Internet Performance and Reliability Measurements for the High Energy Physics Community ⋆

David E. Martin

HEP Network Resource Center, Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

R. Les Cottrell, Connie A. Logg

Stanford Linear Accelerator Center, Stanford, CA 94309, USA

Collaborative HEP research is dependent on good Internet connectivity. Although most local- and wide-area networks are carefully watched, there is little monitoring of connections that cross many networks. This paper describes work in progress at several sites to monitor Internet end-to-end performance between hundreds of HEP sites worldwide. At each collection site, ICMP ping packets are automatically sent periodically to sites of interest. The data is recorded and made available to analysis nodes, which collect the data from multiple collection sites and provide analysis and graphing. Future work includes improving the efficiency and accuracy of ping data collection.

Key words: Wide Area Network, Internet, Monitoring, Reliability, Performance, ICMP, IP, Ping

1 Introduction

High energy physics (HEP) research is characterized by large collaborations whose members are widely scattered at universities and laboratories throughout the world. Although rarely mentioned prominently in project plans, wide-area networking is critical to the success of most collaborations. Much of the day-to-day work of a collaboration is done over computer networks, from such

⋆ This work was supported by the United States Department of Energy under contracts DE-AC03-76FO0515 and DE-AC02-76CH03000

simple tasks as reading electronic mail to such complex tasks as event reconstruction. Today, the Internet and the Internet Protocol (IP) are used for almost all HEP data exchange.

The current worldwide Internet consists of over 30,000 "networks" (both local-area and wide-area) interconnected at various points to provide a seemingly single network to end users. Each network is typically run by an organization that monitors the network's physical links, routers and logical interconnections. For example, the Energy Sciences Network (ESnet) runs a network to interconnect major energy research facilities in the United States. ESnet provides 24-hour monitoring of all lines and equipment. Likewise, MCI runs and monitors the vBNS network that interconnects research institutions with National Science Foundation (NSF) supercomputer centers.

However, connections between users on different networks are rarely monitored. In order for an ESnet site to reach an NSF center, the traffic will cross a number of different networks. Monitoring performance and reliability of connections across many networks is difficult since no single organization has access to statistics stored on all intermediate nodes. Traditional network monitoring tools based on protocols like the simple network management protocol (SNMP) are unusable with such access.

In 1994, the Stanford Linear Accelerator Center (SLAC) embarked on a task to study connections to research sites collaborating with SLAC.[1] SLAC staff developed a system to collect network performance and reliability statistics and present them in both tabular and graphical formats. In 1996, the ESnet Site Coordinating Committee (ESCC) formed the Network Monitoring Task Force which chose to extend the SLAC work to allow for monitoring of connections between many different sites. The HEP Network Resource Center has been leading the effort in developing this new system. This paper details the techniques for data collection and dissemination used in the new system. A companion paper details the analysis and presentation of data.[2]

2 Technique Used

Since HEP traffic often crosses many different networks, it is impractical to try and gain access to statistics of transit nodes. Negotiating access rights to router statistics with even a few transit networks has proved to be impossible. The decision was made, therefore, to treat the entire network of intermediate nodes as a black box and monitor end-to-end performance only. Throughout this paper, such end-to-end connections will be referred to as links. Although this technique greatly simplifies data collection, it somewhat limits the utility of the data in diagnosing problems. Because Internet Control Message Protocol (ICMP) messages are almost universally supported, and because the ping command is ubiquitous, ICMP ECHO_REQUEST messages as generated by the UNIX ping command were chosen as a basis for network monitoring.

All nodes running IP are required to respond to ICMP messages, a family of packet types used to perform various low-level IP routing maintenance and network diagnostics.[3] An ICMP ECHO_REQUEST packet (also known as a *ping*) has an IP and ICMP header (which contains a sequence number), followed by an 8-byte timestamp, and then a number of "pad" bytes used to fill out the packet to a specified length. When an Internet node receives such a packet, it responds with an ICMP ECHO_RESPONSE packet with the same timestamp, sequence number and pad bytes. Since this is a datagram protocol, either the ECHO_REQUEST or ECHO_RESPONSE packet may be lost or duplicated.

A very common application of this protocol is the UNIX *ping* command which (by default) sends a single 64-byte ECHO_REQUEST packet to the host specified and reports whether a resulting ECHO_RESPONSE packet was received within twenty seconds.[4] Typical options to the ping command allow control of the number of ECHO_REQUESTs sent, the interval between each request, the number of pad bytes, and the time to wait for an ECHO_RESPONSE. When used in batch mode, the ping command gives the percentage of packets lost and the minimum, maximum and average response times over all responses received.

ICMP messages are not usually available at the user level and, in fact, on a UNIX system a normal user is forbidden from sending or receiving any type of ICMP packets. On UNIX, therefore, the ping command runs at the root level by doing a *setuid* upon invocation. Receipt of ECHO_REQUEST packets and response with ECHO_RESPONSE packets are performed at a low level in the operating system without user-level intervention, making it a good probe of network response time rather than system response time. Unless a system is very heavily loaded, ping packets should be received and responded to without significant delay. An exception to this is some brands of routers, which give low priority to ICMP messages. They may ignore ICMP messages even during relatively light load.

At first anecdotal evidence was used to verify ping as a good measure of user-perceived network performance and reliability. User complaints about a link could often be matched to large packet loss or high response time on that link. Similarly, user reports of improved performance were often matched to reductions in packet loss or response time. Although the correlation was not perfect, data from ping studies was successfully used to choose Internet service providers for SLAC telecommuters, among other uses. In order to provide a more rigorous validation, a study was done to compare times of Hypertext

Transport Protocol (HTTP) transfers with ping response times. The study showed that the response time seen by ping is a good predictor of application-level network performance.[5] This correlation was also shown in [6].

Ping, though, is not a perfect measure. It is more likely to give a false positive, indicating a problem where there is none, rather than a false negative, indicating the network is fine when it is not. Also, pings only give an instantaneous view of the state of the network so periodic pings may miss transient problems.

3 Data Collection and Distribution

Data collection is performed on UNIX workstations by running a Perl script that is scheduled by the *cron* facility. This Perl script is called *pingtime*. Every thirty minutes, a list of hosts is scanned sequentially. For each entry in the list, one ping packet with a 100-byte payload is sent to the host, then ten such packets are sent to the host, then ten ping packets with a 1000-byte payload are sent to the host. All pings are sent at intervals of at least one second. The first ping is used to prime router caches and address resolution tables and its results are discarded. 100-byte pings were chosen to represent interactive traffic, 1000-byte pings to represent batch transfers. The results of the of the 100-byte and 1000-byte bursts are stored in a single line that contains:

- IP name of destination node;
- IP address of destination node;
- date and time of beginning of batch job (in long format);
- percentage of loss, minimum, maximum and average of response times (for 100-byte pings);
- percentage of loss, minimum, maximum and average of response times (for 1000-byte pings);
- date and time of first ping to this particular node (in both UNIX ctime and long format).

The line is written into a file containing the entire month of data collected by the source site. This format is based on the original SLAC system. Note that it does not record the source node since all data collection was done from a single node at SLAC.

A new Perl script is being phased into use. It does a number of pings in parallel to increase the number of nodes that can be pinged. In addition it uses a more compact format:

- IP address of source node;

- IP address of destination node;
- size of pings;
- date and time of first ping (in UNIX ctime format);
- percentage of loss, minimum, maximum and average of response times. (If percentage of loss is 100%, minimum, maximum and average are omitted.)

Like the original system, all data for the month is stored in a single file. Since current data analysis and presentation software is still based on the original format, a Perl script is available to convert from the new format to the old.

In the original SLAC system, all pinging originated at a single node. This made analysis and reporting simple, but provided only a limited view of the network. The current system improves the breadth of sites examined by providing multiple *collecting sites*. A collecting site is one that has agreed to compile a list of nodes (called *remote sites*) to ping and has agreed to run *pingtime* on a local node. In addition, this node must run the *ping_data* CGI/Perl script and an HTTP server. The *ping_data* program allows a remote site to retrieve data for links over a specific time period. The goal is to have several collecting sites per collaboration or other affinity group.

In the original SLAC system, all data analysis was done on a single node at SLAC. The current system makes use of a number of *analysis sites*. Analysis sites run the *ping_collect* Perl program which collects data from the collecting sites and store a copy locally. Analysis sites run the SAS environment which provides for graphical and tabular analysis of the data. This analysis is discussed in detail in [2]. The results of the analysis are made available through the world-wide web in pre-packaged overnight reports and dynamically generated reports.

4 Future Work

One problem with the current system is that network monitoring at a fixed interval can completely miss performance changes with a periodic nature. Switching to random times between bursts of ping packets would provide better coverage. By using a Poisson distribution, the same average number of ping packets sent per day could be maintained, thus using the same bandwidth as the fixed interval probes. However, this is impossible to achieve with the current technique of using cron to schedule probes. The probe Perl script must be re-written as a daemon with a built-in scheduling system.

Another problem in the current system is that monitoring only end-to-end performance provides little help in trouble resolution. Treating the connecting internetwork as a black box simplifies monitoring, but provides only an indication of the severity of any problems. Often, a network manager will try to diagnose a problem by using *traceroute* then pinging successive nodes in list generated. Such steps could be automated and performed whenever the base system shows a problem link. Still, a network manager must be available to evaluate the information generated by the automated detailed testing.

References

[1] Connie A. Logg, R. Les Cottrell, "Network Monitoring and Performance Management at SLAC," SLAC-PUB-95-6744, Stanford Linear Accelerator Center, Stanford, CA. Presented at Networld+Interop Engineers' Conference, Las Vegas, March, 1995.

[2] R. L Cottrell, Connie A. Logg, David E. Martin, "What is the Internet Doing For and To You?," submitted for publication in Proceeding of the Computing in High Energy Physics Conference, Berlin, Germany, Apr., 1997.

[3] John Postel, "Internet Control Message Protocol," RFC-792, DARPA Internet Program, Sep., 1981.

[4] SunSoft, "SunOS 5.4 Reference Manual, Section 1M," SunSoft, Mountain View, CA, 1994.

[5] R. Les Cottrell, Charles D. Granieri, John H. Halperin, Gary Haney, Connie A. Logg, David E. Martin, William R. Wing, "Internet Monitoring in the Energy Research Community," submitted for publication in IEEE Communications Magazine, 1997.

[6] Robert L. Carter and Mark Crovella , "Server Selection using Dynamic Path Characterization in Wide-Area Networks," Proceedings of IEEE INFOCOM '97, Kobe, Japan.