

Network Realism In Testing

Philip Joung

Workshop on
Performance and Reliability
October 2003



Spirent Communications
1175 Borregas Avenue
Sunnyvale, CA 94089
Phone: 408.541.1010
Fax: 408.541.1098

www.spirentcom.com

Copyright 2003 by Spirent Communications, Inc. All rights reserved.

Table of Contents

1 Introduction	1
2 Study Methodology	1
3 The Spirent Communications Internet Realism Study	2
4 Network Realism	3
4.1 Packet Loss	3
4.1.1 Study Results	5
4.2 IP Fragmentation	5
4.2.1 Study Results	6
4.3 Link Speeds	7
4.3.1 Study Results	8
5 Conclusion	9
6 References	10
7 Acknowledgement	10

1 Introduction

Has your experience on the Internet been completely acceptable today? How about in the past week, or the past month? Did all the Web pages you visited load properly and promptly? Did a download fail halfway through? Indeed, the performance and reliability of the Internet continues to arise as a question, and many of us begrudgingly accept these common problems. Why does such a critical, valuable network continue to fail in the face of such innovation, investment, competition, and sweat? While we may never see problems disappear on large networks such as the Internet, what we should expect, and indeed, what is possible is certainly better than what we have today. Robust testing can help.

The benefits of performance testing and capacity assessment have been well established. Indeed, many presentations, books, and magazine articles discuss why testing is important and how it can contribute to many benefits, including reliability, availability and performance. In order to achieve maximum benefits from the testing effort, one must take the time to ensure robust testing and procedures.

Part of robust testing involves achieving realism in the test traffic. Indeed, significant differences do indeed occur with oftentimes minor changes to the traffic behavior. A change in the way someone uses a Web site, or a change in the way an application writes data into the database can create major performance changes, especially when multiplied many times across hundreds or thousands of users. The other aspect of realism that is just starting to gain more attention is the aspect of network realism—being able to simulate and measure the impact of common network issues such as packet loss, IP fragmentation, and link speeds.

This paper focuses on several items to consider when conducting load testing and capacity assessments, including results discussion from several in-house studies. Your thoughts and feedback on this are requested and much appreciated.

2 Study Methodology

In the Spirent Communications Internet Realism Study sought to assess a Web server's ability to handle different types of traffic. After establishing a baseline with clean traffic, one parameter was varied at a time and the test rerun, saving the detailed performance results for later analysis.

The test network is shown below in Figure 1. It consisted of one Web server, one 10/100 Ethernet switch, and one Avalanche capacity assessment appliance, which has the ability to inject and simulate network realism parameters.

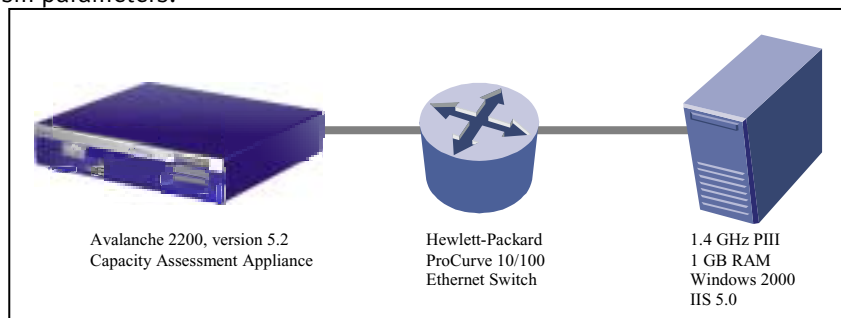


Figure 1: The test network used for the Spirent Communications Internet Realism Study.

The Web server was built using standard PC components, including an Intel server motherboard. It had one 1.4 gigahertz Intel Pentium III processor, 1 gigabyte of ECC SDRAM, one 20 gigabyte 5400 rpm IDE hard drive, and one 10/100 Intel network interface. The operating system used was Windows 2000 Server with service pack 2, and the Web server software was Microsoft Internet Information Services version 5.0.

The switch was a Hewlett-Packard ProCurve 4000M with a switching fabric capacity of 3.8 gigabits per second, well beyond any bandwidth encountered in this study.

The Spirent Communications Avalanche 2200 appliance was running software version 5.2, and was equipped with four 10/100 interfaces. More information about Avalanche can be found at the Spirent Communications Web site at www.spirentcom.com/avalanche/.

One interface from the Web server and one interface from the Avalanche were connected to the switch using standard category 5 unshielded twisted pair cabling. The remaining three interfaces in Avalanche were disabled in the software configuration. A VLAN was setup on the switch to isolate test traffic from any other traffic.

The Avalanche was configured to send simulated traffic to the Web server with the following conditions, defined as the “clean-run”:

- Test duration: 127 seconds
- Files: 1024 bytes, 128 bytes, 256 bytes, 512 bytes, 10 kilobytes, 128 bytes, 256 bytes, 512 bytes
- Browser emulation: Microsoft Internet Explorer 5.0
- HTTP 1.1 protocol
- Congestion control enabled
- No user think time
- No packet loss
- No link latency (full 100Mbit speed)
- No IP fragmentation

During each run, Avalanche increased the load over a 60-second span to 3,600 simultaneous users, maintained the 3600 users for another 60 seconds, and then continued receiving responses until they were completed. This load of 3,600 users, under the clean-run parameters above, allowed the Web server to successfully serve all HTTP requests without errors or failures.

3 The Spirent Communications Internet Realism Study

The results of the clean-run are shown in table 1.

Attempted number of transactions	378,672
Successful number of transactions	378,672
Successful transactions/sec	2,981
Average response time	13.5 ms
Total TCP connections established	47,334
Average incoming bandwidth in Mbps	45.69

Table 1: A summary of the results of the clean-run assessment in the Spirent Communications Internet Realism Study.

4 Network Realism

4.1 Packet Loss

We have long suffered the ill effects of packet loss. For the end user, packet loss translates into slow network downloads, browser timeouts, network errors, and poorly functioning applications, especially time sensitive real-time applications such as streaming audio and video. Whenever a network packet does not reach its intended destination or arrives corrupted, packet loss has occurred. Since the data has been lost, it is often up to the originating device to detect that an acknowledgement has not been received and resend the data or the request. This detection can take seconds to complete, creating long delays that can, at best, lead to user frustration and worse, might result in lost data or lost revenue. Often, frustrated Web users become impatient and simply attempt to reload the page, further adding to the original network load. Alternatively, users give up and go elsewhere.

Packet loss primarily occurs in two ways: network congestion and errors in network transmission. As packets traverse the Internet, any packets that exceed a certain network threshold may be discarded. A busy Web server could also be the source of packet loss, essentially ignoring incoming packets that arrive successfully to the server. The IP protocol, which carries most of the data on the Internet, has built-in checksums to validate the integrity of the data being transferred. If, during transmission, the checksum does not match (most likely from network noise/errors), the packet will be discarded.

Slow start and congestion avoidance have been devised as methods to help mitigate the effects of packet loss [RFC2001]. Slow start works by having both the network sender and receiver start with smaller amounts of information that require immediate acknowledgement. As the connection proves to be stable and loss-free, more information is transmitted, allowing for improved performance. The IP protocol uses congestion avoidance when packet loss is detected, sending data in smaller sized chunks to reduce the likelihood of continued congestion and packet loss. Both of these techniques (which are not required in IP) can help mitigate the prevalence and adverse effects of packet loss, but at the cost of reduced performance. Therefore, packet loss can still cause problems even for networks connections that successfully get rid of packet loss through slow start/congestion avoidance.

How common is packet loss? Figure 2 provides evidence for the prevalence of packet loss, with global packet loss averaging about 4%. One can assume that the reported packet loss values occur even with slow start and congestion avoidance. Internet Traffic Report takes measurements from throughout the globe to obtain its ratings, providing a useful glimpse of the state of the Internet now and in the past. See [ITRo3] for further information.

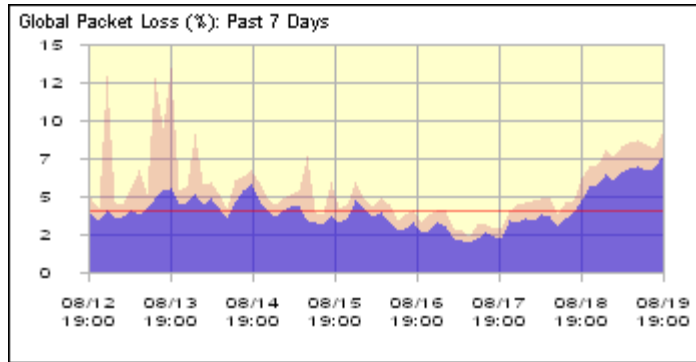


Figure 2: Global packet loss across the Internet over a 7-day period. Red indicates maximum and purple indicates the average. Data from The Internet Traffic Report.

The United States National Air and Space Administration (NASA) conducted a study in 1998 to find the impact packet loss has on their network [NASA98]. They discovered that packet loss caused a significant performance issue for the FTP protocol, dropping throughput by 50% with only 3% packet loss (Figure 3).

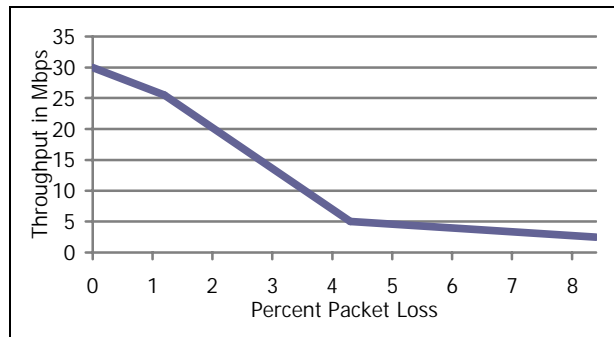


Figure 3: FTP throughput degradation with packet loss from NASA study on packet loss [NASA98].

4.1.1 Study Results

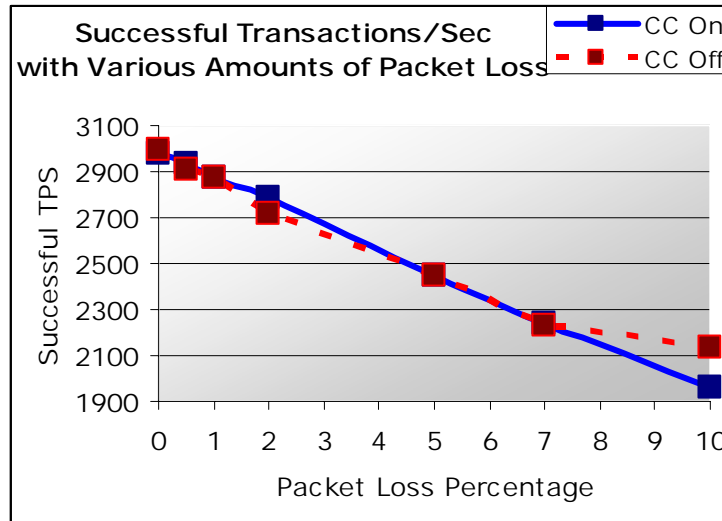


Figure 4. HTTP successful transactions/second with varying amounts of packet loss. The effect of congestion control is also included on this graph, with the solid line denoting congestion control on and the dotted line corresponding to congestion control off.

Figure 4 shows the result of packet loss with and without congestion control. The detrimental effect of packet loss is quite apparent in this graph, causing drops in transaction rate at all levels of packet loss. The effect of congestion control is less clear in this study, with little change in performance regardless of its use or not.

4.2 IP Fragmentation

Network devices (including, of course, ones on the Internet) have a configured maximum transmission unit (MTU) in bytes. A sender on the network will typically start a network transmission by sending packets that fit its MTU. As a packet is transferred, as long as each subsequent link’s MTU equals or exceeds the initial MTU, the packet will arrive at the destination unfragmented. However, if the packet encounters a link with a smaller MTU, the packet must be fragmented (unless the packet is marked with a don’t-fragment flag, in which case the packet is simply discarded—another potential source of packet loss).

What happens to a packet that is fragmented? Let’s take a hypothetical example where an original packet of size 500 encounters an MTU of 100. It will be fragmented into 5 packets, right? Actually, because of overhead in the headers, it is fragmented into at least 8 packets as shown in Figure 5.

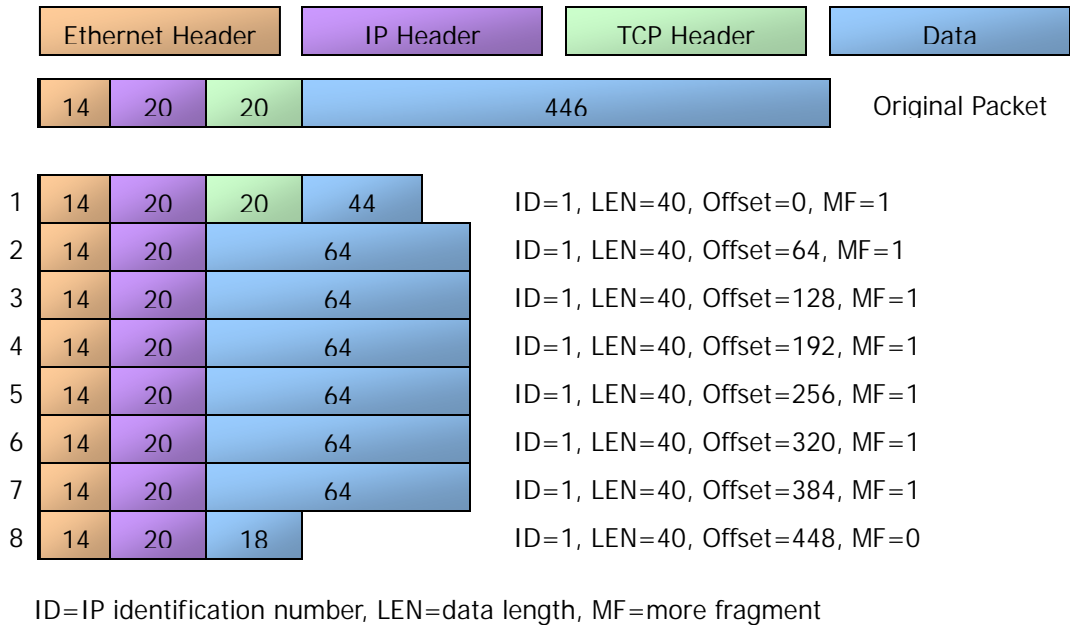


Figure 5: A 500-byte packet that encounters a 100-byte MTU will be fragmented into 8 packets. The TCP header will appear in only the first packet. The offsets must be a multiple of 8.

Some other peculiarities of IP fragmentation:

- No rule exists to prevent a packet from being fragmented into more than the minimum number of packets. Therefore, the above example in Figure 4 could legally be fragmented into 16 fragments.
- The destination device handles the fragment reassembly.
- Once a packet is fragmented, each fragment is handled as a unique and separate packet. Therefore, fragments can arrive at the destination in any order.
- If any fragment is lost (from packet loss typically), the remainder of the fragments are discarded, as there is no information on recreating lost data from the other fragments. The source host must resend the data which will likely be fragmented.
- If a packet hasn't already been fragmented from the start, and fragmentation must occur, a router will process the fragmentation, adding overhead to busy routers.

Recent research has estimated that about 8% of Internet traffic is fragmented [CAIDA02].

Fragmentation is therefore a troubling issue for networks, and one that will remain a problem until new technologies such as IPv6, which disallows fragmentation, predominate.

4.2.1 Study Results

The results of IP fragmentation in this study did not create any significant differences. Indeed, this parameter was tested with configured MTUs of 100, 512 and 768 bytes, the server's performance remained the same regardless of the MTU, even when Avalanche reversed the packet order. After considering the reasons, it became clear that the outbound requests from Avalanche were very small, well under 100 bytes, meaning that fragmentation was not going to occur even with small 100 byte packet sizes. While the inbound responses from the server are indeed greater than 100 bytes, the

server itself does not perform any fragmentation itself—this step typically done by routers. In order to properly test IP fragmentation effects, several testing scenarios could be considered:

- Include a router in the testbed to determine the effects on overall network performance
- Reduce the MTU to VERY small values, such as 5, 10, and 20.
- Configure Avalanche to send large data streams. This could be done by having Avalanche POST large amounts of information to the server. However, the testing of dynamic processing are beyond the current scope of this paper.

This issue warrants further study, and will be done at a later time.

4.3 Link Speeds

We're all familiar with the slow download speeds associated with dial-up modems. Even with the rapid uptake in broadband usage, most people still connect to the Internet with slow dial-up modems [NIELSEN03].

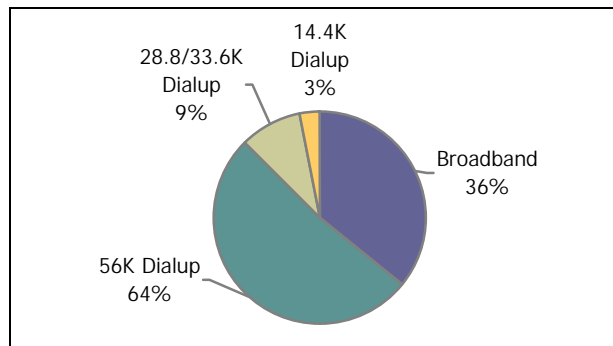


Figure 6: The use of connection speeds on the Internet from Nielsen/Netratings, May 2003.

Figure 6 shows a large 64% majority of Internet users still using dial-up modems. Broadband, including DSL, cable and other faster technologies, is quickly rising in use, but as shown by the fact that 3% of Internet users still use the 10-year-old technology contained in 14.4 kbps modems, the deep market penetration of broadband will most likely take a long time.

Why does link speed matter? Often, people expect slower speeds to create less load on a server because the requests arrive slower, giving servers more time to work on the various requests. However, the opposite is true. As a server receives requests, it queues the incoming data until it has the entire request, then processes it. For slow link speeds, the request takes much longer to arrive completely, causing extra work for the server to wait and receive the request. When the server completes processing the request and is ready to respond, a slow connection speed means that the server must continue to hold on to the response until the slow link transfers the entire message. Network connections must also stay open longer in order to accommodate the slow ingress and egress of data and the similarly slow connection open and closing handshakes.

4.3.1 Study Results

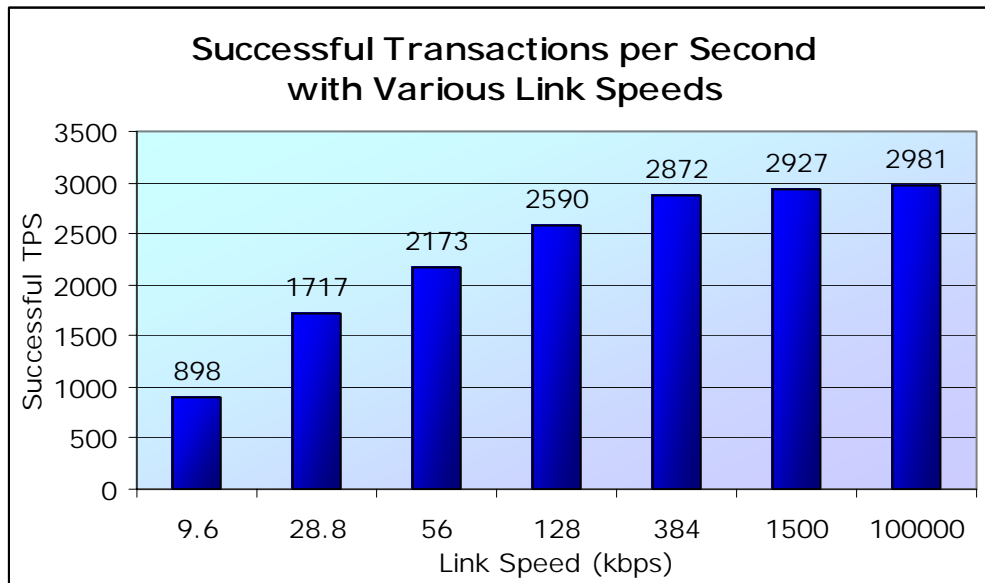


Figure 7: The effect of various link speeds on successful transactions per second.

The result in figure 7 shows how a slow link speed will indeed cause the server to respond more slowly. Indeed the server is having to do more work to manage and serve the slow connections for the users.

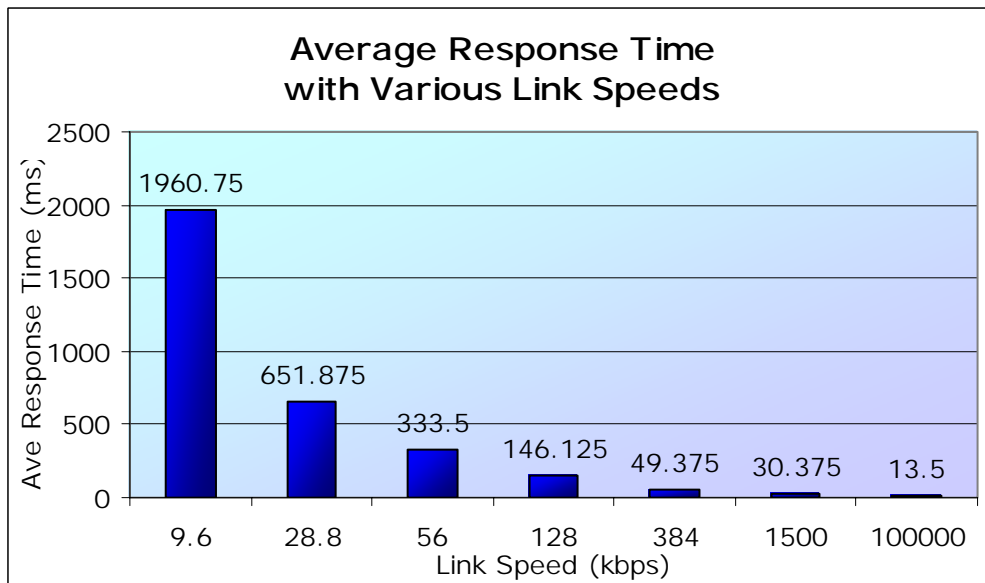


Figure 8: Average response time with various link speeds.

Finally, figure 8 shows the change in average response time as the link speed increases.

5 Conclusion

A vast amount of information and functionality is readily available from the global network called the Internet, and people and businesses are increasingly relying on it for their daily tasks and operations. As a result, the Internet's users continue to push for reliability and performance.

The very things that make the Internet such a valuable network can also make it difficult to scale. Multitudes of users, devices, and interconnections create a complex network with troublesome behaviors stemming from the users, the applications and the network itself. Capacity assessment must capture and simulate the volume and the behaviors to adequately stress network systems to the same degree seen or expected in the production network.

Rigorous capacity assessment serves as the foundation from which a high-performing, reliable Internet can continue to grow. It can help reduce problems and discover bottlenecks, ultimately improving the overall quality of the systems that undergo capacity assessment. The improvements achieved from rigorous capacity assessment also allow the enterprises and organizations that use it to realize oftentimes significant cost savings.

A great many user and network behaviors contribute to real-world capacity assessment. Different behaviors will matter to different networks, but capacity assessments that can incorporate most of these behaviors stand to achieve results with the most reliability and accuracy. In the end, real-world capacity assessment can help make the Internet what it was meant to be: reliable, high-performing, secure, and ultimately, valuable.

6 References

- [BERNERSLEE96] Network Working Group, T Berners-Lee, R Fielding, and H. Frystyk
"RFC1945: Hypertext Transfer Protocol – HTTP/1.0"
<http://www.ietf.org/rfc/rfc1945.txt>, May 1996
- [CAIDA02] University of California at San Diego, C Shannon, D Moore, and K.C. Claffy
"Beyond Folklore: Observations on Fragmented Traffic"
IEEE/ACM Transactions on Networking, 709-720, December 2002.
- [ITR03] AnalogX
"Internet Traffic Report"
<http://www.internettrafficreport.com/>, August 2003
- [LAKSHMANAN99] Yegappan Lakshmanan
"IP Fragmentation: Questions and Answers"
<http://www.geocities.com/SiliconValley/Vista/8672/network/ipfrag.html>,
October 1999
- [NASA98] National Aeronautics and Space Administration, Integrated Service Network
"The Effect of Packet Loss on TCP Application Performance"
<http://nileweb.gsfc.nasa.gov/advtech/packet-loss/study.html>, December 1998
- [STEVENS97] Network Working Group, W. Richard Stevens
"RFC2001: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms"
<http://www.ietf.org/rfc/rfc2001.txt>, January 1997
- [WEBPDA03] Internet.com Webopedia
"Online Dictionary"
<http://www.webopedia.com/>

7 Acknowledgement

I would like to thank **Lucas Yun-Nikolac** for his invaluable assistance in conducting and interpreting the study results.