Measurement Considerations for Assessing Unidirectional Latencies*

Kimberly C. Claffy and George C. Polyzos
kc@cs.ucsd.edu, polyzos@cs.ucsd.edu
Computer Systems Laboratory
Dept. of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114

Hans-Werner Braun
hwb@sdsc.edu
San Diego Supercomputer Center
San Diego, CA 92186-9784

Abstract

This paper presents a study of single direction latencies to selected destinations of the Internet utilizing a variety of paths. The objective is to demonstrate that round-trip latencies are an insufficient and sometimes misleading method to determine unidirectional delays. This claim has significant implications for high-speed, multi-application, wide area, traffic aggregating networking environments which often require predictability of precise delay.

Keywords: measurement, delay, unidirectional, jitter, latency.

1 Introduction

In the Internet community a common method for assessing network latencies is to measure round trip delivery time, the time it takes for a packet to get to and return from a target host. Dividing this value in half to arrive at an outgoing or return latency implicitly assumes that the path to a target host is symmetric.

There are both static and dynamic components of transmission latencies which contribute to inaccuracies of this method. Statically, the route might be asymmetric. Dynamically, resource contention for network components causes congestion along the paths, and can change rapidly and asymmetrically.

We hypothesized that these factors introduce distortions into latency assessments, significant enough to justify a re-evaluation of the current measurement paradigm. This claim has significant implications for high-speed, multimedia networking environments. For example, digital continuous media data flows will require predictability of unidirectional delays. The assumption that this delay is independent of the flow direction is dubious at best in today's often congested Internet.

*This research is supported by a grant of the National Science Foundation (NCR-9119473), a joint study agreement with the International Business Machines, Inc., and an agreement with the Corporation for National Research Initiatives, with support from the National Science Foundation and the Defense Advanced Research Projects Agency, on *research into gigabit networks (CASA)*.

Network requirements of real-time continuous media applications radically differ from those of traditional, mostly text-based, applications. Psychological factors impose strict deadlines on the presentation of multimedia information, and low variance in delay is typically more critical than occasional packet loss. In order to meet such fundamentally different needs over internetworks without explicit performance guarantees, continuous media applications may use error concealment techniques to enable limited loss recovery of transmitted packets, and destination buffering to compensate for the jitter introduced by the network.

Since large internetworks can not generally offer absolute bounds on delay, applications must operate with a "statistical" delay bound, i.e., a certain percentage of packets will experience delay below the given threshold and the remaining will be ignored by the application. Ideally, the application will be able to continuously monitor the directional path delay through the network and adapt to it by controlling buffering and application parameters, such as video resolution or frame rate. Clark, Shenker, and Zhang [1] refer to such applications as *tolerant and adaptive*.

Even though we do not expect wide deployment of such applications on the current Internet, there have already been sizable experiments, notably the multicasting of audio and video from the July 1992 Internet Engineering Task Force (IETF) meeting. In any case, designers of real-time applications should be aware that, especially across the periphery of wide-area, heavily traffic aggregating networks, symmetric delay is a poor model of network performance. It is therefore important for such applications to obtain accurate statistics on unidirectional delay through the network.

This paper presents a study of the variances of single direction latencies to selected destinations. Our results indicate that roundtrip measurements are inadequate indicators of network behavior, and separate assessment of the two latency components is important for an accurate understanding of network-wide resource contention. Through the course of our study we not only verified our belief in the asymmetry of latencies, but also discovered several orthogonal issues in problematic protocol and kernel implementations.

2 Measurement Methodology

The latency measurement tool we used was a modified version of the common `ping` program, which provides statistics on the round-trip latencies of ICMP Echo Request/Reply messages.[5] After sending an ICMP Echo Request to a destination site, the measuring host waits for the destination to reply with an ICMP Echo Reply. The process involves two time measurements: one occurs before sending the echo request packet, the other occurs after receiving the echo reply packet. Both of these measurements occur at the sending site and the difference between the two times represents the round-trip latency of the packet.

Although this method of measurement has the advantage that it does not require time synchronization across sites, it is impossible with this method to determine the separate outbound and return latencies relative to the target host. We modified the `ping` utility to address this question. Rather than sending ICMP Echo Request packets, our modified implementation sends ICMP Timestamp Request packets to the destination site. The destination responds with ICMP Timestamp Reply packets back to the original source. Now, rather than two timestamps, there are four:

T_0: Before sending the ICMP Timestamp Request packet, the source puts its current time value into the Originate Timestamp field of the ICMP packet.

T_1: After receiving the timestamp request packet, the destination site inserts its current time into the Receiving Timestamp field of the packet.

T_2: Before sending the timestamp reply packet, the destination puts its current time into the Transmit Timestamp field.

T_3: After receiving the ICMP Timestamp Reply packet, the source site calculates its current time, and therefore now has all four time values needed for the one-way latency calculations.

Notice that the first and last of the four timestamps correspond to those used by the ICMP Echo Request/Reply packets. The two new timestamps are introduced by the target host of the ICMP Timestamp Request. The ICMP packet returned to the originating host holds three 32-bit-wide millisecond timestamps; the originating host then inserts the final timestamp upon packet reception. We would like to use the differences between the timestamps as indicators of delays across the network, according to the following relationships:

$$outbound\ latency = T_1 - T_0,$$

$$return\ latency = T_3 - T_2.$$

However, the timestamp differences are not necessarily indicative of *delays* across the network, but rather of *clock states*, together with their characteristic drift and asynchronicity due to local clock adjustment methods to compensate for such drifts. A brief digression on clocks and time measurements will elucidate the situation.

2.1 Of Times and Clocks

The accuracy of the above equations depends on synchronization between the source and the destination sites. If the two end clocks are not kept synchronized according to a protocol such as NTP (Network Time Protocol [4]), the timestamp differences will reflect this and thus interfere with latency measurements which indicate the "absolute time".

Such synchronization between host pairs in the Internet environment may be the case under specific circumstances, but is not generally a reliable assumption. While the NTP time synchronization method assured us of synchronization between the clock on our source host with the T3 NSFNET backbone clocks, which are themselves NTP-synchronized to highly accurate Stratum-1 clocks[1], we had no guarantee of similar clock accuracy on the destination hosts targeted by our probes.

Fortunately, it does not matter.[2] A lack of tight synchronization between the end system clocks will certainly impede a researcher making inferences about absolute time values of a

[1] Stratum-1 clocks in turn receive their synchronization from national atomic time servers via radio signals.

[2] But as it turns out, a lack of time drift signatures in our measurement data indicates that destination host clocks were also using some method of time synchronization to maintain time accuracy.

single measurement. However, the effects we target in our measurements are different. Our thesis is about *fluctuations* of unidirectional latencies. We aim to assess the variation in, rather than the absolute value of, one-way delays.

The core of our hypothesis is that the variance in latencies on the two opposing paths of a connection is often significant. Investigating this hypothesis does not requires accuracy in one single measurement of outgoing and return delay, but rather consistency among a continual series of measurements. We do not compare the two directions of any one single measurement. Thus, we may make references to, but do not depend on, the accuracies of timer mechanisms in machines across the Internet. Tight clock synchronization is not essential to this performance study, nor to the integrity of our conclusions.

We drive this point home by discussing further obstacles to clock accuracies among Internet hosts, which can interfere with interpretation of delay measurements between two Internet sites. Three impediments are machine architectures:

- First, machine architectures on modern workstations have their own internal clock drift phenomena, and many use countermeasures to compensate for it. We observed clock drifts on unsynchronized workstations in the range of 3 to 60 milliseconds per hour.
- The operating system, and the treatment of ICMP packets within its kernel, may differ across machines.
- The delay between machine reception of the ICMP packet and the timestamp execution may not only depend on a specific ICMP protocol implementation, but also on the machine load at packet reception time. Thus, there are both static and dynamic components which contribute to errors in the machine internal timestamp mechanism.

A further obstacle is machine independent, and involves NTP, the very mechanism for clock synchronization among many Internet sites. In order to maintain clock integrity, NTP performs statistical assessment of the clock drift of a host relative to other hosts in the Internet, and induces clock adjustments based on that assessment. If a clock drifts relative to another clock at a specific site using NTP, delay measurements will exhibit periodic oscillation behavior as the clock drift alternates with NTP's attempt to resynchronize it. The maximum amplitude of this drift may approach the order of 10 milliseconds.[3]

Although the above handicaps to latency measurement seem bleak, the recognition of the following fact protects us, and it is a distinction vital to understanding our thesis: the observed effects of all of these factors are dwarfed, by orders of magnitude, by the effects of asymmetric delay variance which we target in our measurements. The effects we target are one to two orders of magnitude larger, as much as 1 second in some cases, than those of NTP or internal machine clock accuracies. The above phenomena, which we list for interested readers, are secondary considerations which do not come close to interfering with our conclusions.

Analyzing the same phenomena with constraints of complete clock synchronization, to determine for example the detailed characteristics of jitter over time, is certainly an admirable project objective, but not the one we have undertaken here. We aim in this paper to present

[3] In fact, our study led us to discover specific flaws in the instrumentation, for example defective NTP implementations, or problems in the time resolution on ICMP Timestamp Replies. See Appendix 6 for details.

and prove a hypothesis about the asymmetry of *delays, not about clocks.* See [2], [3], and [4] for several dedicated studies of clocks in the Internet.

Modifications to the ping utility and synchronization of our source host clock to a highly reliable source were the only instrumental changes we made in preparation for the experiment. Despite the limitations in clock accuracies, we contend that the experiment design is robust enough to allow unidirectional testing with impunity and reasonable confidence in the results.

3 Measurement Procedure

We identified a set of strategically interesting sites to use as targets for our measurements. Our experiments originated from the San Diego Supercomputer Center, at source host *upeksa.sdsc.edu.* We wanted to include both relatively distant and nearby targets; we thus chose a host in Europe, one in Japan, and two at the U.S. east coast:

- *mcsun.eu.net*, in the Netherlands (via the T1 NSFNET and the UUNet link to Europe)
- *scslwide.sony.co.jp*, in Japan (via the T1 NSFNET and the PACCOM Link to Japan)
- *ncri.cise.nsf.gov*, in Washington, D.C. (via the T3 NSFNET)
- *athena.mit.edu*, in Boston (via the T3 NSFNET)

We measured delays to the four hosts from our source host in San Diego. We performed all of the experiments simultaneously for a 96-hour measurement interval, from Saturday, 25 April 1992 at 0:00 to Tuesday, 28 April 1992 at 23:59 GMT. The interval therefore included weekend as well as business hour time. Our graphs use the GMT standard for the axis labels, translation to local time zones is complicated by daylight savings time. Pacific standard time (PST) is GMT-8; Japanese time is GMT+16. 0:00 GMT is thus 8:00 in Japan, the beginning of business hours.

During the 96-hour period, measurements consisted of sequential bursts of 20 ICMP Timestamp Request packets to each target host every minute. The bursts sent a new ICMP Timestamp Request as soon as an ICMP Timestamp Reply was received from the specified destination. All tests used a 1500 byte packet size to utilize the maximum packet size achievable without fragmentation across an Ethernet.[4] We then used the formulas discussed above for the latencies for both directions, $T_1 - T_0$ for the outbound and $T_3 - T_2$ for the return latency.

4 Results

Figures 1, 3, 4, and 5 display absolute differences and round trip delay values between site pairs. The upper half of each figure presents the absolute value of the differences between

[4]Not all of the transit networks between our source and destinations support this maximum packet size. We are in fact aware that the two international sites did indeed fragment packets, with maximum fragment sizes of 1100 and 572 bytes to the Japanese and European sites, respectively. There was no fragmentation along the path to Boston and Washington, D.C.

outgoing and return delays between the source and the remote site, as a function of sample index. Each point corresponds to the difference between the outgoing and return delay assessments for one `ping` packet. Note that the y-axis uses a logarithmic scale. The lower half presents the round trip time (RTT) between the two sites. Note that the y-axis scale for these graphs is linear.

These graphs demonstrate remarkable variances in the differences between the measured timestamps on the two paths, sometimes sustained for significant periods, and often simultaneous with increased round trip delays. Heterogeneous or distant paths between end-systems especially intensify this behavior.

Figure 2 presents the signed value of the difference for the destination site in the Netherlands. The following tables provide summary statistics for the four data sets.

Table 1: Summary Statistics for **Round Trip Delays** from SDSC to Four Remote Hosts (msec)

Destination	Min.	25%	Median	75%	Max.	Mean	Std. Dev.
Europe	474	486	508	642	23520	617.2	305
Japan	913	934	948	1006	4904	1046.0	311
Boston	131	135	136	139	4218	141.0	46
Washington, D.C.	175	181	184	198	14540	190.9	49

Table 2: Summary Statistics for **Differences in Single Direction Delays** from SDSC to Four Remote Hosts (msec)

Destination	Min.	25%	Median	75%	Max.	Mean	Std. Dev.
Europe	-1992	45	59	112	23090	104.2	280
Japan	-3104	60	73	99	5985	116.7	304
Boston	-2120	17	26	35	3616	26.4	102
Washington, D.C.	-14320	33	39	45	3180	41.3	94

Table 3: Summary Statistics for **Absolute Differences in Single Direction Delays** from SDSC to Four Remote Hosts (msec)

Destination	Min.	25%	Median	75%	Max.	Mean	Std. Dev.
Europe	0	49	62	136	23090	130.4	269
Japan	0	62	74	108	5985	152.7	288
Boston	0	19	27	36	3616	32.5	100
Washington., D.C.	0	33	39	45	14320	42.2	94

4.1 Changes in Intermediate Paths

Selecting a specific site to examine in more detail, we performed additional measurements to `mcsun.eu.net`. Figure 6 reflects these measurements and illustrates a particularly interesting phenomenon. The figure indicates a significant jump in delay in the outbound direction from SDSC to Europe, sustained over several hours, and precisely coincident with a drop in the TTL field[5] from 238 to 235.

In most cases, the TTL field is an indication of the hop count of the packet. A router is expected to decrement the TTL field of a packet every second, and once as it switches the packet through the router. Thus, in noncongested networks, where nodal queues are unlikely

[5] TTL refers to the Time to Live field of an IP packet. See [6].

to hold packets for longer than one second, one can fairly safely derive the hop count of a packet by subtracting the final TTL value from the initial TTL value.

Changes in TTL values during specific performance tests can thus reveal changes in network routing. For two sequential packets from the same source to the same destination, a change in the TTL field likely indicates a path change between the source and destination.

A host has access to the final TTL value for packets which are destined for it, but not for those received by remote destinations. Therefore, while our instrumentation allows us to monitor delay in both directions, it only allows us to monitor the TTL field for the return direction.

This background explains our limitations in interpreting certain segments of delay data. It is likely that the TTL decrease mentioned above reflects a routing change, specifically, an increase in the number of hops *in the direction from Europe toward SDSC* (Figure 6), and would likely result in a change of the delay in that direction. One can most likely attribute this shift in single direction latencies, compounded with the dynamic jitter, to an asymmetric route change between the two end-systems in Europe and the U.S.

Unfortunately, as outlined above, we do not have access to the TTL for our outgoing direction. Thus we cannot correlate an increase in delay with an increase in an outgoing hop count. The existing instrumentation limits us to somewhat speculative interpretation of the data: perhaps a link or node failure has caused routing mechanisms to alter one path, leading to a significant jump in delay in one direction but not the other.

5 Discussion

We contend that the data we have presented justify reasonable doubt in the perception of validity of the "divide-by-two" approach to assessing single direction latencies in a highly aggregated, heterogeneous, wide area network environment. Our measurements reveal three points of interest:

1. In the course of observation we see long periods of time where a major and lasting change appears in the values of the delay differences. Given our fine-grained view of the data and the lasting nature of the changes (hours in some cases), we view these events as static, relative to the dynamic jitter on a packet-by-packet basis. In some cases these events suggest route changes, possibly asymmetric, as indicated by a concurrent change in the receiving TTL of the ICMP Timestamp Replies.

2. Much more prevalent than these longer-term changes is the dynamic jitter which occurs on a packet by packet basis. These asymmetric discrepancies in delays are especially visible on long paths with limited bandwidth, particularly in areas where the volume of data in one direction far surpasses that in the other, yielding non-symmetric resource contention. We suspect that queueing delays due to congestion are responsible for the variance in delays. These variances in delays are far too large in magnitude and rapid in fluctuation to be an artifact of clock offsets, drifts, or NTP infirmities. Rather, they indicate definite distortion of symmetry over significant periods of time.

3. During the course of the project we observed several strange phenomena, some of which we were able to explore further. As described earlier, the use of ICMP Timestamps to assess single direction delays assumes strict coupling between the end-systems of the measurement to yield accurate millisecond time. Certain end-systems revealed problematic behavior in certain protocol and kernel implementations, which we detail in Appendix 6.

6 Conclusion

Our experiments demonstrate that round trip latency measurements can serve only to approximate directional delays in the network. In determining single direction latencies, one should acknowledge that dividing round trip latencies in half can yield misleading results. The distortions are attributable to both static and dynamic components. Statically, asymmetric paths can exert a lasting influence on the delay differences, where path changes can cause sudden and persistent changes in latency in one direction. On a more dynamic basis, resource contention in the network introduces queuing delays in nodes which manifest themselves through short term jitter. Our measurements indicate that both static and dynamic effects can lead to asymmetric delays, and thus should receive consideration in a viable model of network latencies.

This claim has significant implications for high-speed, multi-application networking environments, and particularly for continuous media applications, which require predictability of delay. For example, designers of real-time multimedia applications on wide-area, highly traffic aggregating networks, should be aware that across the periphery of such an environment, symmetric delay will serve unpredictably as a model of network performance.

Appendix A: Historical Note – Uncovered Implementation Flaws

Although not one of our intended objectives, in the course of our study we discovered problematic behavior in certain protocol and kernel implementations. The anomalies which we observed include:

- The measurement data exhibit periodic, wavelike time drifts cycling over hour periods with a 1-10 msec amplitude range, with the phases of the two directions being offset by 180 degrees. In testing the hypothesis that this behavior is a result of NTP's compensatory behavior, we disabled time synchronization to see if the phenomenon disappeared. Indeed it did, lending credibility to our theory that this phased behavior is an effect of the NTP synchronization protocol compensating for time drifts at the observed nodes. NTP compensations create the appearance of a time shift forward for one path, backward for the opposite path, leaving a low amplitude NTP signature which is offset by 180 degrees in the two directions. We discussed specific time drift and compensation issues with Dave Mills, designer of the widely deployed NTP protocol in the Internet[4], who helped us further investigate this effect. Based on dispersion between hosts according to NTP daemon-provided information received from NSFNET backbone logs, Mills suspects an overloading of the primary time servers with more clients than they can handle with reasonable integrity. A more distributed national high precision time service over the national backbone would address this problem.

- There was a conspicuous lack of millisecond-level resolution in ICMP timestamp responses from some major network nodes. We notified the responsible parties at the relevant locations, who appreciated our pointing out the error and assured us the problem would receive attention.
- Gradual but substantial clock drifts characterized several remote systems which were not time-synchronized with the rest of the Internet, rendering many gathered data sets difficult to use. Evidence for clock drift includes a constant slope in offset between the timestamps of the two measurement end points, where the slope of the drift expresses the magnitude of inaccuracy of the local timekeeping in hosts. Calibrating such data requires one to develop a model for the drift and to adjust the measured data to derive more faithful delay values.
- A 32-bit clock timer imposes limits on resolution, as revealed by rollovers in millisecond clocks on hosts during some of our measurements. A program analyzing the timestamps must compensate for such protocol deficiencies by taking the times before and after the rollover into account.
- Millisecond clocks will prove insufficient for finer-grained changes in very high speed networks, e.g., in gigabit projects, and even for T3 networks. A new ICMP packet type should address the issue of nanosecond clock resolution on ICMP Timestamps.

Although not central to our research, these anomalies either have resulted in a positive impact, or suggest areas where more critical investigation is necessary.

Acknowledgements

We would like to thank Dr. Joseph Pasquale for encouraging us to pursue this study.

References

[1] D. D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *SIGCOMM 92*, pages 14–26, August 1992.

[2] D. Mills. "Internet Time Synchronization: the Network Time Protocol". Internet Request for Comments Series RFC 1129, 1989.

[3] D. Mills. "Measured Performance of the Network Time Protocol in the Internet Systems". Internet Request for Comments Series RFC 1128, 1989.

[4] D. Mills. "Network Time Protocol, Version 2, specification and implementation". Internet Request for Comments Series RFC 1119, 1989.

[5] J. B. Postel. Internet Control Message Protocol. Internet Request for Comments Series RFC 792, 1981.

[6] J. B. Postel. Internet Protocol. Internet Request for Comments Series RFC 791, 1981.

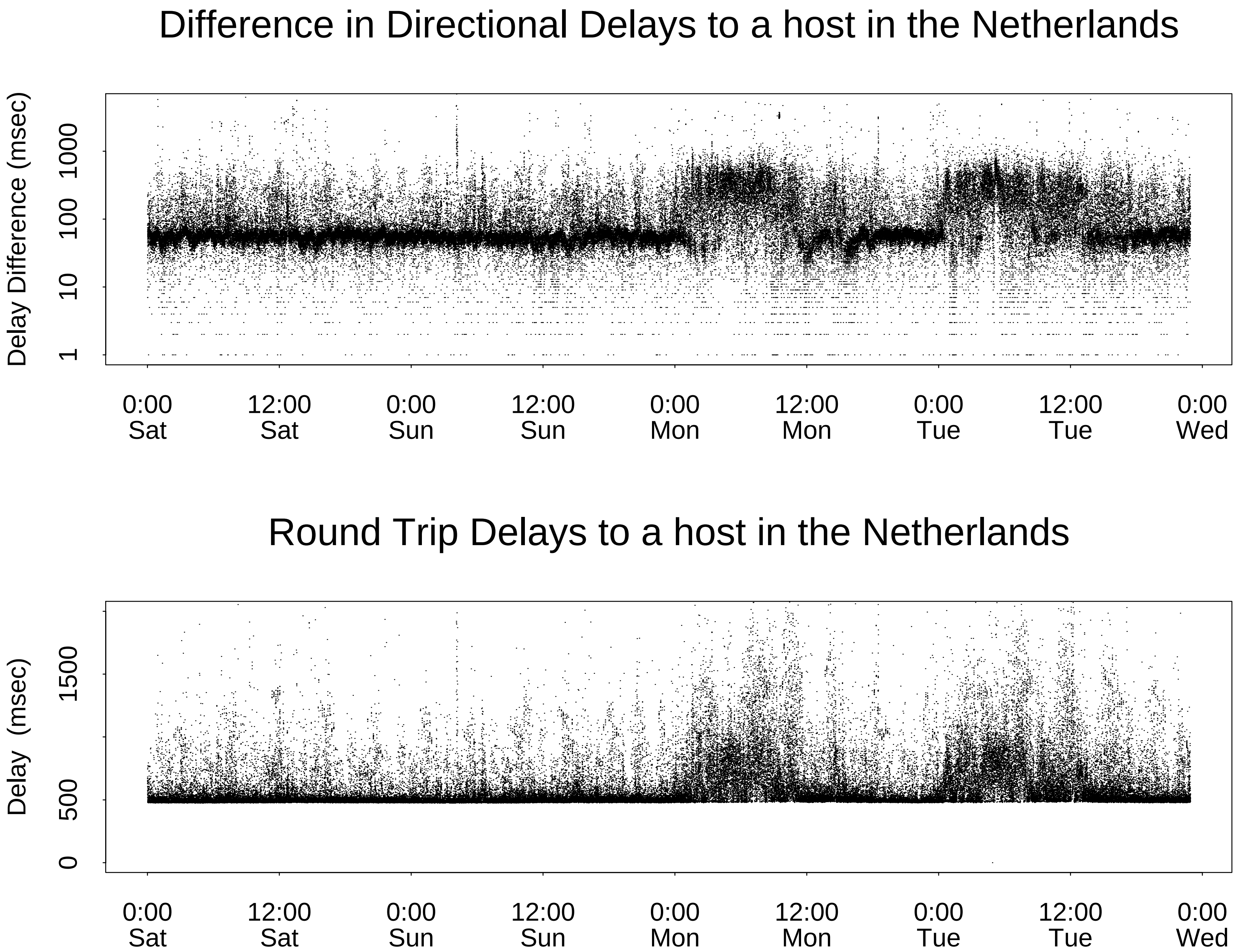

Figure 1: Absolute Differences in Unidirectional Delays and Round Trip Delays from SDSC to *mcsun.eu.net*

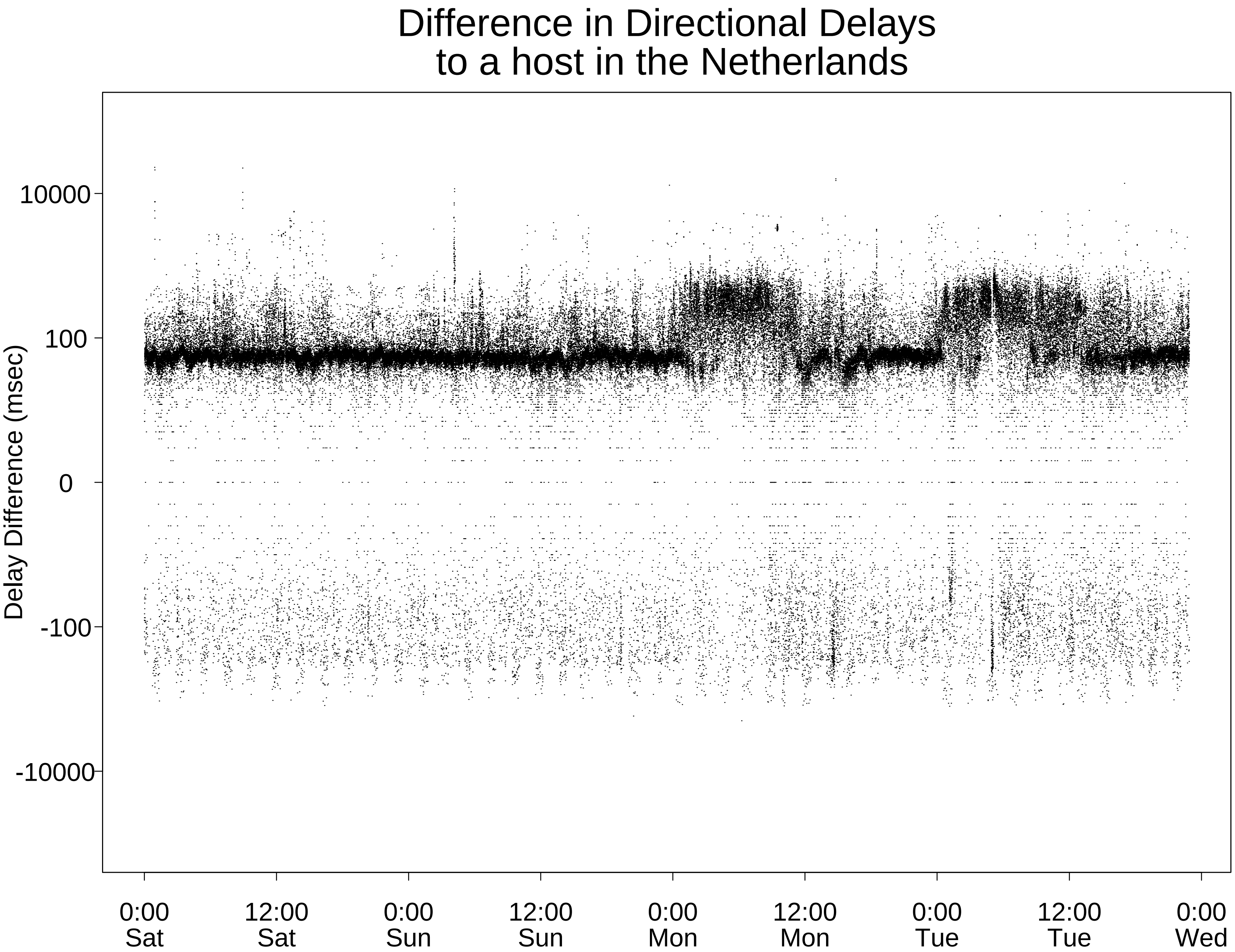

Figure 2: Differences in Unidirectional Delays from SDSC to *mcsun.eu.net*

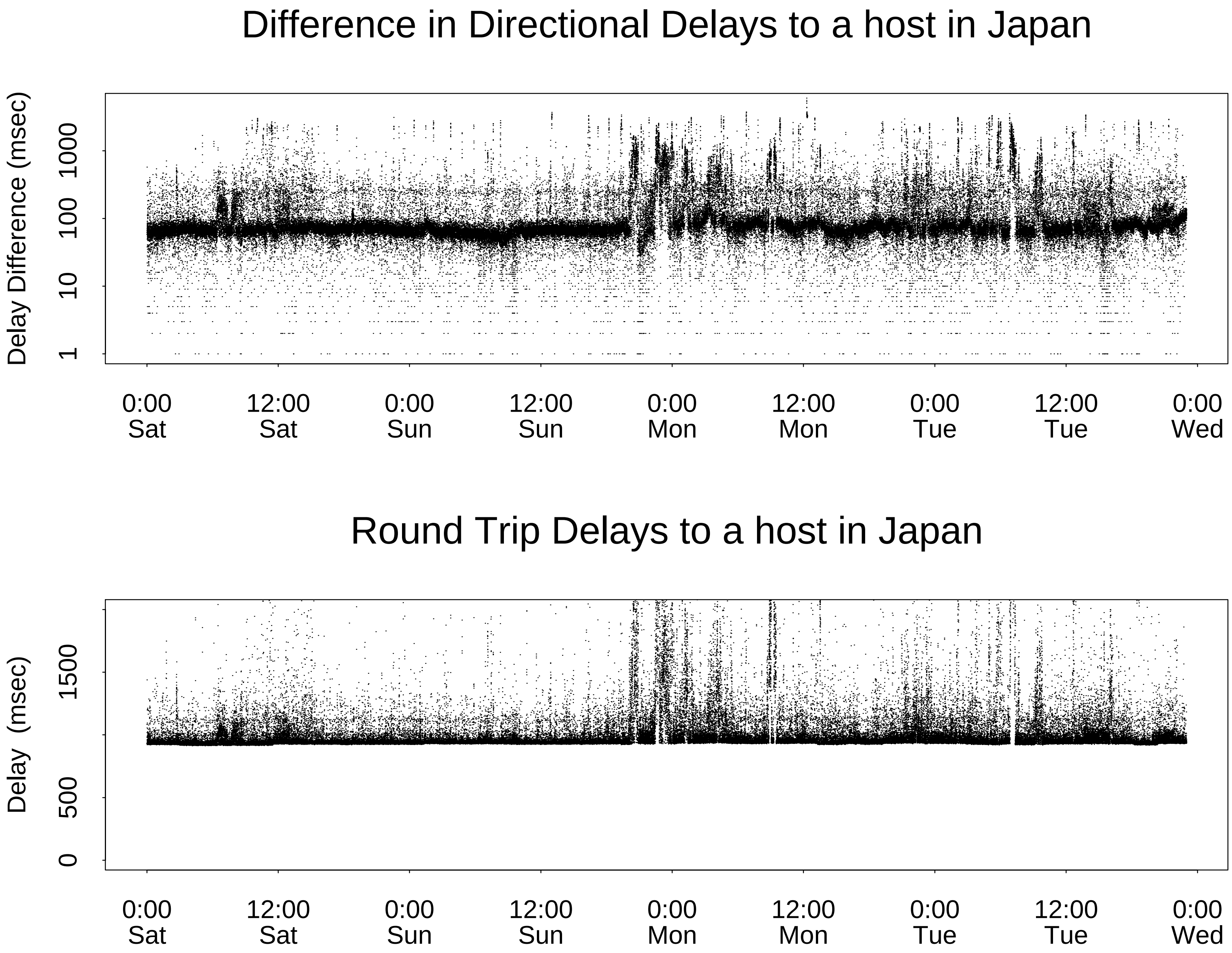

Figure 3: Absolute Differences in Unidirectional Delays and Round Trip Delays from SDSC to *sc-shuide.sony.co.jp*

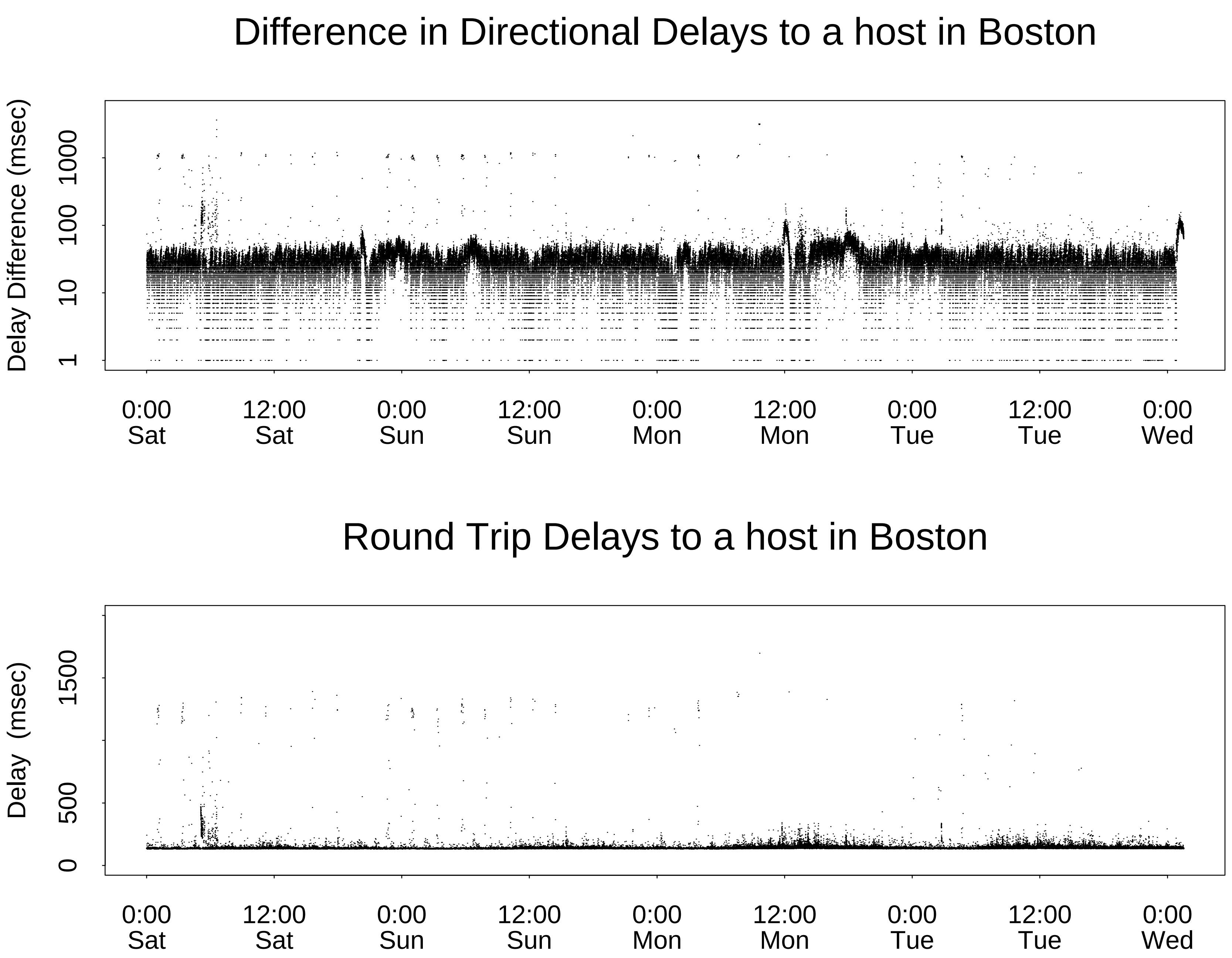

Figure 4: Absolute Differences in Unidirectional Delays and Round Trip Delays from SDSC to MIT for *athena.mit.edu*

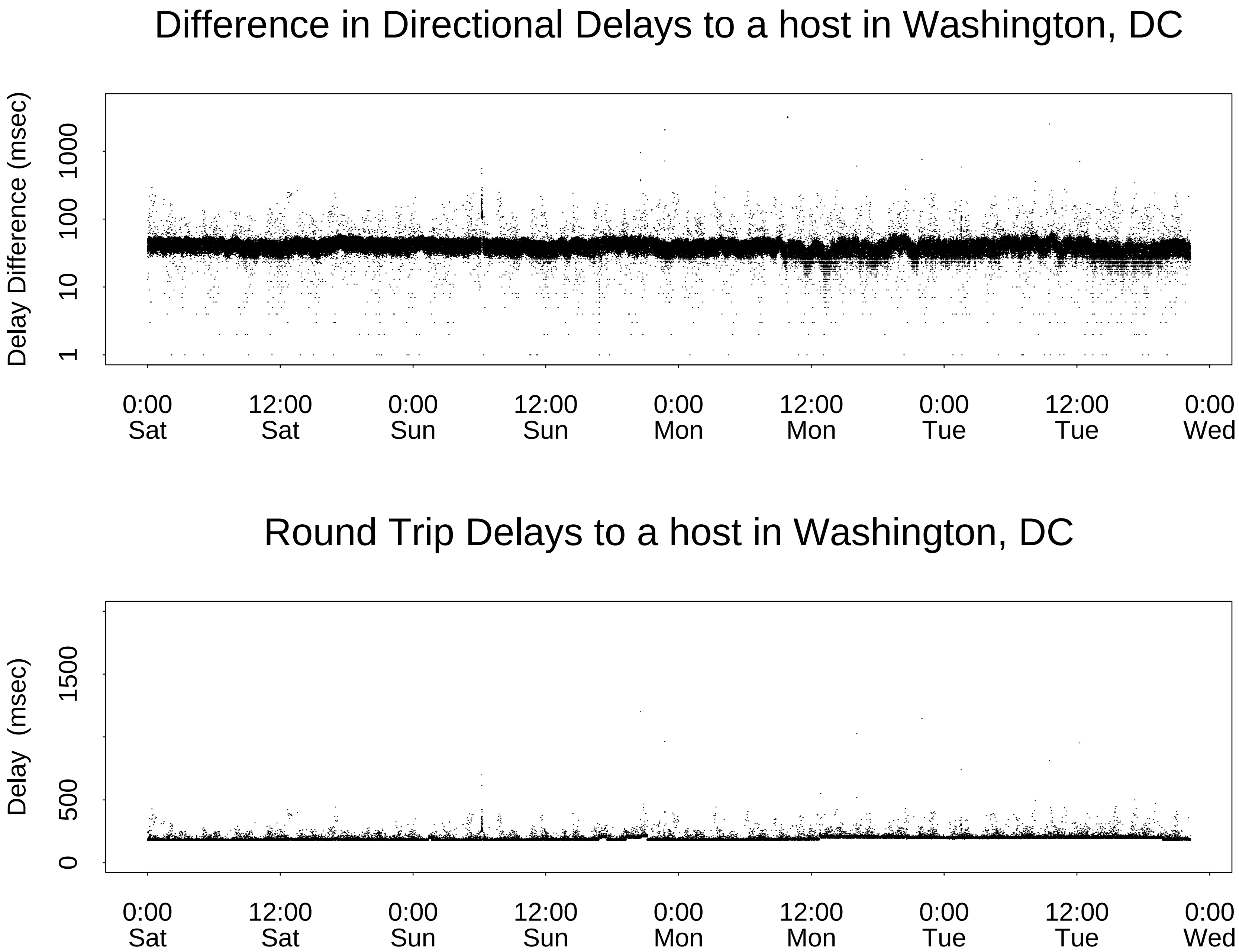

Figure 5: Absolute Differences in Unidirectional Delays and Round Trip Delays from SDSC to *ncri.cise.nsf.gov*

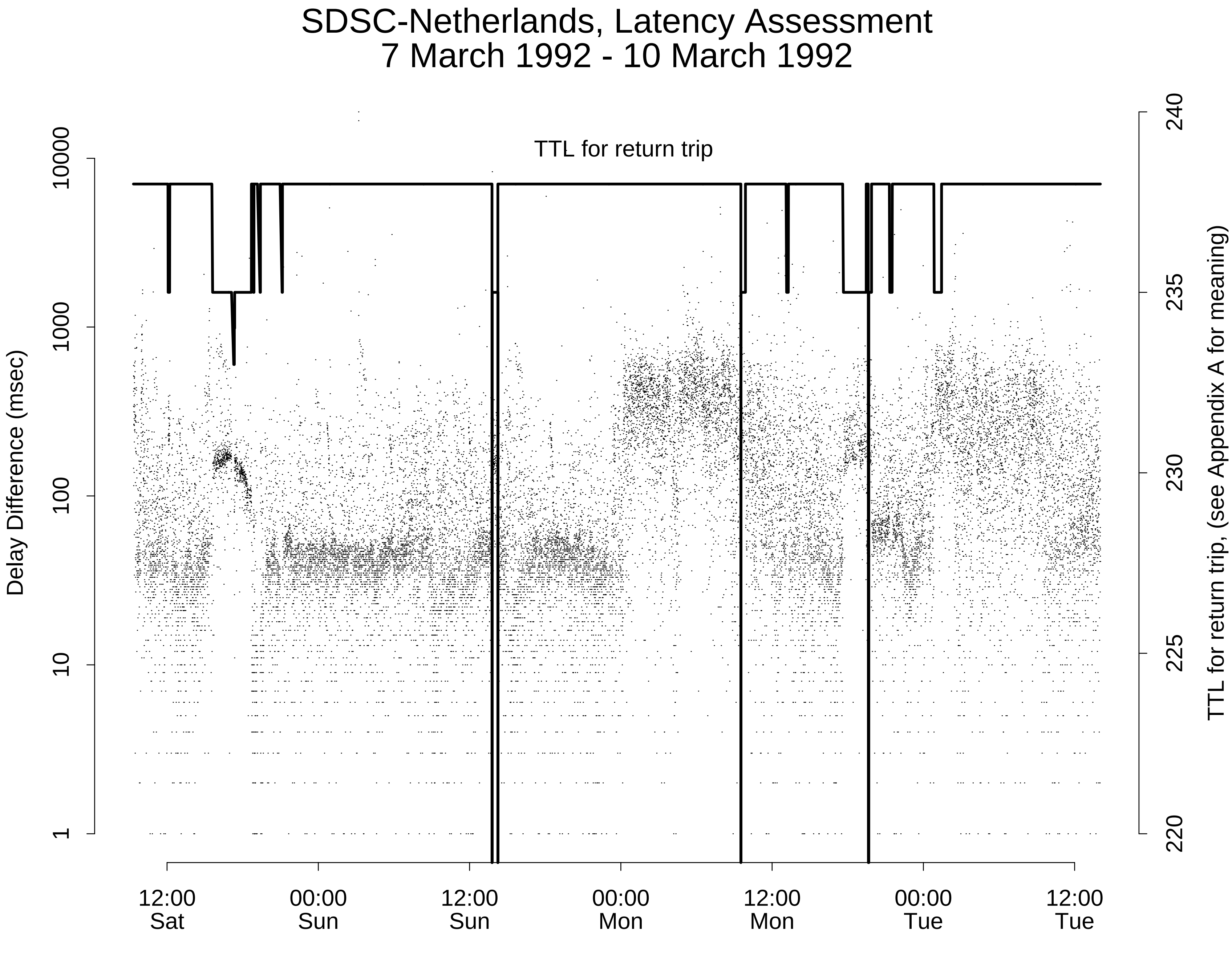

Figure 6: Single Direction Delays from SDSC to *mcsun.eu.net*