X Windows Network Traffic Analysis

Mark P. Biegler and Howard J. Hamilton

Technical Report CS-97-06
Dec., 1997

Department of Computer Science
University of Regina
Regina, Saskatchewan, CANADA
S4S 0A2

ISSN 0828-3494
ISBN 0-7731-0355-4

X Windows Network Traffic Analysis

Mark P. Biegler
Howard J. Hamilton
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada

Abstract

This paper quantifies the performance of X applications on an Ethernet network. We replicated experiments performed by Basil and Williamson [2] in a single network segment to determine if the results would be similar in a different computing environment. Then we extended the work by running similar tests across more than one network segment.

In general, results were similar, although network load and data throughput was consistently higher by a small factor. One particular application, `twm`, displayed very anomalous results, possibly attributable to a version difference in the software. Tests across multiple LAN segments showed no significant differences from single-segment tests.

Since X applications are in common use in many UNIX LAN environments, the demands they make on the network are worthy of empirical study. Applications vary in the type and size of their demands on the network. Based on our results, the number of workstations running X applications on a network segment needs to be limited. We observed that traffic between network segments is not significantly more costly than that within a segment, which suggests that workstations should be partitioned into relatively small segments according to logical patterns of use; any traffic outside the segment will not be significantly slowed down.

1 Introduction

This paper reports empirical results concerning network traffic due to X Windows applications. This work directly builds on experiments performed by Basil and Williamson [2]. We first replicated their experiments to determine if their results are dependent on their computing environment. Then we extended the experiments from a single Ethernet network segment to multiple segments of a local area network (LAN). Early work on X Window system performance was reported in [3].

X Windows applications (or *X applications*) are computer programs running under the X Windows System [1]. Seven common X applications were studied: `twm`, `gnuplot`, `xdvi`, `xedit`, `xfig`, `xterm`, and `xv`. Although an eighth application, `videopix`, was studied in [2], it was not studied here because the software was unavailable. Three runs of tests, each 100 seconds in durations, were made for each application to obtain the results. As in [2], the `tcpdump` network traffic monitoring tool was used to collect the network packets generated by the X clients and servers. All statistics reported were obtained from the `tcpdump` traces.

In general, empirical results about the network performance of the X applications should facilitate network planning and design because of the increasing use of X terminals and the X Windows system. Our results from the replication experiment were generally similar to those of [2], but our network utilization and data throughput were consistently higher by a small factor, which we attributed to better workstation I/O and CPU speeds, rather than the X protocols.

Performance across multiple LAN segments were on the whole consistent with the single-segment results from our replication experiment and from [2].

Some differences may be attributed to the bridge having to forward packets from many sources and thus splitting larger packets into smaller packets in some cases.

The next section explains the methodology used in the experiments, including the experimental design and statistics collection for both the single LAN segment and the multiple LAN segment experiments. Results for both experiments are presented in Section 3 and analyzed in Section 4. Finally, conclusions and suggestions for future research are given in Section 5.

2 Measurement Methodology

In this section, the expermental design for the replication experiment is presented. This experiment duplicated work reported in [2] in a different environment. Then the experimental design for the multiple LAN segment experiment is described. Finally, an explanation of statistics collection is provided.

2.1 Design for the Replication Experiment

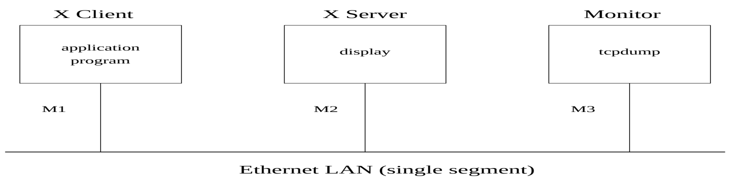

Figure 1: Single-Segment Ethernet LAN

The experimental setup for the replication experiment, which is shown in Figure 1, was designed to closely match the setup described in [2]. The network used is a 10 Mbits/second (Mb/s) Ethernet LAN. The network segment used for the replication experiment has 15 DECStation workstations, of which three (called

M1, *M2*, and *M3*) were used for testing. All workstations were running ULTRIX and X Windows X11 Revision 5. During the tests, there was light usage of the other machines, with perhaps only two other workstations in use at the same time. The other usage was relatively insignificant to the experiments because (1) only successfully transmitted packets were counted in our statistics and (2) increases to packet transmission times would be infrequent and small on a lightly loaded Ethernet segment.

To run `tcpdump`, workstation *M3* required a kernel recompilation to support the ULTRIX *packet filter* software, which allows a promiscuous mode of operation. In *promiscuous mode*, the workstation accepts **all** network packets whenever `tcpdump` is in use, rather than only those packets destined for the workstation.

To collect the raw packet headers of the X windows packets traveling between two hosts, the following command is issued on the monitoring station, *M3*:

```
tcpdump -w tcpfile tcp and host M1 and host M2
```

This command ensures that only TCP packets[1] traveling between the hosts *M1* and *M2* are collected. The command also writes the *packet trace* (a sequence of raw packet headers) to the file called `tcpfile`. If `tcpdump` were run without the host qualifiers, it would collect all packets on the network. Because the `tcpdump` command was not issued until monitoring was to begin, and because the experimenter was the sole user of workstations *M1* and *M2* during the time of monitoring, only X packets generated by the client and server machines were captured during the trace interval.

[1]X uses TCP/IP rather than UDP/IP because TCP offers a reliable end-to-end connection, even though a little more overhead is incurred.

To convert the raw packet headers produced by `tcpdump` into useful information, the `tcpfile` file is processed to generate summary statistics for each packet. The following command generates a text file called `logfile` for the packet trace collected above:

```
tcpdump -r tcpfile > logfile
```

Each line in the log file represents packet information from a single packet. Various statistics are obtained by parsing the log file, as described in Section 2.3.

2.2 Experimental Design for Multiple LAN Segments

The second portion of this research involved testing the X applications across multiple LAN segments. The network used for these tests is shown in Figure 2.

In this setup, the X Server, *M4*, was located on a network segment geographically separated from the segment containing *M1* and *M3*. The two network segments were interconnected via a high-speed switched Ethernet bridge, but all link speeds were 10 Mb/s. Packets destined for hosts on a different segment from the originating host must travel through this bridge.

2.3 Statistics Collection

The performance metrics used for analysis are exactly those used in [2]: bidirectional packet transfer rate, unidirectional packet transfer rate, bidirectional data transfer rate, unidirectional data transfer rate, percent network utilization, packet size distribution, and packet interarrival times. The transfer rates measure the number of packets or bytes per second in one direction (from client to server or from server to client) or both directions. The network utilization

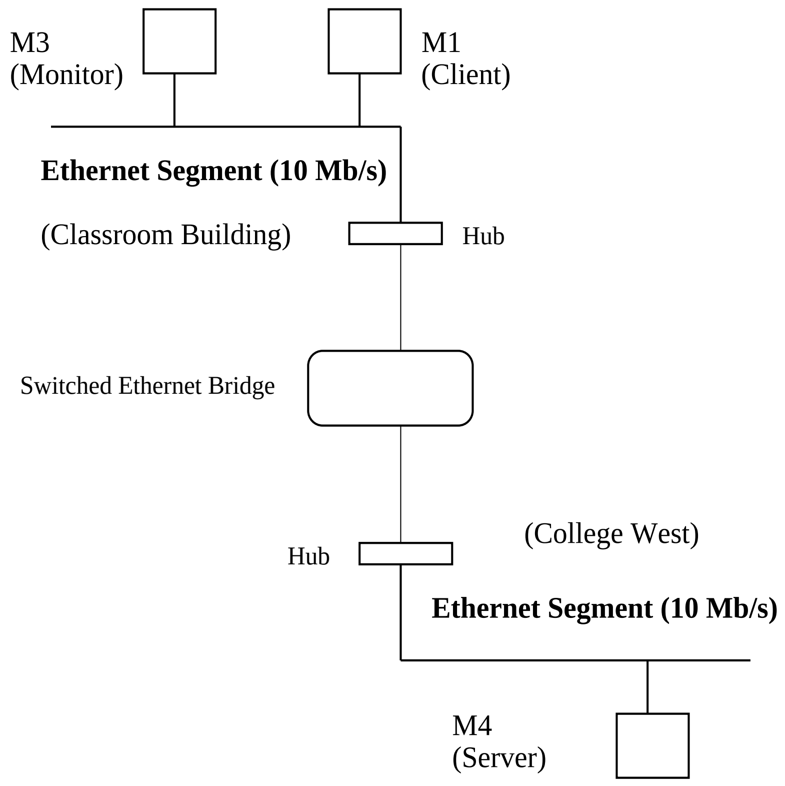

Figure 2: Multiple-Segment Ethernet LAN

percentage is obtained by calculating the ratio of the total number of bits transmitted per second to the maximum number of bits that could be sent over the network per second. Packet size distributions are examined to determine what typical traffic for each application might look like. The interarrival times provide an indication of how quickly successive packets are sent by each application.

Script files were created to process the `tcpdump` log files, and generate the required statistics. These were written in a mixture of `csh`, `awk`, and `C`.

3 Results

3.1 Replication Experiment

The results of the single-segment network tests are summarized in Tables 1 to 4 in Appendix A. These results are in the same format as those in [2], which are also provided for comparison. We will concentrate on reporting the major differences in our results when compared to those in [2].

The results are reasonably close for most of the performance metrics tested, but significant differences do occur in the `twm` and `xv` application results. In general, all applications yielded higher data throughput and network utilization figures than the respective results in [2]. As well, all applications except for `gnuplot` and `xdvi` had lower packet interarrival times; `gnuplot` had significantly higher interarrival times. The uniformly higher throughput (between 1.1 to about 1.8 times higher for most of the applications) can be attributed to better system architecture of the workstations used.

The `twm` results differ significantly from those obtained in [2]. The `twm` benchmark involved rapidly iconifying and deiconifying a fully-zoomed `twm` window filled with text. While the bidirectional traffic rate in [2] in bytes/sec was fairly close (5.4 K versus the 6.1 K obtained here), the number of packets per second rose from 12.8 to 98.7. Network utilization stayed under 1.0%. The unidirectional traffic results are quite different: 96.9% of the bytes transferred in results in [2] went from the client to the server; only 40.9% of the bytes did likewise in our results. The packet size distributions are also quite different. Basil and Williamson's distribution is clearly trimodal, while our distribution is unimodal, with over 90% of the packets being less than 200 bytes in size.

These differences suggest a version difference in the actual `twm` application itself. There are many more packets being transferred, and each packet is (on average) smaller. The 40/60 ratio of bytes transferred (as opposed to 97/3) could be attributable to more acknowledgement frames being transferred from the X server to the client, and no screen data being transferred; transferring the screen data for each window update would result in much more data being sent from the client to the server and thus more packets.

A tool called `xmon`, which passively logs X protocol messages transferred between hosts, was used to determine if the screen text was being transferred during the deiconification process. `xmon` revealed that the screen contents are not actually transferred across the network during this test. Possibly the version of `twm` used in the Basil and Williamson's environment did repeatedly send the screen contents across the network. If so, this would account for the larger packets obtained in their results.

Differences in the `xv` application were not nearly as dramatic. The main differences are that the bytes/second rose from 82.1 K to 226.1 K while network utilization went from 7.3% to 18.9%. Network utilization peaked at 70%, which is much higher than the 30% peaks reported in [2]. Furthermore, there was a peak of about 800,000 bytes per second in throughput, compared with 300,000 bytes per second in [2]. The increased throughput is possibly a function of a better combination of network interface card and internal data bus: the more quickly the data can get to the interface card from memory, the faster it can be put on the network. The less dramatic increase in number of packets per second reveals that the X protocols are effective in buffering data before sending a complete packet, thereby increasing network efficiency. There was also a higher percentage of large packets relative to those in [2].

Another difference was that the `xterm` packet size distribution in our experiment more closely approximates a trimodal distribution than the bimodal distribution reported in [2]. There was also a higher percentage of medium sized packets than in the results in [2].

Another interesting difference is the increase in interarrival times for `gnuplot`. 90% of the packets arrived in under 82 ms, while in [2], 90% arrived in under 26 ms, about three times more quickly. This is difficult to explain, and could be a function of the test equations used, and perhaps a slightly different version of the `gnuplot` software. The difference could also be attributable to the speed at which the graphs were calculated, and not necessarily the ability to send the data. The data set in [2] would have to be used and timed to determine if this is, in fact, the case. This could then account for the slight difference in interarrival times for `xdvi`, as well.

3.2 Multiple LAN Segment Results

For the multiple LAN segment tests, bidirectional and unidirectional traffic figures appear in tables in Appendix B.

Although most results are similar to those obtained for the single-segment tests, the results for `xterm` are noticeably different. The number of packets, throughput, and network utilization all increased dramatically – almost by a factor of 2. At the same time, the packet distribution went from a trimodal distribution to unimodal. This is interesting, since the unidirectional packet and data percentages did not change very dramatically. (That is, had there suddenly been a much higher percentage of packets transferred relative to the number of bytes, one could say that the previously large packets were simply split into many small packets, thus accounting for a change in distribution.) As well, packet interarrival times went from 92ms to 39ms.

Other minor differences include a doubling of `xdvi` throughput and network utilization. As well, the `xedit` utilization decreased from previously obtained results, more closely aligning to the results in [2]. The packet distribution also appears to be bimodal in nature rather than trimodal.

4 Conclusions and Suggestions for Further Research

Generally, the results obtained parallel those obtained by Basil and Williamson [2]. The average packet transfer and data transfer rates for most of the tested X applications are not very high. The notable exception was `xv`, with `xdvi` attaining a high network utilization in the multiple LAN testing. More data and packets are transferred from the X client to the server, except for the interactive applications, `twm` and `xfig`, where the reverse occurred.

Our results for the `twm` application differed from those in [2]. The ratio of bytes transferred between client and server was the key difference. The `twm` version in [2] should be tested with `xmon` to determine where the difference in results stems from.

The high load which `xv` is able to place on a network (just under 20% average utilization) implies that a few workstations running `xv` could seriously hamper the network response times for others. Only a few workstations designed for intensive graphics use should be placed on a single network segment to isolate such traffic. In general, the number of workstations running the X Window System on the same network segment should also be limited, even if they are only run `xedit` or `xterm` sessions. Ethernet physically permits a maximum of

1000 stations,[2] but running X applications would restrict the network to a small fraction of this number.

The multiple LAN tests suggested that X applications generate similar traffic across multiple network segments as on a single segment. No major differences were observed that could be attributed to the extra network segment. Our results may have been affected by unknown network traffic conditions over the selected segments. Further testing should be performed to examine the results of extra network segments on X applications in a more controlled environment.

As mentioned, according to our results, the number of workstations running X applications on a network segment needs to be limited. We observed that traffic between network segments is not significantly more costly than that within a segment, which suggests that workstations should be partitioned into relatively small segments according to logical patterns of use; any traffic outside the segment will not be significantly slowed down.

Further research should also be directed to testing each X application while the network was already being used with a certain pre-determined load. For instance, on a network of 15 workstations, one could automate a process on 12 of them to simulate the normal use of applications such as `twm` and `xterm` sessions. Once this load has been determined, each of the above applications could be reanalyzed under this new load to determine what the effect was. Conversely, the effect on each of the other systems could be monitored while the X applications were tested to see how they perform when new loads are added to the network.

[2]The maximum size of an Ethernet LAN consists of 2500 meters of cable (five 500 meter segments connected with 4 repeaters), and allows one station at every 2.5 meters.

A Numerical Results

The following two tables illustrate the bidirectional traffic characteristics of the seven X applications tested. Basil and Williamson's (B&W) results are presented first.

Table 1: Bidirectional traffic characteristics - B&W

X Window application	Packets per sec	Bytes per sec	Utilization per sec	Packet sizes	Interarrival times
twm	12.8	5.4 K	0.5%	Trimodal	90% $\leq$ 400 ms
xfig	88.3	6.2 K	1.1%	Unimodal	90% $\leq$ 33 ms
xterm	19.8	12.4 K	1.1%	Bimodal	90% $\leq$ 170 ms
xedit	24.6	12.9 K	1.2%	Trimodal	90% $\leq$ 130 ms
gnuplot	42.9	28.8 K	2.6%	Trimodal	90% $\leq$ 26 ms
xdvi	79.8	53.1 K	4.8%	Trimodal	90% $\leq$ 7 ms
xv	170.1	82.1 K	7.3%	Trimodal	95% $\leq$ 4 ms

Table 2: Bidirectional traffic characteristics

X Window application	Packets per sec	Bytes per sec	Utilization per sec	Packet sizes	Interarrival times
twm	98.7	6.1 K	0.9%	Unimodal	90% $\leq$ 36 ms
xfig	186.0	17.9 K	2.2%	Unimodal	90% $\leq$ 7 ms
xterm	32.6	21.2 K	1.8%	Trimodal	90% $\leq$ 92 ms
xedit	54.9	23.1 K	2.06%	Trimodal	90% $\leq$ 49 ms
gnuplot	47.1	34.9 K	2.96%	Trimodal	90% $\leq$ 82 ms
xdvi	118.6	68.7 K	5.95%	Trimodal	90% $\leq$ 15 ms
xv	216.4	226.1 K	18.9%	Trimodal	95% $\leq$ 3 ms

The following two tables present the unidirectional traffic results for the X applications. Again, B&W's results from [2] are presented first.

Table 3: Unidirectional packet and data transfer - B&W

X Window	Packets transferred		Bytes transferred	
application	ClientToServer	ServerToClient	ClientToServer	ServerToClient
twm	55.3%	44.7%	96.9%	3.1%
xfig	42.3%	57.7%	56.2%	43.8%
xterm	54.5%	45.5%	98.9%	1.1%
xedit	51.2%	48.8%	97.5%	2.5%
gnuplot	66.1%	33.9%	99.8%	0.2%
xdvi	76.8%	23.2%	99.7%	0.3%
xv	60.5%	39.5%	98.6%	1.4%

Table 4: Unidirectional packet and data transfer

X Window	Packets transferred		Bytes transferred	
application	ClientToServer	ServerToClient	ClientToServer	ServerToClient
twm	61.8%	38.2%	40.9%	59.1%
xfig	41.5%	58.5%	73.1%	26.9%
xterm	71.3%	28.7%	99.4%	0.6%
xedit	60.5%	39.5%	96.6%	3.4%
gnuplot	78.5%	21.5%	100.0%	0.0%
xdvi	71.2%	28.8%	98.3%	1.7%
xv	80.5%	19.5%	99.9%	0.1%

B Multiple LAN Results

The following two tables illustrate the bidirectional and unidirectional traffic characteristics of the X applications when run across multiple network segments.

Table 5: Bidirectional traffic characteristics - Multiple LAN

X Window application	Packets per sec	Bytes per sec	Utilization per sec	Packet sizes	Interarrival times
twm	101.4	5.8 K	0.9%	Unimodal	90% $\leq$ 23 ms
xfig	165.9	19.1 K	2.1%	Unimodal	90% $\leq$ 11 ms
xterm	70.9	42.8 K	3.7%	Bimodal	90% $\leq$ 39 ms
xedit	40.3	10.5 K	1.0%	Unimodal	90% $\leq$ 62 ms
gnuplot	49.0	34.6 K	3.0%	Trimodal	90% $\leq$ 82 ms
xdvi	197.3	133.2 K	11.5%	Trimodal	90% $\leq$ 15 ms
xv	276.4	254.2 K	21.4%	Unimodal	95% $\leq$ 3 ms

Table 6: Unidirectional packet and data transfer - Multiple LAN

X Window	Packets transferred		Bytes transferred	
application	ClientToServer	ServerToClient	ClientToServer	ServerToClient
twm	57.6%	42.4%	48.4%	51.6%
xfig	42.0%	58.0%	74.9%	25.1%
xterm	67.5%	32.5%	99.2%	0.8%
xedit	46.7%	53.3%	91.2%	8.8%
gnuplot	77.7%	22.3%	100.0%	0.0%
xdvi	73.5%	26.5%	99.2%	0.8%
xv	71.0%	29.0%	99.9%	0.1%

References

[1] Barkakati, Nabajyoti. *X Window System Programming.* Prentice Hall, 1992.

[2] Basil, Nipun and Carey L. Williamson. *Network Traffic Measurement of the X Window System.* IEEE 13th Annual International Phoenix Conference on Computers and Communications (IPCCC), Phoenix, AZ, pp. 148-155, April 1994.

[3] Droms, Ralph and Wayne R. Dyksen. *Performance Measurements of the X Window System Communication Protocol.* Software: Practice and Experience, Vol 20, October 1990, pp 119-136.