NetDyn Revisited: A Replicated Study of Network Dynamics *

Julie Pointek, Forrest Shull, Roseanne Tesoriero and Ashok Agrawala

Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland
College Park, MD 20742

October 1, 1996

Abstract

In 1992 and 1993, a series of experiments using the NetDyn tool was run at the University of Maryland to characterize network behavior. These studies identified multiple design and implementation faults in the Internet. Since that time, there has been a wide array of changes to the Internet. During the Spring of 1996, we conducted a replication of the NetDyn experiments in order to characterize end-to-end behavior in the current environment. In this paper, we present and discuss the latest results obtained during this study. Although the network seems to be stabilizing with respect to transit times, our current results are similar to the results from past experiments. That is, networks often exhibit unexpected behavior. The data suggest that while there has been improvement, there are still problem areas that need to be addressed.

*This work is supported in part by ONR and ARPA under contract N66001-95-C-8619 to the Computer Science Department at the University of Maryland. The views, opinions, and/or findings contained in this report are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, ONR or the U.S. Government.

1 Introduction

In 1992 and 1993, a series of experiments using the NetDyn tool was run at the University of Maryland to characterize network behavior [12, 13]. These studies reported four major performance measures, which for the most part are not included in the standard operational measures: transit time of packets, number of lost packets on the link, number of duplicate packets on the link, and number of out-of-sequence packets on the link. The NetDyn tool measures characteristics of network behavior on a per-packet basis for end-to-end paths (reflecting the impact that will be felt at the user level). NetDyn does not rely on reliable transport protocols, such as TCP, which are designed to hide errors at the network layers. Experiments using the NetDyn tool have identified multiple design and implementation faults in the Internet.

The past few years have seen a wide array of changes to the Internet. Among others, there have been changes in topology (such as the migration of most U.S. traffic from NSFNET to interconnected network providers [14]), changes to hardware (such as the retirement of the T1 network [14]), and changes in user patterns (studies have found that the number of hosts on the Internet has increased more than 7 times between 1993 and 1996 [7, 8]; the number of websites in the same period has risen from almost none to over 100,000 [6]).

At the same time, as the Internet has penetrated into mainstream culture there has been a rise in efforts aimed at developing new Internet applications (e.g. electronic fund transfers), which bring with them a whole host of security and quality of service concerns, and which will all place differing load requirements on the networks.

In such an environment it becomes crucial to characterize network behavior, both in order to identify problem areas and to validate assumptions about expected behavior in order to develop and test new network protocols. We feel a characterization of end-to-end behavior in the current environment would be of great benefit. To perform such a study we have undertaken a series of experiments aimed at replicating the previous NetDyn experiments in the current environment.

Other studies of network behavior have appeared in the literature [11, 5, 4, 9]. In [11], the study focuses on building analytical models for network experiments. While we use some of the same analysis techniques, our focus is on characterizing network traffic. The problem of inadvertent synchronization is discussed in [5], which observed network behavior using ping. Ping uses ICMP packets which are treated differently than user level packets by some gateways, and so would be unsuitable for a study of user-level behavior. The experiments discussed in [9] rely on kernel modifications to observe network behavior. In [4], user level performance was investigated. In this study, all of the hosts for the experiment were in one location (UC-Berkeley). Our experiments differ from these studies in that we wanted to observe user level performance of end-to-end paths over a wide-area with minimal interference.

2 Experimental Setup

We used the NetDyn tool [12, 13] to send probe packets to specified hosts throughout the network. Rather than using the traditional throughput measure to characterize end-to-end network path behavior, we were concerned with round-trip time (RTT), losses, and reorders, as well as observations of anomalies in expected network behaviors.

NetDyn consists of four programs: *Source, Echo, Sink* and *Logger*. Each program is run as an application process. Source and Sink are generally run on the same local host. Note, however, that though we conducted our experiments in such a way that the Source and Sink were always the same host (i.e. packets were always sent on a round trip from the Source) it may still be useful to think of them as conceptually different entities, since each is responsible for different activities. Echo is located on the remote host. The Logger process may run on any host, but,for convenience, is on the local host for these experiments.

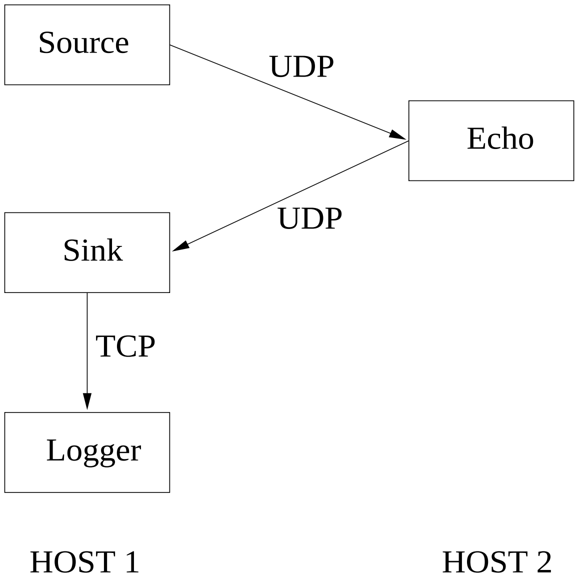

Figure 1: Experimental Setup.

Source generates packets and sends them to Echo. The Echo process then forwards packets to the Sink process. Sink creates a log record for each received packet and passes the record onto Logger. The Logger process finally writes the packet information to a log file on disk. Data stored in the log can then be used to compute round-trip times, losses, reorders, and duplicate packets. Figure 1 shows the setup of the NetDyn tool.

The Source, Echo, and Sink processes all place a timestamp in the packet. Additionally, Source and Echo assign sequence numbers to the packets. Each packet is big enough to only hold these three timestamps and two sequence numbers. This minimal packet size is used so that the experiments will not contribute significantly to network congestion, which might skew the results.

Both the Source and Echo processes forward packets using the User Datagram Protocol (UDP) [1]. [1] UDP is a best-effort protocol which does not provide flow control or reliability checks such as detection of lost packets, in-order delivery, etc. The absence of quality of service guarantees allows us to study network behavior at the IP [2] level. Also, UDP allows us to do our IP performance evaluation from the user level without any kernel modifications. Note that we could not use TCP [3] for this purpose since losses, reorders, etc. are not visible to application processes. TCP is, however, used to transfer log information between the Sink and Logger processes so as

[1] For a more detailed discussion of why UDP was chosen over alternate methods of performance appraisal such as IP_LSRR and ping, see [12, 13].

No.	Date	Time (EDT)	Source/ Sink	Echo host	Losses			Reorders
					Total	Forward(%)	Reverse(%)	
1	4.15	0400	UMD	Texas	577	0.11	0.47	80
2	4.15	1600	UMD	Texas	21628	16	6.2	185
3	4.16	0400	UMD	UMass	4771	3.5	1.3	77
4	4.16	1600	UMD	UMass	12351	2.5	10.1	305
5	4.17	0400	UMD	Stanford	629	0.19	0.44	25
6	4.17	1600	UMD	Stanford	3712	0.46	3.3	121
7	4.22	0400	Stanford	Texas	445	0.33	0.12	46
8	4.22	1600	Stanford	Texas	9659	9.4	0.29	40
9	4.26	0400	Texas	UMass	2286	0.57	1.7	55
10	4.26	1600	Texas	UMass	22785	16	8.3	91
11	4.29	0400	Stanford	UMass	197	0.11	0.08	17
12	5.01	1600	Stanford	UMass	628	0.34	0.29	228
13	4.22	0400	UMD	Bari	37019	35	2.9	91
14	4.22	1600	UMD	Bari	6028	3.5	2.7	58

Table 1: Losses and Reorders

to assure uncorrupted recording of our results.

We augmented the data analysis of NetDyn results to include two additional measures. First, we implemented an autocorrelation function to detect any periodic patterns in network behavior. We also included a function to determine on which leg of the trip (forward or return path) packets were lost.

Packets were sent from Source every 40 ms. The same interpacket delay was used in the previous NetDyn experiments, and we maintain this parameter setting to serve as a basis for comparison. Also as in the earlier experiments, we sent 100,000 packets in each experiment. It takes the Source process a little over an hour to generate and send all of these packets. The duration of the experiment is considered sufficient to observe both short and longer term network behavior changes and patterns.

We executed two experiments per local-remote link. One experiment was conducted at 4:00 am EDT and the other at 4:00 pm EDT. We believe these two times adequately represent off-peak and peak times with respect to Internet load. To initially determine these times, we conducted two sets of 24 hour experiments to observe overall average RTT and loss rate changes over the course of a day. In these experiments, we sent 10,000 packets every hour on the hour from UMD to both a nearby host at Loyola College in Baltimore and a cross-country host at Stanford University.

3 Results

Table 1 reports losses and reorderings for each of the experiments. A packet is counted as a loss if it is never received at the Sink. By examining the sequence numbers assigned to a packet by the Source and Echo and looking for gaps, we can infer whether the loss occurred on the forward

No.	Date	Time (EDT)	Source/ Sink	Echo host	Roundtrip Times (ms) Min	Max	Avg	Std. Dev.	Count >500ms	>1000ms
1	4.15	0400	UMD	Texas	48	658	52.87	16.66	7	0
2	4.15	1600	UMD	Texas	52	707	107.76	33.33	17	0
3	4.16	0400	UMD	UMass	25	717	45.52	24.72	48	0
4	4.16	1600	UMD	UMass	25	844	67.85	40.83	31	0
5	4.17	0400	UMD	Stanford	79	801	86.97	18.78	16	0
6	4.17	1600	UMD	Stanford	78	1015	93.52	27.97	33	1
7	4.22	0400	Stanford	Texas	78	476	78.87	7.72	0	0
8	4.22	1600	Stanford	Texas	78	492	93.08	8.53	0	0
9	4.26	0400	Texas	UMass	87	501	102.89	13.52	1	0
10	4.26	1600	Texas	UMass	90	2189	152.08	31.24	31	2
11	4.29	0400	Stanford	UMass	86	340	89.61	8.16	0	0
12	5.01	1600	Stanford	UMass	86	637	103.48	20.26	15	0
13	4.22	0400	UMD	Bari	248	1717	421.52	222.97	9322	3799
14	4.22	1600	UMD	Bari	287	797	319.20	29.07	361	0

Table 2: Roundtrip times

(Source to Echo) or reverse (Echo to Sink) path. We report this information as the percentage of the packets sent on each path that never reached the appropriate destination.

The number of reorderings was calculated as the number of packets that arrived at the Source previous to the arrival of a packet with a smaller sequence number. This number represents the number of packets that would have to be buffered by the receiving process in a loss-free environment before delivery could be made to the application process. This calculation ignores duplicate copies of packets that may arrive out-of-order.

We also looked for duplicate packets, that is, packets for which multiple copies arrive at the Sink even though only one copy was sent from the Source. Unlike the 1992 and 1993 experiments, which found duplication rates of up to 1%, we found no instances of duplicate packets arriving at either the Echo or Sink. (We did, however, notice bursts of duplicate packets during our 24-hour polling experiment between UMD and Loyola.)

The loss rates are especially interesting. As might be expected, the highest number of losses occurred in the link across the Atlantic to Italy. The trend seems to indicate that losses for this link correlate more with the traffic patterns at the Italian end of the link than with the Source/Sink in the U.S.: losses were only 6% at the U.S. peak time but were 37% at U.S. off-peak time (corresponding to 10AM at the Italian Echo host).

Even leaving aside the trans-Atlantic link, however, loss rates were observed to be higher than expected, with peak times incurring higher loss rates than off-peak times. Peak loss rates are seen to range from 0.6% to almost 23%, while off-peak time rates ranged from 0.2% to 4.7%. Although the 23% loss rate was unexpectedly high, it does not appear to be a fluke, since three out of our six peak-time within-U.S. experiments had rates greater then 10%.

Less of a pattern can be observed in the data for reorders. While it is true that the number of reorders was about the same or somewhat higher at peak time than at off-peak, the difference

Date	Eastern Time	Losses	Duplicates	Reorders
4.17.96	0400	629	0	25
4.17.96	1600	3712	0	121
1.93	0015	1415	9	52
1.93	1345	650	17	17
1.93	1830	658	35	35
5.29.92	1530	4825	921	75
5.29.92	2130	2857	913	12
5.30.92	0330	2164	1056	4

Table 3: Losses, Duplicates, and Reorders from UMD to Stanford

was not extreme. Values ranged from 0.017% to 0.3%. Additionally, the number of reorders was uncorrelated with the number of losses for an experiment.

Finally, it can be noted that there is no one link which performed worst at both times of day, although the Stanford-UMass link always had the lowest loss rate. Since a second cross-country link to Stanford (UMD-Stanford) also had the second lowest loss rate for peak hours, we attribute the low cross-country loss rates to the higher-reliability backbone linking these areas.

Table 2 summarizes the performance of the links in each of the experiments in terms of the minimum, maximum, and average round trip times experienced by packets on that link, as well as the standard deviation. It can be observed that packets sent during peak times exhibited a higher average RTT than packets sent at off-peak times. There is also more variation in the RTTs of packets sent during peak time, as shown by the higher standard deviations. While the maximum RTT for a link can be seen to fluctuate between peak and off-peak experiments, the minimum RTT was always comparatively constant for each link, leading us to be confident that this value captures the lowest possible RTT permitted by the link in the absence of congestion. The number of packets taking longer than 500 ms to make the round trip is also seen to be a negligible percentage of packets in within-U.S. experiments.

The previous iterations of the NetDyn experiments share a common link with the current set of experiments. Each time, a link from the University of Maryland to Stanford University was included. Table 3 shows the differences in losses, duplicates and reorders over replications for this link. Table 4 presents the differences in round-trip times.

4 Observations and Discussion

The observations from the current experiment allow us to draw general conclusions about network behavior. Since the links which we examined in this series of experiments were selected to cover the same regions as in the previous studies, we discuss our results in the light of past results. We assume that our links are representative of behavior on the network as a whole, and so can give insight into general network behavior.

Date	Eastern Time	Roundtrip Times (ms)				Count	
		Min	Max	Avg	Std. Dev.	>500ms	>1000ms
4.17.96	0400	79	801	87	19	16	0
4.17.96	1600	78	1015	94	28	33	1
1.93	0015	74	441	85	15	0	0
1.93	1345	74	758	85	14	14	0
1.93	1830	74	758	84	23	86	0
5.29.92	1530	74	898	105	40	244	0
5.29.92	2130	78	706	87	17	40	0
5.30.92	0330	74	671	85	16	25	0

Table 4: Roundtrip times from UMD to Stanford

Network Stability

The previous NetDyn experiments [12, 13] provide an interesting point of comparison for our results because they collect the same performance measures using the same experimental setup. From this comparison, we see that there has been an improvement in network stability over the last few years. The variability in transit times in the current experiment appears to be smaller then the variability observed previously. The minimum RTTs are increasing while the maximum RTTs are decreasing. In the 1992 NetDyn experiment, the standard deviation for transit times ranged from 16ms to 118ms. In the 1993 NetDyn experiment, the range was from 14ms to 54ms. In the current experiment, the range was from 8ms to 40ms when the trans-Atlantic link is excluded. Although variability in transit times appears to be decreasing in general, individual links (such as the UMD/Stanford link in Table 4) may not demonstrate this trend.

Another indication of increased network stability is the near elimination of duplicates. In previous experiments, duplicates were present in nearly all runs of the experiments. In all of our runs, there were no duplicates observed. We should note however, that duplicates appeared in some of the UMD-Loyola runs during our 24-hour polling experiments to determine peak vs. off-peak times. Since we noticed this phenomenon in no other run of the experiment, we conclude that this behavior is peculiar to that link and not an Internet-wide problem.

Delays and Losses

Although the network appears to be more stable, excessively high round-trip times and significant packet losses are still being observed. In the current experiment, we saw three runs with packets having RTTs greater than 1 second. In one case, the packet RTT exceeded 2 seconds. Loss rates were as high as 37%. When considering only the runs within the continental U.S., losses still were as high as 23%.

We had assumed that the major cause of lost packets and excessively high RTTs is congestion on the network. However, we noticed during our experiments many instances of behavior that seem to contradict this assumption:

- One would expect that bursts of network traffic would persist long enough to interfere with multiple packets in a row. Although there were many instances of consecutive packet losses, most losses occurred one at a time.

- If losses were primarily caused by buffer overflows due to congestion, lost packets would be preceded by packets with high RTTs (as the buffers fill up). While we noticed many instances of such behavior in our experiments, this was not always the case. Figure 2 contains examples of losses occurring with no appreciable preceding increase in RTT.

- Similarly, losses caused by buffer overflows will be followed by higher RTTs which gradually taper off, as the buffers empty until a normal threshold is again reached. However, we observed several instances in which further lost packets were interspersed within this decline, as well as a few cases in which RTTs actually increased after a loss.

Furthermore, congestion alone can not explain large round-trip times of more than a second. If excessive cross traffic was the cause, gateways would have to queue up to one second's worth of cross traffic within 40 ms. Servers along the path could not, however, have enough memory for such long queues. Also, if congestion was the reason for large RTTs, as the buffers are emptying, we would expect to see a gradual return to the average transit time instead of the sharp drop as seen in Figure 2.

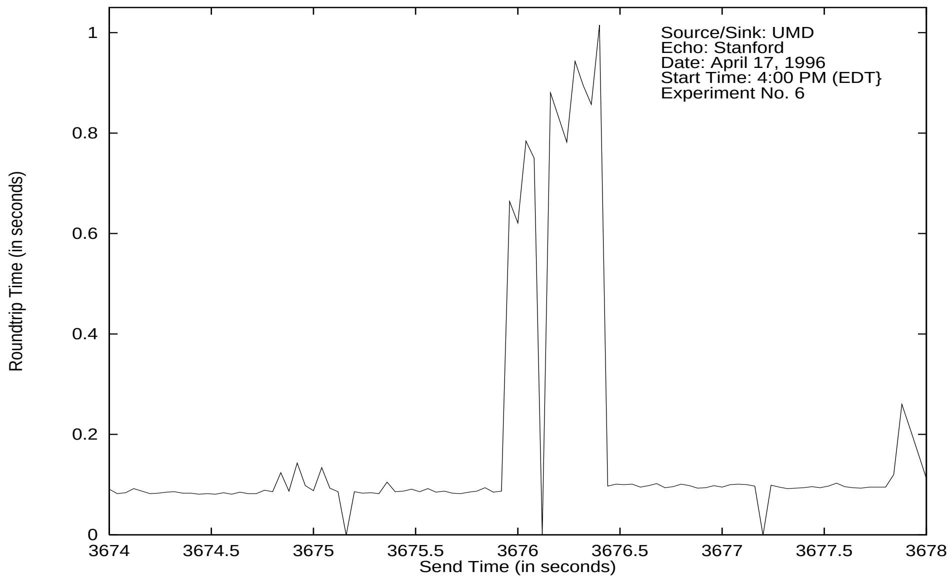

Figure 2: Unexpected RTT behavior. (Losses are indicated by RTTs of 0.)

For these reasons, we believe that factors other than buffer congestion significantly contribute to the dropping and delay of packets. Since today's networks usually support very low bit-error rates [10], bit errors due to the network must also be discarded as a significant explanation.

Several possible reasons for this unexpected behavior have been suggested in the literature [13, 5]:

- problems with interface cards in gateways
- router updates in non-work-conserving routers
- network administrators have been known to accidentally leave the debugging option on at gateways
- faulty memory management policy at gateways
- periodic packet drops by routers

Another possible explanation is that operating systems may have timing problems with asynchronous interactions. This explanation accounts for losses occurring between the network and application layers.

Reorders and Alternating Paths

We attempted to find explanations for the number of reorders that we observed. Our first thought was that reorders were caused by packets taking different paths through the network. If a router switched packets between two paths to compensate for congestion, we would expect to see the RTTs oscillating between higher and lower levels. Experiment 9 shown in Figure 3 exhibits such a step behavior. Note the sudden increase in RTT, followed by a leveling off at a lower RTT threshold. Packets fluctuating between two different routes from the source to the sink could explain this type of behavior.

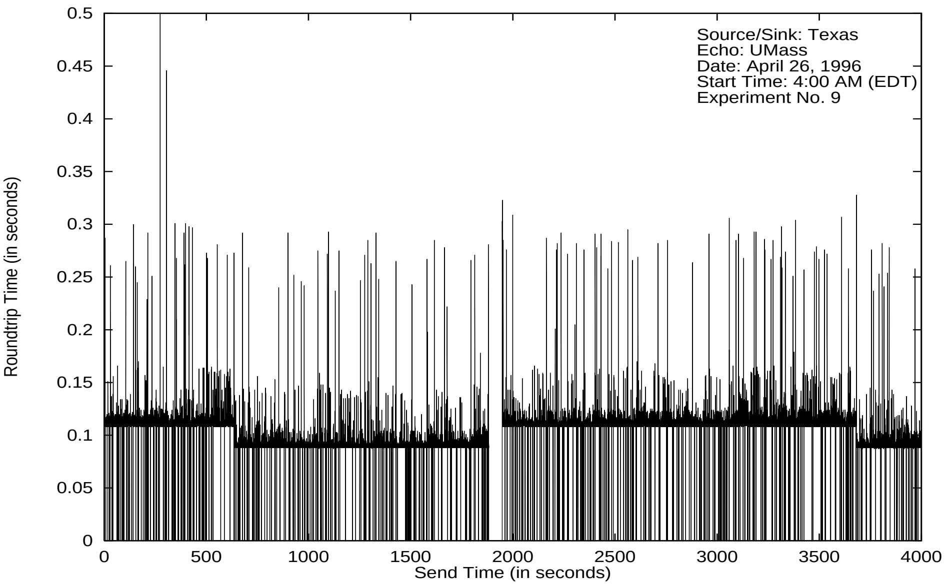

Figure 3: Step pattern behavior. (Losses are indicated by RTTs of 0.)

The observed step behavior does not correspond to the experiments with the highest numbers of reorders. Therefore we cannot accept alternating between two paths as the primary cause of packet reorders. However, if the routers were switching quickly between many paths, this could account for high reorders without noticeable step behavior.

Additionally, [13] mention that they had encountered cases in which small numbers of packets had reached the destination in exactly reverse order. From this it seemed as though the gateway software had implemented a stack rather than a queue for processing IP-optioned packets which could lead to observable reorders.

Periodicity

In our experiment, we observed periodic behavior along the Stanford-UMass link. There were periodic peaks in transit time at 60s intervals. Figure 4 shows the RTTs and Figure 5 shows the autocorrelation. The spike at 60 seconds shows that packets sent 60 seconds apart are highly correlated with respect to transit time. Similar studies [11] have used a threshold autocorrelation value of 0.1 to indicate a significant relationship.

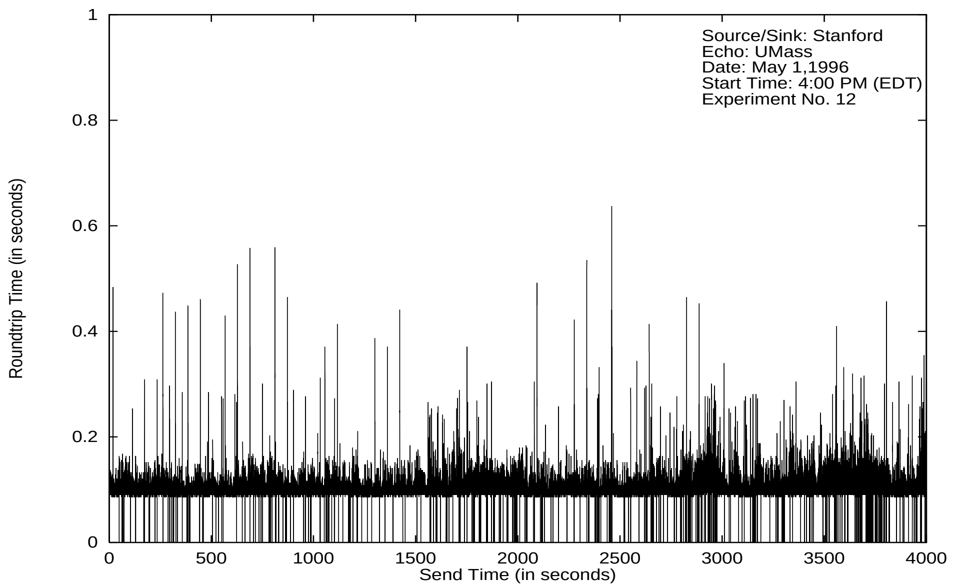

Figure 4: Periodic behavior: RTT vs. Send Time. (Losses are indicated by RTTs of 0.)

Interestingly, other studies have found periodic behavior on nearly identical links. In the 1992 NetDyn experiment [12], periodic peaks in transit time were found at intervals of 90 seconds along the Stanford-MIT link. [5] report periodic behavior between Berkeley and MIT and attribute it to problems with NEARnet core routers.

5 Conclusion

Use of the Internet has increased significantly over the past several years. The topology of the Internet has also changed from a single-backbone network to a multiple-backbone network. In this paper, we discuss the latest results in a series of experiments obtained using the NetDyn

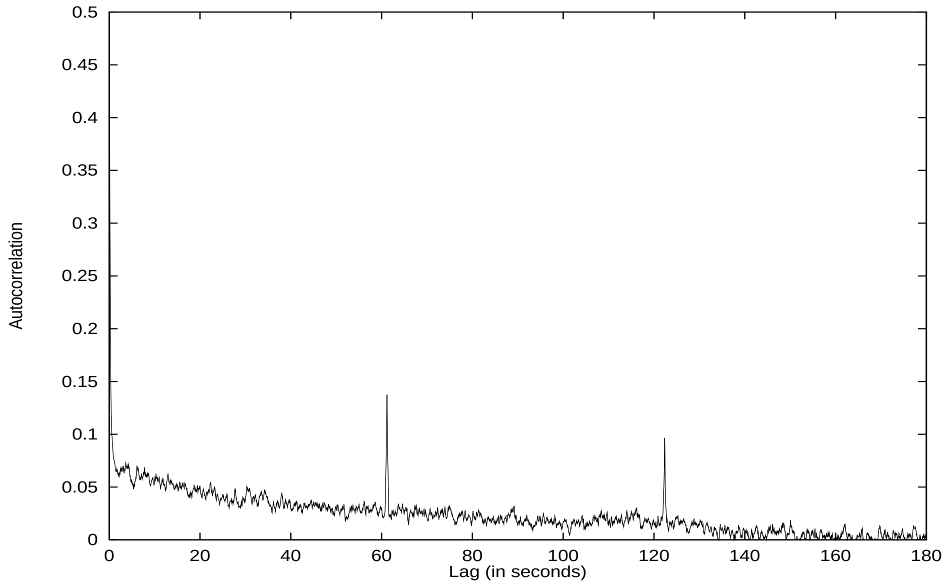

Figure 5: Periodic behavior: Autocorrelation

tool. We feel that such results are particularly interesting because few other studies examine user-level, end-to-end behavior. Using NetDyn has facilitated the examination of multiple links to many different regions allowing a picture of overall network behavior to be built. We were also able to compare our current results with previous experiments which also used NetDyn.

The increase in network volume and complexity does not appear to have affected transit times adversely. The variability in round-trip times has become smaller. This may indicate that the network is stabilizing. Another indication that the network may be more stable is the reduction in the number of duplicate packets observed.

Although the network seems to be stabilizing with respect to transit times, our current results are similar to the results from past experiments. That is, networks often exhibit unexpected behavior. We still see round-trip times that are excessively high. Periodic and step change behavior are still present in the network. Unexplained losses are still occurring. The data suggest that while there has been improvement over the past several years, there are still problems that need to be addressed when designing protocols and applications for networks.

References

[1] User Datagram Protocol, Request for Comment: RFC-768. Network Information Center, SRI International, August 1980.

[2] Internet Protocol, Request for Comment: RFC-791. Network Information Center, SRI International, September 1981.

[3] Transmission Control Protocol, Request for Comment: RFC-793. Network Information Center, SRI International, September 1981.

[4] L.F. Cabrera, E. Hunter, M. Karels, and D.A. Mosho. User-process communication performance in networks of computers. *IEEE/ACM Trans. on Software Eng.*, 14(1):38–53, 1988.

[5] Sally Floyd and Van Jacobson. The synchronization of periodic routing messages. *IEEE/ACM Trans. on Networking*, 2(2):122–136, April 1994.

[6] M. Gray. Measuring the growth of the Web: June 1993 to June 1995.
URL: http://www.mit.edu/people/mkgray/growth, Visited May 1996.

[7] M. Lottor. Internet growth (1981-1991), Request for Comment: RFC-1296. Network Information Center, SRI International, Jan 1992.

[8] M. Lottor. Number of internet hosts. URL: http://www.nw.com/zone/host-count-history, Visited May 1996.

[9] Christos Papadopoulos and Gurudatta M. Parulkar. Experimental evaluation of SUNOS IPC and TCP/IP protocol implementation. *IEEE/ACM Trans. on Networking*, 1(2):199–216, April 1993.

[10] Craig Partridge. *Gigabit Networking.* Addison-Wesley, Reading, Massachusetts, 1994.

[11] V. Paxson. Empirically derived analytic models of wide area TCP connections. Technical Report LBL-34086, Lawrence Berkeley Laboratory, May 1993.

[12] D. Sanghi, A. K. Agrawala, O. Gudmundsson, and B. Jain. Experimental assessment of end-to-end behavior on Internet. Technical Report CS-TR-2909, University of Maryland, Computer Science Department, College Park, Maryland, 1992.

[13] D. Sanghi, O. Gudmundsson, and A. K. Agrawala. Study of network dynamics. *Computer Networks and ISDN Systems*, 26(7):371–378, 1993.

[14] Robert Zakon. Hobbes' Internet timeline v2.4a.
URL: http://info.isoc.org/quest/zakon/Internet/History/HIT.html, Visited May 1996.