely to be used again in the near future. Spatial locality implies a high probability of references to neighbouring addresses. While the notion of neighbouring addresses makes sense in memory referencing and file referencing behaviour, it is not so clear for networks.

The terms *persistence* and *concentration* have also been used to characterize locality behaviour [3]. Persistence refers to the tendency to repeat a particular reference (or group of references) in succession (an example of temporal locality). Concentration, on the other hand, refers to the tendency for the references to be limited (or concentrated) to a small subset of the total reference space.

Two similar characteristics are evident in most analyses of network traffic locality. First, successive packets on the network tend to be highly related. That is, a packet travelling from host A to host B on the network is nearly always followed by another packet from A to B, or a packet from B to A [8]. This is primarily a short-term characteristic, demonstrating temporal locality and persistence. Second, a small number of hosts tend to account for a large number of the packets transmitted on the network. This is primarily a long-term characteristic, demonstrating concentration.

Two locality measures are proposed for each of these characteristics. For short-term characteristics, the measures *persistence* and *address re-use* are proposed. For long-term characteristics, the measures *concentration* and *reference density* are proposed. These measures are defined in more detail below.

2.1 Persistence

Persistence refers to the tendency for an address, once referenced, to be referenced again and again, consecutively, on the network. A burst of consecutive packets from the same sender would be said to have high source address persistence. A burst of consecutive packets heading to the same destination would be said to have high destination address persistence. Workload generated by bulk data transfers, for example, would be expected to yield reference streams with high source and destination address persistence.

The presence of persistence in network traffic would make very simple caching strategies effective when handling packets. For example, some implementations of TCP use a single element cache to store the protocol control block for the most recently active transport-level connection. With a highly persistent reference stream, only the first reference causes a cache miss, and all repeated references are resolved in the cache.

Thus our first step towards characterizing network traffic locality is with a measure of persistence. The aim is to find the probability of seeing w consecutive packets from the same source, or to the same destination. That is, starting from a given packet at a random point in the reference stream, what is the probability that the next w packets are from the same source as the current packet (source address persistence) or to the same destination as the current packet (destination address persistence).

The persistence measure P_w is computed for various values of w, where w is called the *persistence window size*. The calculation is as follows. Let N_w denote the number of times that the addresses of the next w packets in the trace all match the address of the current packet. Let T_w denote the total number of places in the trace where this persistence could possibly occur (i.e., $T_w = n - w + 1$, where n is the number of packets in the trace). P_w is then simply the ratio

$$P_w = \frac{N_w}{T_w}$$

The persistence measure P_w is computed separately for source and destination addresses.

2.2 Address Re-use

One drawback of the persistence measure defined above is the requirement for related packets to or from the same address to appear consecutively on the network. In a packet-switched network that is statistically multiplexed among many hosts, the presence of persistence may be completely masked in the aggregate traffic. Thus a new measure, address re-use, is proposed as a generalized form of persistence. Rather than requiring related packets to appear consecutively on the network, they need only appear "close" together.

Address re-use is defined as the tendency for the address used in one network packet to reappear as the destination address of a future packet. Specifically, address re-use determines the probability that the address (source or destination) used in the current packet appears as the destination address of at least one of the next w packets, for small values of w.

Address re-use provides a more general measure of temporal locality than the persistence measure provides. To study this aspect of locality, the complete reference trace is considered w packets at a time. The window size w determines how far ahead in the reference trace you look.

Destination address re-use, D_w, is computed as follows:

$$D_w = \frac{N_w}{T_w}$$

where N_w is the total number of times that a destination address is re-used within the next w packets, and T_w is the total number of places where destination address re-use could possibly occur in the trace. A similar method is used to determine the source address re-use S_w.

2.3 Concentration

Concentration is defined as the tendency for the host addresses seen in a particular period of the trace to be limited to a small subset of the total host addresses seen in the entire trace. During any finite interval of w packets, activity is concentrated on a small number of hosts called the *working set* of active hosts.

In general, concentration is important for applications such as cache management. It is important to know how much of the total address space that is referenced during the entire duration of a trace is referenced during a specific interval. If only a small portion of the total address space is referenced, only a small portion of it needs to be kept in the cache. Our proposed concentration measure addresses this issue. Again, the measure is defined separately for source addresses and destination addresses.

The working set model, used extensively in the characterization of program behaviour [4], seems to provide a good framework for expressing the notion of concentration in a reference stream. If w is the working set window size, then the working set size W_w is the number of different hosts that send (or receive) at least one of the next w packets. So when sets of w packets are considered at a time, W_w shows the number of different addresses that appear in such sets. The smaller the working set size, the higher is the degree of concentration in the trace.

One of the drawbacks of the traditional working set model as a measure of concentration is that the W_w values are not directly comparable for different values of w. For example, if $W_{100} = 20$ and $W_{200} = 28$ for the same trace, it does not necessarily mean that higher concentration is being observed for the window of size 100. In fact, one could argue that the window of size 200 shows higher concentration since there are only 28 different addresses observed (14% of a possible 200), while for a window size of 100, 20 different addresses appear (20% of the possible 100). So the concentration for different window sizes cannot be compared simply by comparing the corresponding values of the working set size. Instead, concentration should be expressed as a function of both the working set size and the window size.

Defining concentration as the ratio of the working set window size w and the corresponding working set size W_w is one alternative, but is also unsuitable. Consider a window size of 10 and a window size of 100. If there is only one address appearing in all the packets in the window, then the concentration measure for the $w = 100$ window would be 10 times higher than that for the $w = 10$ window, even though both examples exhibit the "same" concentration behaviour, in the sense of one active host dominating all activity. Furthermore, the concentration values fall non-linearly as the working set size increases, making comparisons difficult (e.g., the concentration is 100 when $W_{100} = 1$, 50 when $W_{100} = 2$, 33 when $W_{100} = 3$, etc.) In other words, this measure of concentration is still not comparable across working set window sizes.

To overcome these deficiencies, a measure for concentration that is an extension to the working set model is proposed. The new measure is normalized to values between 0 and 1. If the addresses of all w packets in the working set window are the same (i.e., $W_w = 1$), then the concentration is said to be maximum ($C_w = 1$). If all the w packets have different addresses (i.e., $W_w = w$), then the concentration is said to be minimum ($C_w = 0$). In between these two extremes, the concentration C_w is defined by the following linear relationship:

$$C_w = \frac{w - W_w}{w - 1}, w > 1$$

This concentration measure captures the intuitive notion of concentration. The smaller the number of different destination addresses in a given subtrace, the higher is its concentration. Also, the longer a subtrace is for a given working set size, the higher the concentration. The new measure is also reasonably comparable across different window sizes.

Another advantage of the new concentration measure is its direct relationship to cache size. Given a trace of length w and concentration C_w, a cache size of approximately $(1 - C_w)w$ entries is required to keep the active working set in the cache. The longer the trace, or the lower the concentration, the larger the cache size required. The concentration measure is applied separately to both source and destination addresses.

2.4 Reference Density

Reference density is defined as the tendency for a small number of hosts to account for a large proportion of the total network traffic. Given h, the number of hosts active during a trace, and n, the number of packets in the trace, it is rare that all the hosts send and receive an equal number (n/h) of these packets. Over a period of observation, there are usually certain hosts that send or receive more than their share of the packets.

Reference density is a measure of this long-term concentration property. The reference density, R_n, is defined as the proportion of references accounted for by the n most active hosts. Reference density is computed from the traces by sorting the hosts in order of activity, and computing their cumulative contribution to the aggregate traffic. Again, this measure is applied separately to both source and destination addresses.

2.5 Host-pair Locality

Up to this point characteristics of source and destination address locality have been considered in isolation. In this section an aspect of locality that involves both the source and the destination address is considered.

Host-pair locality is defined as the frequency of occurrence of at least one of the next w packets that is traveling between the same host pairs as the current packet. This measure is similar to the address re-use measure discussed in Section 2.2. This measure includes unidirectional as well as bidirectional traffic between pairs of hosts.

3 Measurement Results

3.1 Methodology

Traces of the activity on an Ethernet local area network were captured using an HP 4972A LAN Protocol Analyzer. The LAN Analyzer is a hardware network monitoring device that operates at the data link level. That is, it understands Ethernet frames (packets) and (48-bit) Ethernet addresses. It has no knowledge of higher level applications and protocols (e.g., TCP/IP). The device was operated in promiscuous mode, attached to one of the LAN segments in our research laboratory. The LAN Analyzer has a 1 megabyte buffer memory for incoming data, and a 20 megabyte Winchester disk. After capturing each trace, the data was transferred to a file server for later analysis.

To provide a variety of samples, ten different traces were obtained during the winter of 1992 and formed the basis for the analysis that follows. Details of the ten traces are given in Table 1.

A trace consists of a timestamp, a source address, and a destination address for each packet seen on the network[1]. The size of each trace was limited by the amount of disk space available on the LAN Analyzer when the trace was collected. In general, the disk provides enough space for about 5 minutes of tracing for aggregate network traffic. The exact duration of each trace was a function of the level of network activity during the trace period.

[1]The analyses considered in this paper are only for data link level (Ethernet) addresses. Similar analyses could be performed on higher-level addresses, such as IP addresses, or TCP port identifiers.

Table 1: Description of the Traces

Trace	Date	Start Time	Size(pkts)	Duration	Hosts	Load(pkts/sec)
1	Feb 10	14:46:43	50068	267 secs	612	188
2	Feb 11	15:20:47	50076	287 secs	630	175
3	Feb 12	18:24:46	80134	344 secs	515	233
4	Feb 14	21:29:16	80326	620 secs	633	130
5	Feb 16	15:06:09	80116	332 secs	511	241
6	Feb 18	13:10:11	60065	322 secs	665	187
7	Feb 20	00:32:01	60922	495 secs	503	123
8	Mar 11	09:06:28	39403	300 secs	534	131
9	Mar 12	11:16:06	51573	267 secs	641	193
10	Mar 14	03:25:43	50970	284 secs	535	180
Averages:			60365	351 secs	578	178

Since our local area network is connected to the campus network and the Internet, the traffic observed in any trace period involved hosts on other parts of the internetwork. From 503 to 665 distinct Ethernet addresses were seen during the periods traced. Of the hosts observed in the traces, 26 were identified as machines in our research laboratory, and 21 others as part of our department's teaching laboratory. The remaining addresses represent unidentified machines that are elsewhere on the internetwork.

To obtain a representative sample of the general activity on the network, traces were captured on different days of the week and at times when different types and levels of activity were expected. Each trace contains a time-stamped reference string of 39,000 to 80,000 packets with a duration varying from 267 seconds to 620 seconds.

Ten traces represent a large body of experimental work. To conserve space, only three of the traces are selected for presentation here[2]. The selected traces are Trace 5, Trace 6 and Trace 8. Trace 5 was chosen because it has the highest average load in terms of packets per second (241 packets/sec). Trace 8 was chosen because of its low load (131 packets/sec). Trace 6 was chosen because it has the highest number of active hosts (665), and a load that is close to average (187 packets/sec).

3.2 Persistence

Our first attempt to characterize network traffic locality is with a measure of persistence. The aim is to find the probability of seeing w consecutive packets from the same source, or to the same destination. That is, starting from a given packet at a random point in the reference stream, what is the probability that the next w packets are coming from the same sender (source address persistence) or heading to the same destination (destination address persistence) as the current packet.

[2]The complete results for all ten traces are available in [7].

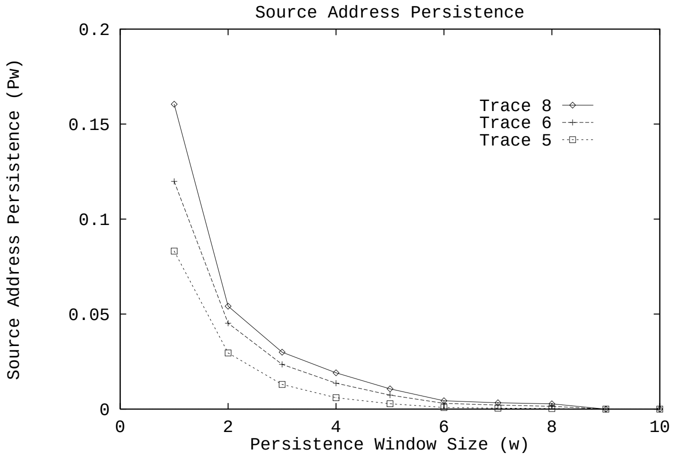

Figure 1: Source Address Persistence

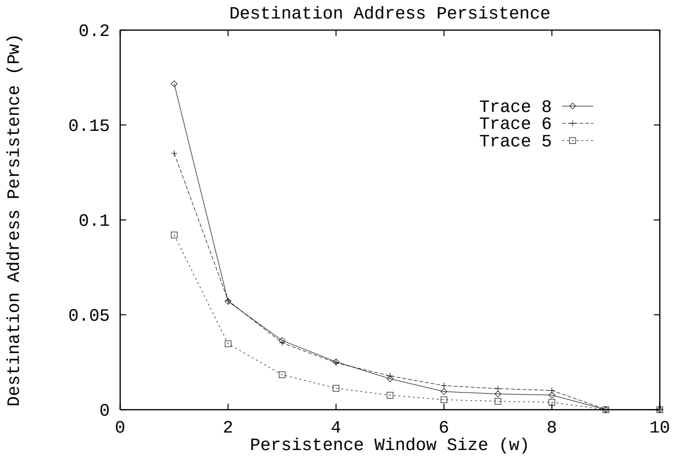

Figure 2: Destination Address Persistence

The persistence data obtained for the three selected traces is plotted in Figure 1 and Figure 2. Figure 1 shows source address persistence, and Figure 2 shows destination address persistence. Both source and destination addresses exhibit very similar characteristics.

For all traces, persistence values drop sharply with increasing window size. Long sequences of persistent references are rare. The longest persistent sequence seen in any of the traces was $w = 9$.

Overall, the persistence values for each of the traces are low. For Trace 5 in Figure 2, for example, some destination address persistence is observed only 18% of the time, and no persistence is seen 82% of the time (i.e., 82% of the time the next packet is destined to a different address than the current packet).

From the analysis of persistence it is concluded that there aren't many instances when successive packets are observed on the network destined for the same host.

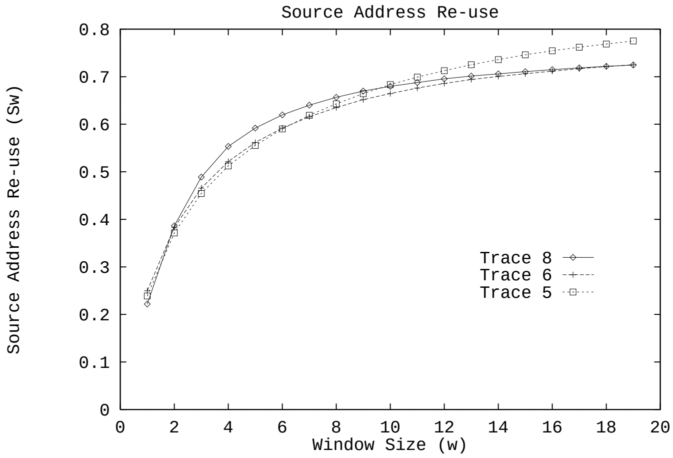

Figure 3: Source Address Re-use

There are two possible explanations for this behaviour. First, low persistence might be due to bidirectional traffic being dominant. That is, packets from A to B are followed by packets from B to A and vice versa. Second, low persistence may be due to the statistical multiplexing of the network among many hosts. That is, since the traffic observed is the aggregate traffic between various hosts on the network, many individual traffic streams may be interleaved, masking their persistence.

The load on the network appears to have a bearing on the overall persistence observed, providing some basis for the latter hypothesis. Trace 5, which has a much higher load than Trace 6 and Trace 8, shows lower address persistence. The persistent traffic is less frequently masked when the other hosts on the network are not very active (low load). Regardless of the network load, however, our results indicate that persistent traffic is not a dominant characteristic of network traffic locality.

3.3 Address Re-use

Address re-use is defined as the tendency for the address (source or destination) used in a particular network packet to reappear "soon" as the destination address in future packets. These packets need not appear consecutively on the network.

Address re-use characteristics for the three selected traces are shown in Figure 3 and Figure 4. Figure 3 shows source address re-use (the probability that the source address in the current packet appears as the destination address of at least 1 of the next w packets), and Figure 4 shows destination address re-use (the probability that the destination address in the current packet appears again as the destination address in at least 1 of the next w packets),

The results show significant destination and source address re-use. The larger the window size w, the higher is the probability that an address is used again within that window. The rise in the re-use values is steep for small window sizes, indicating the presence of temporal locality in the traffic. For larger window sizes, the rise

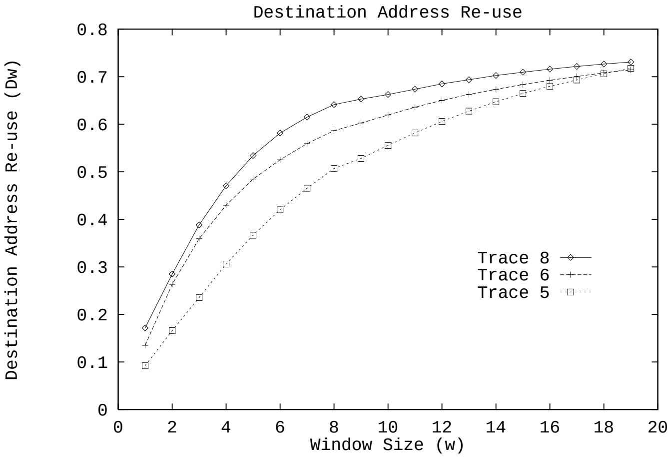

Figure 4: Destination Address Re-use

is slower: once an address becomes "stale", the probability of it being re-used in the future diminishes, resulting in a smaller increase in the address re-use values. Source address re-use values are higher than destination address re-use values for small window sizes w.

The load on the network also has an effect on the address re-use results. The trace with highest load (Trace 5) has the lowest destination address re-use values (Figure 4). Trace 5 "catches up" with the other traces at larger window sizes since the active working set of hosts is fairly small. That is, if a sender is going to send another packet to a specific destination, it will probably be among the next 20 packets sent on the network.

Overall, the address re-use results suggest that there is significant predictability in the aggregate network traffic. Aside from its application to the modelling of network traffic, many network system design problems can benefit from this type of predictability in the traffic. For example, caching could be used to improve the performance of network interconnection devices. By caching the destination address and the source address of recently received packets, a prediction can be made about the destination addresses of the next few packets. The extent to which this predictability can be exploited was investigated by simulation and is reported in Section 4.

3.4 Concentration

Concentration is defined as the tendency for the host addresses seen in a particular period of the trace to be limited to a small subset of the total hosts. Two possible measures of concentration are the working set size and our new concentration measure C_w.

Tables 2 and 3 show the working set size characteristics for the selected set of traces, for source and destination addresses, respectively. The working set size shown is the average working set size (rounded to the nearest integer) over the full duration

Table 2: Source Address Working Set Sizes for Different Window Sizes

w	10	30	60	100	150	200	300	400	500
Trace 5	6	13	18	25	32	38	50	61	71
Trace 6	6	12	20	28	38	46	62	75	88
Trace 8	6	14	22	31	46	58	77	93	106

Table 3: Destination Address Working Set Size for Different Window Sizes

w	10	30	60	100	150	200	300	400	500
Trace 5	6	12	16	21	24	28	33	37	40
Trace 6	5	11	16	21	25	29	35	39	42
Trace 8	5	10	15	20	24	29	35	39	43

of the trace, for the indicated value of the working set window size.

These results show that significant concentration exists in the references. For any trace interval of length w, the number of hosts W_w active during that period is significantly less than w, and the increase in W_w with increasing w is clearly sub-linear. Our traffic shows even higher concentration than that observed by Jain [9]. For example, for a window size of 100, the working set size was 40 in the traffic studied by Jain. The reason for the difference in results is the lower number of hosts directly attached to the network used in this study.

Our new concentration measure is plotted in Figure 5 and Figure 6 for a variety of window sizes. Only Trace 5 is shown, since the three selected traces show almost identical behaviour with regard to both the source and destination address concentration. The jaggedness of the curves in these figures is due to integer roundoff in the working set size computation.

Concentration increases with window size initially, but flattens out for larger window sizes. The concentration seen in an interval of 200 packets is almost the same as that seen in an interval of 400 packets. In other words, the rate at which new addresses appear in the trace (causing misses in a cache) is fairly constant throughout the trace. In all the traces studied, both concentration measures asymptotically approach 1 as $w \to \infty$.

Destination addresses show significantly higher concentration than source addresses. For example, at a working set window size of $w = 500$, destination address concentration is 0.92, while source address concentration is 0.86. Applying the cache size estimate $(1 - C_w)w$ from Section 2.3, this means that a cache for source addresses would need to be roughly twice as large as a cache for destination addresses in order to achieve a similar hit rate. This observation is confirmed by the working set size results reported in Tables 2 and 3.

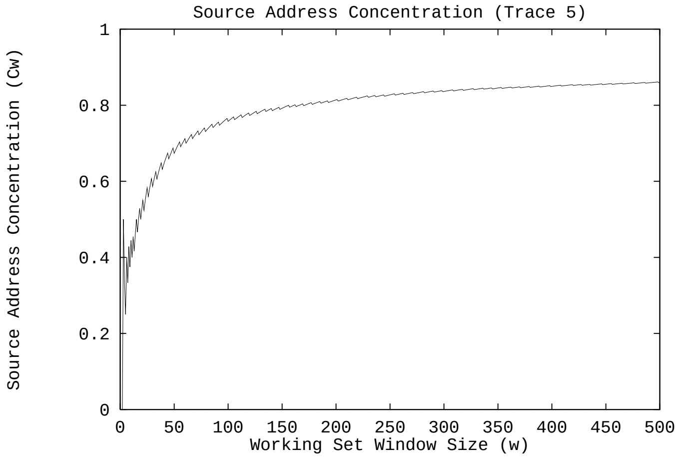

Figure 5: Source Address Concentration for Trace 5

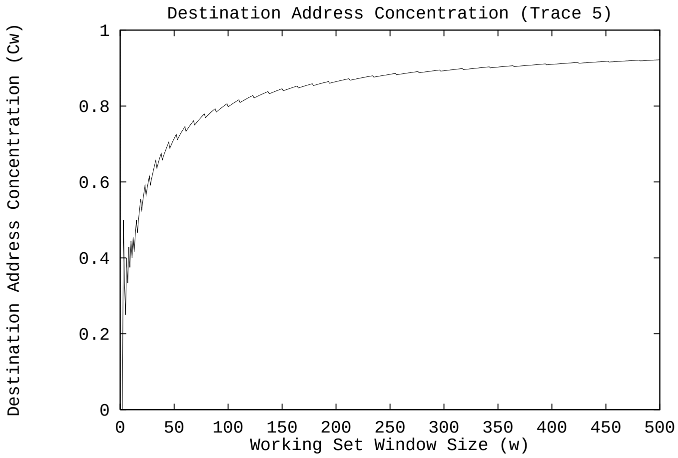

Figure 6: Destination Address Concentration for Trace 5

3.5 Reference Density

Reference density is defined as the tendency for a small number of hosts to account for a large proportion of the total network traffic. Figures 7 and 8 show the graph of cumulative reference density of the 25 most active hosts for the selected set of traces. Source and destination address reference densities are almost identical.

All three traces show similar behaviour for reference density. The curves show a non-linear increase that is rapid for the first few hosts and then levels off. This behaviour is a natural consequence of considering the hosts in order of activity. As more hosts are considered, their contribution to the cumulative reference density diminishes causing the gradient of the curve to decrease.

References are definitely non-uniformly distributed across the address space. In the case of Trace 5 in Figure 8, for example, the 10 most active destinations (2% of

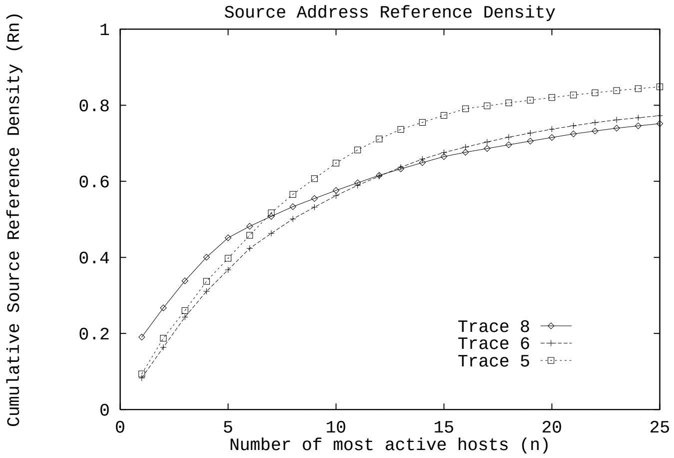

Figure 7: Source Address Reference Density

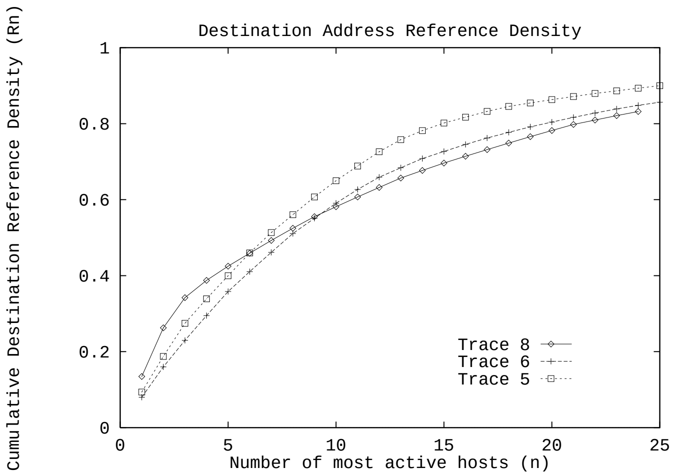

Figure 8: Destination Address Reference Density

the active hosts) receive 65% of the packets, and the 24 most active destinations (5% of the active hosts) receive 90% of the packets. The high reference density values reflect the fact that there are certain hosts on the network that are contributing a large fraction of the traffic. These hosts are the file servers and a few hosts that are used by a large number of users.

The reference density values depend somewhat on the number of active hosts on the network. For example, Trace 6 has more active hosts than either Trace 5 or Trace 8, and has slightly lower reference density.

Overall, our results show even higher reference density than previous studies. For example, results obtained by Jain [9] showed 5% of the hosts accounting for 55% of the traffic. Again, our higher reference density results are most likely due to the smaller number of hosts on the network considered in this study.

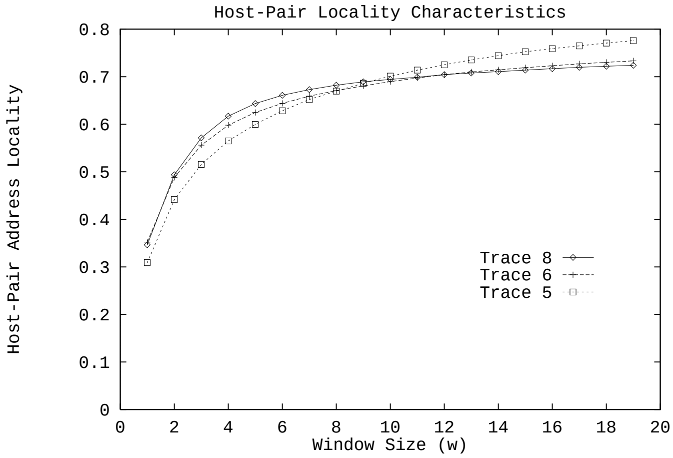

Figure 9: Host-Pair Locality

3.6 Host-Pair Locality

Host-pair locality is defined as the probability that at least one of the next w packets seen on the network is travelling between the same host pairs as the current packet. This measure is similar to the address re-use measures discussed previously, but includes both unidirectional and bidirectional traffic between a pair of hosts.

Results for the host-pair locality measure are shown in Figure 9. Host-pair locality increases quickly for small window sizes, and then levels off, similar to the address re-use characteristics. For small w, values for host-pair locality are even higher than the values for source and destination address re-use.

The measurement results for host-pair locality, combined with the address re-use characteristics, indicate that most of the traffic seen on the network is bidirectional. For example, consider the case where a packet has just been seen travelling from host A to host B. For Trace 5, the destination address re-use for $w = 1$ is $D_1 = 0.09$ (see Figure 4). This means that in 9% of the cases, the next packet on the network is also destined to host B (but not necessarily from host A). At the same time, the source address re-use for $w = 1$ is $S_1 = 0.24$ (see Figure 3), meaning that in 24% of the cases the next packet is destined to A (but not necessarily from B). So in 33% of the cases the next packet is destined for either A or B (but not necessarily from either B or A, respectively). The host-pair locality measure for $w = 1$ is 0.31, indicating that the next packet is between A and B 31% of the time. Thus almost all of the source address re-use is caused by bidirectional traffic. Only 2% of the time $(0.33 - 0.31)$ is address re-use for $w = 1$ caused by a packet to or from some other host on the network.

The calculation above shows that the bidirectional traffic is more than twice the amount of unidirectional traffic (0.24 versus 0.09). This result is not surprising since higher layer protocols, such as TCP, often generate such a pattern (i.e., data packets in one direction, and acknowledgement packets in the other direction, with one acknowledgement packet seen for each data packet). The presence of bidirectional traffic helps to explain the low source and destination address persistence values observed in the traffic and highlights the extent to which the source address of one packet can predict the destination address of the next one.

3.7 Summary

Several observations are evident from our locality measurement results. First, there is relatively little persistence in packet traffic on local area networks. Long sequences of persistent references are rare. Second, the lack of persistence does not mean there is no network traffic locality. Other measures, such as address re-use, do a much better job of quantifying temporal locality. Third, all traces show significant address concentration, address re-use, and reference density. The higher the address concentration, the lower the number of addresses that need to be stored in a cache to predict the destinations of future arrivals. Finally, network traffic locality is both a short-term and a long-term phenomenon. Packets on a local area network are concentrated to small subsets of the address space over significant periods of time.

4 Performance of a Locality-Based Network Interconnection Device

There are several possible applications of the phenomenon of network traffic locality [2, 6, 14, 15]. Among these are implications on the design and performance of network interconnection devices. To take specific examples, the presence of temporal locality suggests that caching recently active host addresses could be effective, and the presence of concentration suggests that only small caches would be required. This issue was investigated further.

In this section, we present a brief analysis of a data link level network interconnection device (i.e., a LAN bridge) designed to exploit network traffic locality by caching recently active host addresses. While the performance benefits achievable by exploiting locality at the data link level in a LAN bridge are minimal, the same locality-based principles can be applied at the network or higher layers of the OSI model, where significantly more software processing is required for each packet (e.g., routing-table lookups for IP addresses at gateways, ARP caches for IP/Ethernet address translation, or TCP packet processing). A LAN bridge is presented only as a simple example of how network traffic locality could be exploited.

A data link level interconnection device is designed to selectively forward packets (or frames) between two or more networks based on the destination address in each packet. The device contains a *forwarding table* that has a list of data link level host addresses, along with the associated network port on which each host is accessible. For each packet that arrives, the device looks up the destination address in the forwarding table, and then either forwards the packet to the appropriate network, or filters (discards) the packet.

By adding a *forwarding table cache*, it may be possible to reduce the number of forwarding table lookups, thereby reducing the forwarding delay of the device. A trace-driven simulation model was developed to estimate the performance of a data link level interconnection device with and without a forwarding table cache. In this paper, we present only two parts of this study:

(1) what replacement policy to use for the cache, and

(2) what fetch policy to use for the cache

A more complete performance analysis can be found in [7].

The cache replacement policy determines which item to remove from a full cache to make room for a newly arriving entry. Five replacement policies were considered: LRU (Least Recently Used), LFU (Least Frequently Used), FIFO (First In First Out), RAND (Random Replacement), and MIN (Optimal Replacement). MIN provides the lower bound for a theoretically optimal cache replacement policy [1], but is not a practically implementable policy since it requires knowledge of future references.

The fetch policy determines what information to put into the cache, and when. Given that the network interconnection device has just seen a packet travelling from host A to host B, is it better to put A (the source address) into the cache, B (the destination address) into the cache, or both A and B into the cache? These fetch policies are called Source Fetch (SF), Destination Fetch (DF), and Source-Destination Fetch (SDF), respectively.

To answer the first question, the simulation program was executed for each of the traffic traces under different replacement policies and cache sizes. In all cases, Destination Fetch (DF) was used for the cache fetch policy.

The simulation results for the five replacement policies are shown in Figure 10. Since all three traces show very similar results, only Trace 5 is shown. The miss ratios drop sharply at first with increasing cache size as active entries are added to the cache. Beyond a certain cache size (here about 15), however, the additional entries retained have a much lower probability of being re-used and the drop in miss ratios becomes more gradual.

As expected, LRU consistently outperforms LFU, FIFO and RAND. LRU works well because of the temporal locality properties (persistence and address re-use) identified in Section 3. LFU performs poorly since the LFU replacement policy does not take into account the fact that a newly fetched entry has a high probability of being used in the next few references. Instead of being retained, this entry is most likely to be the one discarded from the cache when a replacement has to be made, and performance is poor. A FIFO cache does slightly better since it retains recently fetched entries until at least $n-1$ more addresses are received, where n is the size of the cache. Again, however, the performance of FIFO is relatively poor since replacements are not done on the basis of recent usage. RAND, which indiscriminately picks an entry to be discarded when a replacement has to be done, performs as well as LFU and FIFO. Beyond a certain cache size (here 50), because of the large number of entries in the cache, the difference in performance of all policies becomes marginal.

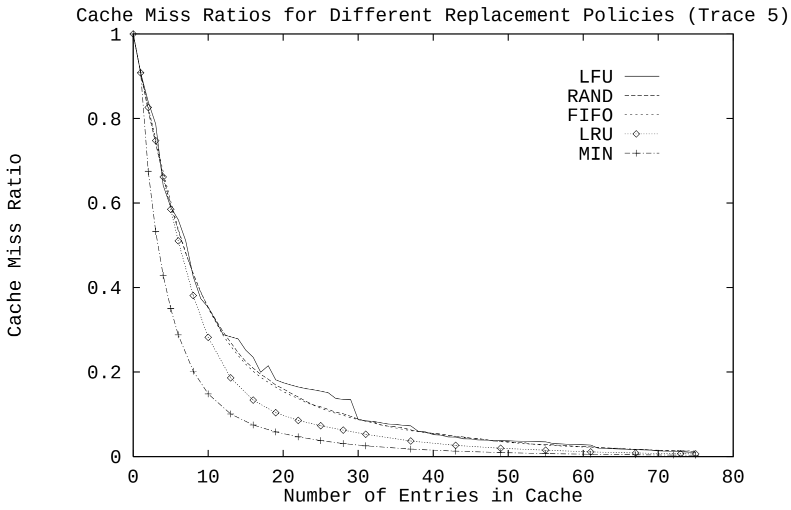

Figure 10: Cache Miss Ratios for LFU, FIFO, RAND, LRU and MIN (Trace 5)

As expected, MIN outperforms all other strategies for all cache sizes, but since it looks ahead in the reference string, it cannot be implemented in a real system. Results for MIN nevertheless show that there is room for improvement in implementable policies. Although LRU is the best among the implementable policies considered, it is still sub-optimal. Here, for example, at a cache size of 20 entries, LRU has miss ratios that are 65% higher than those of the optimal strategy (0.096 versus 0.058).

The second question is what fetch policy to use for the cache (i.e., what information to put in the cache, when). The three fetch policies (SF, DF and SDF) were simulated with LRU as the replacement policy. The results are shown for Trace 5 in Figure 11. Trace 6 and Trace 8 show very similar results.

For all three fetch policies, the miss ratios fall sharply up to a certain cache size (about 15 in this example), and then decrease slowly beyond this point. For SF, the miss ratios become almost constant beyond this cache size, while the other two policies continue to show a gradual decrease.

There are two distinct regions of interest in Figure 11. For cache sizes below 15, the three fetch policies perform similarly, although SF performs marginally better than SDF, and both perform better than DF. The better performance of SF and SDF over DF in this region can be attributed to the fact that source address re-use for small window sizes tends to be higher than destination address re-use, as discussed in Section 3.3. Hence fetching just the source address into the cache for each packet received would produce fewer misses than fetching just the destination address. That is, source addresses are good predictors of future destination address references in the short-term future.

For larger caches (more than 15 entries in this example), it is more important to cache destination addresses than source addresses. Caching only source addresses (the SF policy) results in virtually no improvement in miss ratio as cache size increases.

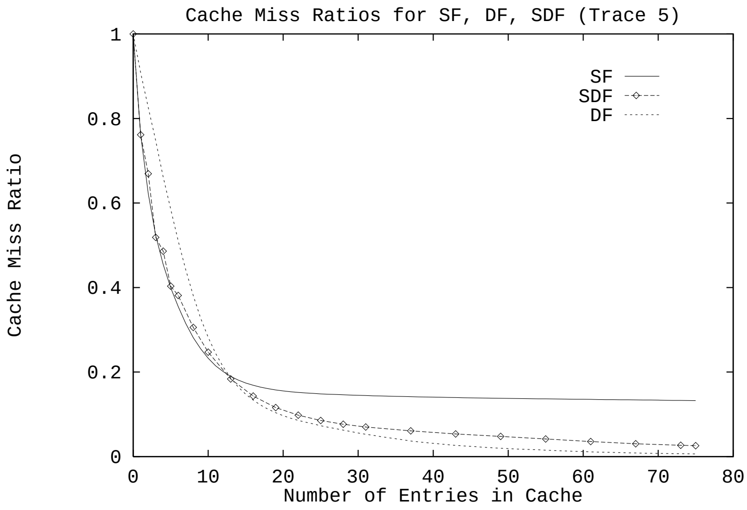

Figure 11: Cache Miss Ratios for SF, DF, SDF (Trace 5)

One reason for this behaviour is the fact that there are many source hosts that send very few packets. In Trace 5, for example, there are 107 hosts that send exactly one packet during the trace, and 200 hosts that send fewer than five packets. There are also some hosts that appear as destinations in the trace, but never as sources. Since SF does not fetch destination addresses into the cache, no entry for these destination hosts is ever fetched into the SF cache, and each packet going to any of these hosts results in a cache miss. DF and SDF do not face this problem.

The SDF policy provides reasonable (but not optimal) performance over the full range of cache sizes. Caching both source and destination addresses (the SDF policy) is an improvement over SF for large cache sizes, but it still does not perform as well as DF. Caching source addresses as well as destination addresses always hurts the performance for larger caches since all "predicted" destinations occupy cache slots, and some "predicted" destinations are never used. This performance degradation is due in part to lower source address concentration. In other words, source addresses are poor predictors for destination address references in the long-term future. Thus an interconnection device can benefit from address caching, but should use *both* source and destination addresses in its cache, with reasonably short timeouts on idle cache entries to keep unwanted addresses from cluttering the cache.

5 Conclusions

Packet traffic in packet-switched networks exhibits a phenomenon called network traffic locality, which is similar to the locality phenomenon observed in memory referencing and file referencing behaviour. Two dominant characteristics are present in network traffic locality. First, successive packets on the network tend to be highly related. Second, a small number of hosts tend to account for a large number of the packets transmitted on the network. These characteristics are evident as both short-term and long-term phenomena.

This paper presents four new measures that capture the salient aspects of network traffic locality. These measures are persistence, address re-use, concentration, and reference density. Packet trace data collected in a local area network environment was used to demonstrate and quantify each measure. Little persistence was found in the aggregate network traffic, but address re-use, concentration, and reference density are high, indicating significant temporal locality in packet traffic. Source addresses are good predictors of short-term future references in packet traffic, but poor predictors of long-term references.

The presence of network traffic locality has a number of potential applications, one of which has been studied in this paper. By adding a forwarding table cache to a network interconnection device, it may be possible to reduce the number of forwarding table lookups, thereby reducing the forwarding delay of the device. High concentration values mean that nominally sized caches (e.g., 20 entries) at such devices can give high hit ratios (e.g., 90%). Although this is only one example, it does suggest that the possibilities of exploiting network traffic locality should be investigated further.

References

[1] L. Belady, "A Study of Replacement Algorithms for a Virtual-Storage Computer", *IBM Systems Journal*, Vol. 5, No. 2, 1966, pp. 78-101.

[2] L. Breslau, D. Estrin, and L. Zhang, "Exploiting Locality to Provide Adaptive Routing of Real-Time Flows in Global Internets", Proceedings of the 4th IEEE COMSOC International Workshop on Multimedia Communications, April 1992.

[3] R. Bunt and J. Murphy, "The Measurement of Locality and the Behaviour of Programs", *Computer Journal*, Vol. 27, No. 3, 1984, pp. 238-245.

[4] P. Denning, "Working Set Model for Program Behavior", *Communications of the ACM*, Vol. 11, No. 5, May 1968, pp. 323-333.

[5] P. Denning, "Virtual Memories", *Computing Surveys*, Vol. 2, No. 3, September 1970, pp. 153-189.

[6] D. Feldmeier, "Improving Gateway Performance with a Routing-Table Cache", *Proceedings of IEEE INFOCOM '88*, New Orleans, March 1988, pp. 298-307.

[7] N. Gulati, "Local Area Network Traffic Locality: Characteristics and Application", M.Sc. Thesis, Department of Computational Science, University of Saskatchewan, 1992.

[8] R. Jain and S. Routhier, "Packet Trains – Measurement and a New Model for Computer Network Traffic", *IEEE Journal on Selected Areas in Communications*, Vol. SAC-4, No. 6, September 1986, pp. 986-995.

[9] R. Jain, "Characteristics of Destination Address Locality in Computer Networks: A Comparison of Caching Schemes", DEC-TR-592, February 1989.

[10] S. Ling, "Characterizing and Modelling the Workload of Ethernet Local Area Networks", M.Sc. Thesis, Department of Computational Science, University of Saskatchewan, 1992.

[11] M. Lorence and M. Satyanarayanan, "IPwatch: A Tool For Monitoring Network Locality", *Operating Systems Review*, Vol. 24, No. 1, January 1990, pp. 58-80.

[12] S. Majumdar, "Locality and File Referencing Behaviour: Principles and Applications", M.Sc. Thesis, Department of Computational Science, University of Saskatchewan, 1984.

[13] D. Mills, C. Boncelet, J. Elias, P. Schragger, and A. Jackson, "Highball: A High Speed, Reserved-Access Wide Area Network", Technical Report 90-9-3, Department of Electrical Engineering, University of Delaware, September 1990.

[14] J. Mogul, "Network Locality at the Scale of Processes", *Proceedings of ACM SIGCOMM '91*, Zürich, Switzerland, September 1991, pp. 273-284.

[15] C. Song and L. Landweber, "Optimizing Bulk Data Transfer Performance: A Packet Train Approach", *Proceedings of ACM SIGCOMM '88*, Stanford, California, August 1988, pp. 134-145.