



www.internet2.edu

What is E2Epi? E2Epi is Internet2's End-to-End Performance Initiative. For more information about us, see: <http://e2epi.internet2.edu/>. There you'll find links to recent projects (like **NDT**) and presentations, as well as other information related to end-to-end performance issues, measurement tools, upcoming events, and related activities.

What is NDT? It is the Network Diagnostic Tool, developed and expanded under several government grants. For more information, see <http://e2epi.internet2.edu/ndt/>.

Try the tool out via one of the publicly accessible servers: <http://e2epi.internet2.edu/ndt/ndt-server-list.html>.

Even Supercomputers Need Help

NDT in Action at SC2004 A Case Study for E2Epi

The annual SC conference attracts a global community of academic, industry, and national laboratory researchers working at the leading edge of computing and networking. Every year researchers bring their demos to the convention floor and use high-speed networks to reach back to their home institutions to satisfy their computer and storage requirements. This year the SC conference brought over a dozen OC-192 links into the Pittsburgh convention center for four days of concentrated high-performance computing.

With a short time to set everything up, and a long path back to a university computer center, it isn't surprising to find some people having problems getting their demo up and running. This year the Internet2 piPEs **Network Diagnostic Tool (NDT)** tool played a small roll in helping one such team solve their demo problems.

It started when a researcher for the Dutch Research Consortium asked "How do you set the TCP send buffer on a Windows XP machine?" NDT developer, Rich Carlson, replied that one can poke around in the Windows registry file or use a GUI tool called DrTcp, available on-line from <http://www.dslreports.com>. (The NDT web page contains a link to this tool.) The GUI allows users to view/change most of the settable TCP parameters that are important to achieving good performance over high-speed networks.

A short time later, the researcher returned and reported that they used the GUI; the GUI indicates that the buffer is set properly, but the test application is still not set so their application doesn't work. After learning that the researchers had spent over an hour already trying to set the TCP buffer in the Window XP laptop, which was the client for their demo, Rich walked over to the booth to see how he could help.

Rich learned that the team was trying to run a visualization application with the server, an SGI computer, located in the Netherlands and the client was the XP laptop. The team had already determined that the network path had a 100 msec round trip time, and they had calculated the buffer space needed to fill this pipe. The application was not performing well, so they had started using a network test application to check the speed of the network. They had determined that the XP was the bottleneck and they were looking for ways to fix this problem.

Rich started by having them bring up a web browser on the laptop and connecting to the NDT server located at the conference center. This would allow them to test the laptop's performance against a known 'good' machine. Since the NDT server was so close, it wasn't surprising that they achieved over 90 Mbps in both directions over the laptop's FastEthernet interface. The good news was that the NDT server also monitors and reports the TCP send and receive buffer size; as a result, they discovered that the laptop had been changed! In less than 30 seconds, the team finally learned that the laptop was not the bottleneck they had thought.

Once NDT had eliminated the PC as the source of the problem, Rich started looking at the remote server. The SGI was accessible, but there was no one at the remote location to run the web-based client. After discussing this with the team, they decided to download and build the command line client on the IRIX operating system. This was a new experience for Rich, as he hadn't done any development work on the SGI. The NDT distribution package was retrieved from the Internet2 web server and copied over to the SGI.

He crossed his fingers as the researcher unpacked and built the NDT command line tool; within 10 minutes, they were able to

start testing from the remote location. The initial test resulted in 17 Mbps in both directions and the team reported that that's what they were getting to the laptop. (An example of the type of data provided is shown below.) However, as we saw with the laptop, the NDT also reported what was limiting the throughput. The NDT report clearly showed that the SGI had a configuration problem – its TCP buffers were set to only 256 Kbytes, which would limit throughput to the 19 – 20 Mbps range. Just what the team was seeing all along!

The next question was “How can we improve this situation?” The team was understandably reluctant to change the parameters on the running network interface, because there was no one in the office on a Sunday evening in Amsterdam. After more discussions, they agreed to experiment using the NDT command line tool. The tool allows the user to set the TCP buffer size for a specific connection using a standard network library routine. While this wouldn't make a permanent change to the SGI, it would allow the team to experiment with the server to ensure that it was capable of fully using the network.

These experiments showed that the SGI was able to achieve over 100 Mbps, on a Gigabit Ethernet network, over the transatlantic network path. Armed with this knowledge, the team was finally convinced that the network wasn't a problem, and if they re-configured the SGI, their application performance would improve.

Thus, the NDT tester provided a valuable service to this team of researchers in a stressful position. Their original assumption, that the laptop PC needed to be tuned, was probably correct, but without the NDT they were unable to verify that the tuning steps had been successful. Within 30 seconds of running an NDT test, the team verified the PC's configuration and could move on to other possible problems – so the NDT reduced the troubleshooting time from 1 hour to 30 seconds.

The NDT also showed that the remote SGI supercomputer was not configured to operate over the transatlantic network path it was being forced to use. Testing in the local environment had failed to show this critical performance problem existed. The NDT was able to clearly identify the problem and allow easy testing of possible solutions. Thus, in a little more than 30 minutes, the NDT tester showed this team of researchers where the real problems were and allowed the system administrators to re-configure the systems to operate properly over this wide area network.

The next day, the researcher reported that the SGI system administrator had reconfigured the default TCP buffer sizes the previous day, but the visualization application still didn't perform up to expectations. The application developer was finally called in and they found a problem with the frame buffer update algorithm. This was quickly fixed and the problem was totally solved, allowing the team to demo their application. (As it turned out, they achieved ~10 Mbps with ~5 frames/sec, which was sufficient; later research in Amsterdam achieved twice that rate, which indicates that the application could be even better tuned.) For more details on the data collected for this problem, see <http://e2epi.internet2.edu/SC04/ndt-sc04.ppt>.

