

# TOOLS FOR APPLICATION PERFORMANCE MANAGEMENT

Surya Chataut, Stan McClellan  
Center for Telecommunication Research and Education (CTER)

Jill Gemmill  
Network Services

University of Alabama at Birmingham  
Birmingham, Alabama

## Abstract

Electronic commerce, Internet-based sales/marketing, and the corresponding need for corporations to re-engineer in order to rapidly deliver goods and services are shifting the dependence of the modern enterprise onto increasingly distributed applications. As a result, the end-to-end management of these applications has evolved from the realm of “simple capacity planning” to the realm of “mission critical business functions”. It has become essential for successful enterprises to monitor the performance of such critical applications, from end-to-end, and from business function to business function. Thus, the terminology and technology of “Application Performance Measurement” (APM) has been tapped by industry analysts to become a multi-billion-dollar market in the next 5 years. The purpose of this paper is to present some representative technologies available in the emerging field of network performance. Along with a brief comparison so some APM products, we include results from specific tests and discuss their interpretations.

## Introduction

The huge growth and commercialization of the Internet in recent years has provided many large corporations and small business, government and private organizations, educational institutions, consumers, and individuals with cause for concern. The Internet has become an integral means for conducting business, engaging in commercial activities, and simply communicating. More and more critical tasks rely on the Internet than ever before. For these and other reasons, the Internet can be viewed as critical infrastructure for a company’s success.

An approach to ensuring successful, application-specific performance of distributed applications is

through the deployment of Performance Monitoring points in the enterprise network. By discovering bottlenecks and understanding the factors related to these problems, network engineers may be able to design better, more reliable systems to allow applications to perform as needed. As a relatively new technology sector, the tools available for APM are widely varied, and some are application-specific. However, all are used to directly measure the Quality of Service (QoS) delivered by mission-critical applications. As such, APM indirectly measures the value of the entire Information Technology infrastructure for an enterprise.

Unfortunately the computer network world has diverse mixtures of devices and technologies that are not controlled by a single entity. Although individual organizations can insure that all of their networking equipment is functioning properly most of the time, proper functioning of the equipment does not mean that the end user or application is getting the required QoS. Data sent from one type of network may end up going through several different networks that use different technologies, affecting their timing at each transition. These events can create considerable variations in the timely delivery of data. All the critical business transaction traffic must traverse the same congested paths and queue at the same congested routers as all other traffic. To address such problems, technologies such as Differentiated Services (DiffServ) have been proposed. DiffServ is a mechanism for classifying traffic flows in IP-based networks. With DiffServ each packet receives a particular forwarding treatment based on the marking in its IP TOS octet (DS CodePoint). Depending on the needs of the traffic classification, DiffServ can provide a form of customizable end-to-end QoS. However even “state of the art” architectures or protocols must be validated in practical situations, and sometimes this validation produces surprising results.

## Background

Modern organizations have only recently realized the importance of network performance. This realization has led companies to develop various tools to help enterprises manage and analyze their network performance more efficiently. There are commercial APM tools that are very costly, and there are free, non-commercial tools. Table 3 lists several of the many tools that are available to perform end-to-end APM for distributed applications and networks. In the table, we list the company and/or product name, the operating system(s) supported, some capabilities of the system, and whether the tool is commercial or non-commercial.

At this point in the evolution of APM technologies, commercial products have significantly more capability to perform various tasks than non-commercial packages. However, commercial products may be expensive. In contrast, non-commercial products are generally free, but may not provide adequate support to manage the spectrum of networking needs.

Performance monitoring software systems have many capabilities that are fundamental for IT managers. Finding problems after they occur is difficult, expensive and time-consuming. Users are rarely tolerant of slowdowns or failures when it impacts their productivity. Traditionally, network managers have approached this issue in a piecemeal fashion. One management tool would outline how a desktop computer was functioning, a second would measure the efficiency of the router, a third would determine various parameters of the server, and a fourth would outline how a database is managing queries. While this approach produces useful insights, it does not provide a comprehensive picture as to how end-to-end performance is impacted. The variance in equipment standards from different vendors obfuscates the process; making it more complicated and time consuming. Tools, which incorporate these disparate sources of information into a single report with an inclusive picture of the network, can be very valuable. Ganymede's *Chariot* and *Pegasus*, and Jyra's *Service Management Architecture* (SMA) are few of the many commercially available software systems that provide the facility to identify the source of network problems. In the next sections, we will examine the characteristics/performance of few of these packages.

### Ganymede: Chariot/Pegasus

Chariot is described as a pre- or post-deployment, ad hoc performance-testing tool. It is used to measure end-to-end network performance, response time and throughput for complex, multiprotocol networks. Chariot software uses *Performance Endpoint* technology. Endpoints are "thin software agents"

installed on computers throughout the network. These agents are used to generate real network traffic. The endpoints use protocol-independent application scripts to emulate application traffic over a network. When a problem is reported, Chariot can be used to emulate the application in question to determine the source of performance problems. Information from such emulation can be used to track down the problem efficiently. Figure 1 shows how Chariot deploys the script to run tests between two points.

In Figure 1, the management station creates a test-script emulating the desired application. This script will be used in running tests between two "endpoints" to analyze the application performance between those endpoints. The test-script is then sent to one of the endpoints, which then communicates with the other endpoint and carries out tests according to the script. The test results from both the endpoints are then sent to the Management Station, which creates reports about the network performance between the two end points.

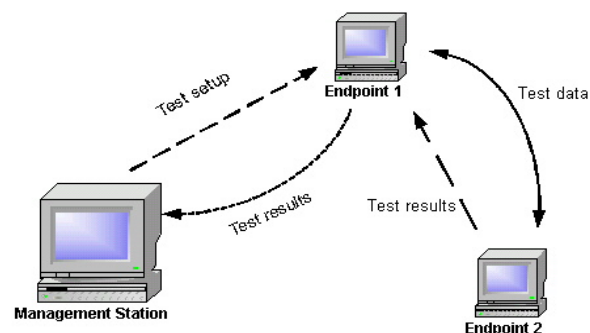
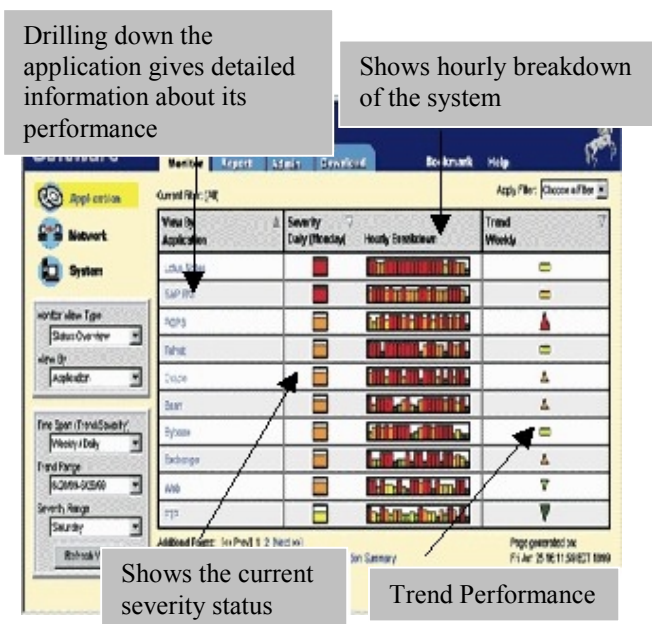


Figure 1. Deployment of a Chariot test scenario.

In contrast with Chariot, Pegasus is a post-deployment performance-monitoring tool that uses some of the same technology as Chariot. Pegasus allows IT managers to see application performance from the user's perspective. It shows the total response time a user is experiencing and then breaks the response time down into client, network, and server components. Sporadic performance problems are unavoidable. Pegasus can be used to identify the source of a problem, and help prioritize which problem to solve first. A very appealing feature that Pegasus provides is benchmarking and monitoring of Service Level Agreement (SLAs). A SLA is a contract between a service provider and its customers that specifies the level of network service that is guaranteed. An SLA can specify bandwidth availability, response times for routine and ad hoc queries, response time for problem resolution (network down, machine failure, etc.). To-date, corporations have been largely unreceptive to SLAs because they are difficult to establish and even more difficult to monitor. However, with sophisticated network monitoring packages becoming available, these

tasks have been simplified. Tools such as Pegasus allow IT managers to craft SLAs in terms that end-users can relate to: response time, throughput, and connectivity. It also gives managers the ability to run tests on a scheduled basis, track performance history and trends, and notify management about performance problems. All of these factors are important in tracking Service Level Agreements.

Figure 2 shows the Pegasus Monitor Console, which provides an interactive overview of active network and passive application information. Using Pegasus, an administrator can generate, view and print a wide range of reports on all network conditions and applications currently being monitored.



**Figure 2. Pegasus Monitor Console**

The Monitor Console contains four primary Web pages: the *Monitor*, *Report*, *Admin*, *Download* pages.

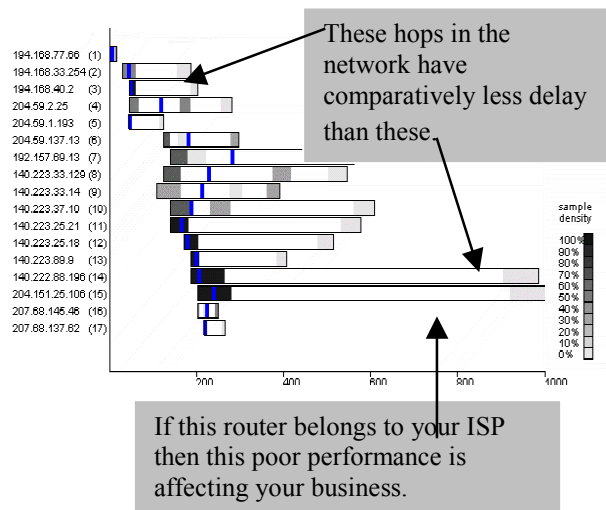
The *Monitor* page is the command center of the Pegasus Console. This page gives the administrator an overview of the current state of the network, applications, and system statistics. Reports containing detailed information about the performance of specific endpoints, network connections, and applications are also available in “drill down” fashion. The *Report page* allows users to view reports, analyze connections, search and select a network connection, and analyze its performance results. The *Admin* tab of the Monitor Console provides a view of the error log and configuration details.

In addition to the use of endpoints that interject arbitrary traffic, Ganymede’s software system is also capable of “Traceroute Analysis”, which depicts

network performance degradation on a hop-by-hop basis. Pegasus executes traceroutes on a scheduled basis, establishing a baseline history of the routes taken. When an exception occurs, the endpoint initiates a traceroute and sends the results to the Pegasus server, which compares the route taken and the hop latencies of the exception traceroute to the baseline traceroute.

**Jyra: Service Level Management (SMA)**

There are numerous other software systems with similar capabilities, and Jyra’s Service Level Management (SLM) is one such package. Shown in Figure 3 is an example of the “Trace Route Chart” report generated by the Jyra SMA, which is similar to Ganymede’s “Traceroute Analysis”.



**Figure 3. Trace Route Chart**

The Jyra trace route report details the individual response times for each hop across a large routed network between two sites. Each bar shown in the report represents the minimum, maximum, and average response time for each hop, as well as the sample density of response times during the monitored period. This assists the network managers in identifying slow links or routers within the network efficiently, and makes them aware of how users’ application response time are affected by it.

**Response Networks: ResponseCenter**

Response Networks’ ResponseCenter is another software system capable of monitoring and diagnosing end-to-end performance from an end user’s perspective. ResponseCenter automatically generates test transactions from lightweight, simulated desktop clients up through the network and into servers, web application, databases, and middleware components – even across the Internet or a service provider network.

Acting as simulated users, the agents measure end-to-end response time, throughput and availability every step of the way, breaking down performance by network, server, application and database processing times.

Despite the similarities in capability between several of the network performance monitoring software systems, there are some baseline differences that are of interest. When an end-user reports a performance problem, the IT manager must determine which component of the network-based system is at fault. There are two approaches to resolving this issue in the context of performance monitoring software. In the first approach, scripts that run on endpoint-software can be deployed to simulate the behavior of the application in question. This simulated application is invoked from the end-user’s premises to establish the nature and location of the problem, thus recreating the problem with information vital for resolving the issue. The second approach does not require creating “scripts”, rather it collects precise usage data in real time from real users, correlates that data with the system and network metrics and creates easy to understand charts. Both the approaches mentioned above have associated pros and cons.

If the software system is designed to collect network/application information only in real time, then various tests may not be available in advance. With pre-written scripts, the application/network performance can be tested before an application is deployed. With real time monitoring, precise usage data from real users is collected and reflects actual network usage conditions. In either approach, this technology provides an approach to managing service levels and making well-informed decisions based on accurate information.

### Test Setup

In this section, we describe specific tests that we conducted using Ganymede’s Pegasus software. The primary focus of our tests was the performance measurement of the network in terms of response time and throughput with large FTPs and UDP streams.

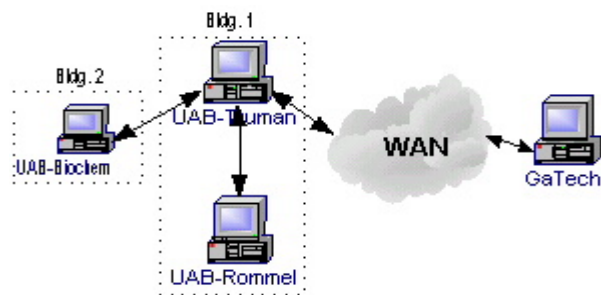


Figure 4. Logical layout

To enable our tests, we installed Ganymede’s Endpoint Software at various locations within the University of Alabama at Birmingham (UAB) network and one at the Georgia Institute of Technology (GaTech). Using these endpoints, we collected throughput and response time data at several specified times. Figure 4 shows the logical layout of the locations used for our tests.

As shown in Figure 4, the tests were carried out between UAB-Truman and UAB-Rommel (a “Local” data path), UAB-Truman and UAB-Biochemistry (a “Campus” data path) and UAB-Truman and GaTech (a “Regional” data path). Table 2 lists the connection type for each path.

Table 2. Connection Type

Location	Connection
• UAB-Truman	• 10Mb shared
• UAB-Rommel	• 10Mb shared
• UAB-Biochemistry	• 100Mb/s X
• GaTech	• 100Mb X

UAB-Truman and UAB-Rommel are located in the same building, and connected to the same 10 Mbps hub. There are other devices that use the same hub as well. UAB-Biochemistry is located across the UAB campus, and GaTech is located in another state.

Using Ganymede’s Pegasus software system, we created various scripts for the intended test. Three FTP and UDP streams with varying payloads were used for our tests, in order to collect data about network performance vs. data size. The FTP sizes used were: 10MB, 100MB and 1GB, and the UDP sizes used were: 100Bps, 100kBps and 1Mbps. The tests ran for a one-month period, and data was collected for all tests. The throughput and response time data collected during this period is analyzed in the following section.

### Results

Without carrying out tests to measure the performance of a network, IT personnel might assume that end-to-end performance between two terminals in the same network would be “better” than the end-to-end performance between two terminals in an inter-network or inter-domain network. Interestingly, the data collected through our “Local”, “Campus”, and “Regional” test scenarios reveals comparative levels of performance which seem to contradict this assumption.

Figure 5 shows 95% confidence interval around the sample mean for the FTP response time for our Local, Campus and Regional connections. From the figure, Regional performance is very high for all FTP tests, regardless of payload size. Not only does the Regional test scenario have uniformly lower response time, but

also it is more stable. On the other hand, Campus performance seems to be very poor. In fact, the Campus performance is as much as 3 times worse than the Regional or the Local, which depicts that there are some significant bottleneck(s) in portions of the network.

Additionally, the large confidence interval in the Local test shows that performance is not consistent. Many reasons for this result are possible, and more research is necessary to determine plausible explanations. The UAB-Truman and UAB-Rommel computers are used for web-based multimedia servers in a relatively busy laboratory where several other machines are connected to the same 10 Mbps hub. As such, certain times of the day have high local traffic profiles, which may contribute to these results.

Figure 6 shows the performance pattern for FTP throughput, which is related to response time. Before examining “real world” data from the Pegasus tests, our intuitive expectations for the throughput of these connections would have predicted a very different performance pattern. For example, we were expecting that the “Local” and “Campus” connections would have much higher throughput than the “Regional” connection.

Correspondingly we observed the same pattern with the UDP streaming tests. Figure 7 and Figure 8 show the confidence interval for Local, Campus and Regional UDP response time and UDP throughput respectively. As with the FTP tests, Local and Regional connections outperformed Campus connections, which display very wide confidence interval (for both response time and throughput). This may reveal some issues with local traffic patterns through the UAB Campus network that could be addressed in order to improve local performance.

## Conclusion

Results of the tests briefly discussed here make it clear that proper function of network equipment “most of the time” does not translate to end-to-end performance benefits. One would assume that performance between two computers on a shared hub (assuming relatively low traffic) would perform far better than inter-network or inter-domain network. However, as can be observed from the test results in Figures 5-8, this is not necessarily the case. This result highlights the importance of network/application performance management tools. In the recent past, end-to-end network performance was not part of the IT manager’s priority list due to the simplicity of the applications and the relatively low importance of transactions over the Internet. However, the increasing complexity of distributed, Internet-based, mission-critical applications

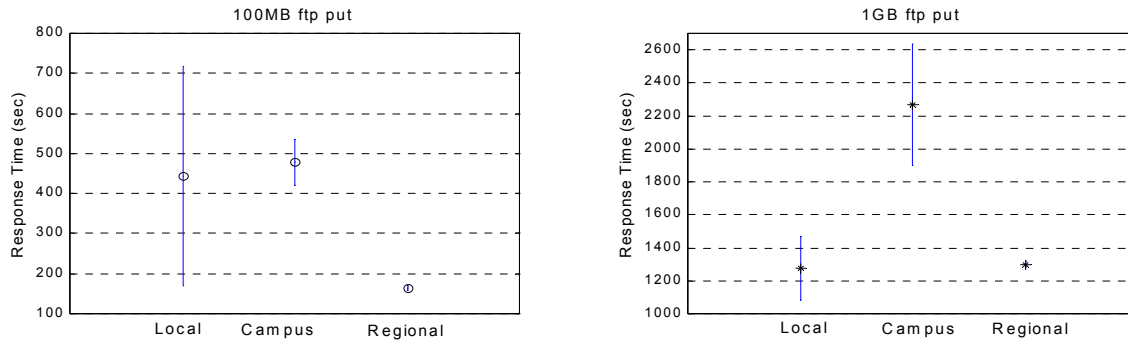
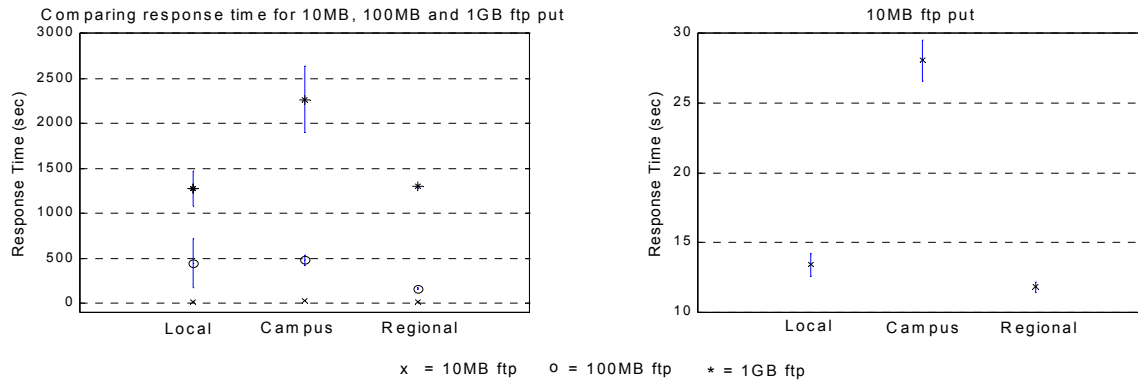
seems to indicate that end-to-end monitoring of performance has become an essential part of network management. As an important aspect of gathering and interpreting performance data, tools such as Ganymede’s Chariot/Pegasus and similar software packages may be critical infrastructure for network administrators.

## Reference

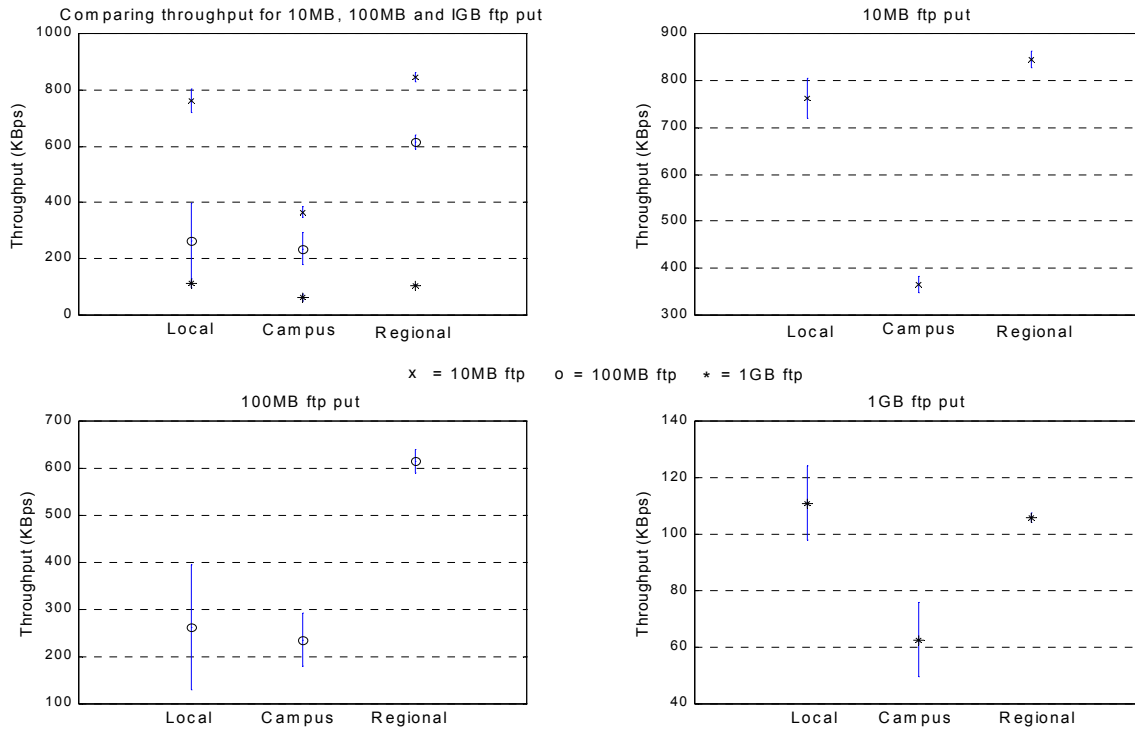
1. <http://www.ganymede.com/>
2. <http://www.jyra.com/>
3. <http://www.nlanr.net/>
4. <http://www.inversenet.com/>
5. <http://www.ins.com/>
6. <http://www.slac.stanford.edu/>
7. <http://www.appliant.com/>
8. <http://www-iepm.slac.stanford.edu/>
9. <http://dast.nlanr.net/>
10. <http://www.responsenetwork.com/>

**Table 3. Commercial and non-commercial products and their features**

Company/•Product Shareware, Freeware or Research	Supported OS	Capability
<p><b>1. NLANR</b></p> <ul style="list-style-type: none"> <li>• Active Measurement Program (AMP)</li> <li>➤ Research Group</li> </ul> <ul style="list-style-type: none"> <li>• Iperf</li> <li>➤ Free</li> </ul>	<ul style="list-style-type: none"> <li>• FreeBSD</li> </ul> <ul style="list-style-type: none"> <li>• Linux</li> <li>• SGI IRIX</li> <li>• HP-UX</li> <li>• Solaris</li> <li>• AIX</li> <li>• Cray UNISCO</li> </ul>	<p>Currently three types of measurement are available:</p> <ul style="list-style-type: none"> <li>• Round trip times (RTT), loss and topology</li> </ul> <p>These tests are continuously run on all the AMP monitors to all the others. The data from these monitors is available through web interface in various forms.</p> <p>TCP Feature</p> <ul style="list-style-type: none"> <li>• Measures TCP bandwidth</li> <li>• Report MSS/MTU size and observed read sizes.</li> <li>• Support for TCP window size via socket buffer.</li> </ul> <p>UDP Feature</p> <ul style="list-style-type: none"> <li>• Create UDP streams of specified size.</li> <li>• Measure packet loss and delay jitter.</li> <li>• Multicast capable</li> </ul> <p>Tests can run for a specified period of time and not only one instance.</p>
<p><b>2. Ganymede</b></p> <ul style="list-style-type: none"> <li>• Chariot</li> <li>• Pegasus</li> <li>➤ Shareware</li> </ul> <ul style="list-style-type: none"> <li>• Qcheck</li> <li>➤ Freeware</li> </ul>	<ul style="list-style-type: none"> <li>• Windows 3.1, 95, 98, NT X86, and NT Alpha</li> <li>• Novell NetWare</li> <li>• IBM AIX, OS/2, and MVS</li> <li>• Compaq True64 Unix</li> <li>• HP-UX</li> <li>• Linux x86 and MIPS</li> <li>• SGI IRIX</li> <li>• SCO UnixWare</li> <li>• Sun Solaris Sparc and x86</li> </ul> <p>Qcheck works on all the above OS.</p>	<p>Chariot capabilities:</p> <ul style="list-style-type: none"> <li>• Evaluate the performance and capacity of network products and troubleshoot network performance problems.</li> <li>• Stress test network device and measure the performance impact of network changes.</li> <li>• Measure the true end-to-end response time and throughput of complex, multiprotocol networks.</li> <li>• Separate network performance and network performance in the network.</li> </ul> <p>Pegasus capabilities:</p> <ul style="list-style-type: none"> <li>• Monitor end-user performance.</li> <li>• Solve performance problems before users experience them.</li> <li>• Benchmark and monitor Service Level Agreements (SLAs).</li> <li>• Analyze history.</li> <li>• Track performance trends, and feed them back into the planning process.</li> </ul> <p>Qcheck can be used to test network response time, throughput, streaming test (network's ability to support multimedia traffic) and server performance .</p>
<p><b>3. Jyra</b></p> <ul style="list-style-type: none"> <li>• Service Management Architecture (SMA)</li> <li>○ Mid level Manager</li> <li>○ Service Level Monitor</li> <li>○ Client Service Level Monitor</li> </ul>	<ul style="list-style-type: none"> <li>• Windows NT 4.0</li> <li>• Widows NT 3.5.1</li> <li>• Windows 95</li> <li>• Solaris</li> <li>• Supports industry standards for SNMPv1 &amp; SNMPv2, RMON 1&amp;2 MIB, MIB II and Java Agents.</li> </ul>	<p>Jyra offers packaged software solutions for monitoring mission critical services. These solutions are broadly categorized into three sections:</p> <ul style="list-style-type: none"> <li>• Corporate – companies who have mission critical dependence on their own, internal IT systems or infrastructure.</li> <li>• dot.Com companies – companies who derive a significant portion of their revenue through the internet.</li> <li>• Service Providers – telecommunications and hosting companies who provide connectivity and advanced services such as mobile wireless voice/data services, managed network services, network outsourcing, application and web hosting.</li> </ul> <p>All of these software packages help IT managers to improve the application performance and discover bottlenecks in the network for an IT infrastructure.</p>
<p><b>4. Appliant</b></p> <ul style="list-style-type: none"> <li>• Lateral Line Network</li> </ul>	<p>Initially, Lateral Line Network provides Web Server monitors for:</p> <ul style="list-style-type: none"> <li>• Netscape Enterprise on Sun Solaris</li> <li>• Microsoft IIS on NT</li> </ul>	<p>Lateral Line Network offers a broad array of IT services for e-business solutions:</p> <ul style="list-style-type: none"> <li>• State-of-the-art service level management with 24x7 alerting.</li> <li>• Real time fault detection and troubleshooting.</li> <li>• Usage analysis to improve site design and navigation.</li> <li>• Trend analysis of site performance, availability and usage over time.</li> </ul>
<p><b>5. Response Networks, Inc</b></p> <ul style="list-style-type: none"> <li>• ResponseCenter</li> <li>➤ Shareware</li> </ul>	<ul style="list-style-type: none"> <li>• Windows NT 4.0</li> <li>• Windows 95</li> </ul>	<p>ResponseCenter provides the following service:</p> <ul style="list-style-type: none"> <li>• Rapidly understand where e-transaction performance problems reside.</li> <li>• Get an early warning of application brownouts or service outages.</li> <li>• Baseline historical e-transaction performance.</li> <li>• Manage end user service levels.</li> </ul>



**Figure 5. Confidence Intervals for FTP Response Time**



**Figure 6. Confidence Interval for FTP Throughput**

Comparing response time for 100Bps, 100KBps and 1MBps UDP stream

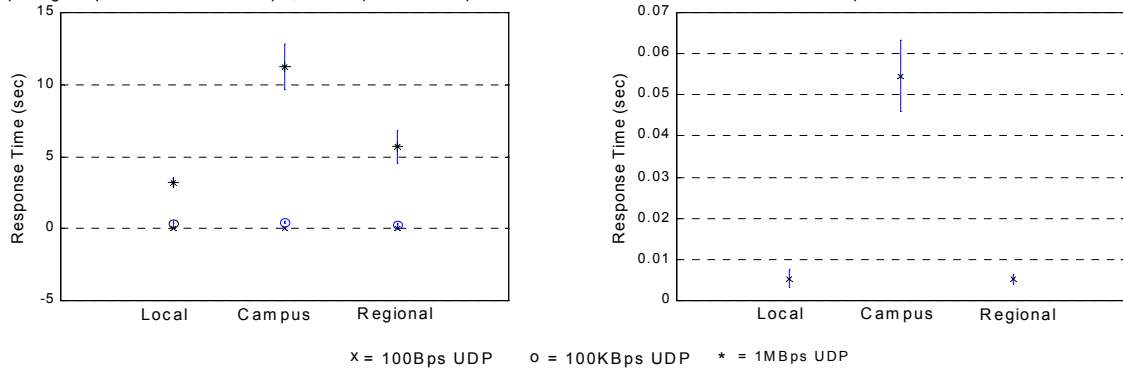


Figure 7. UDP Response Time (Mean Vs. Variance Plot)

Comparing throughput for 100Bps, 100KBps and 1MBps UDP stream

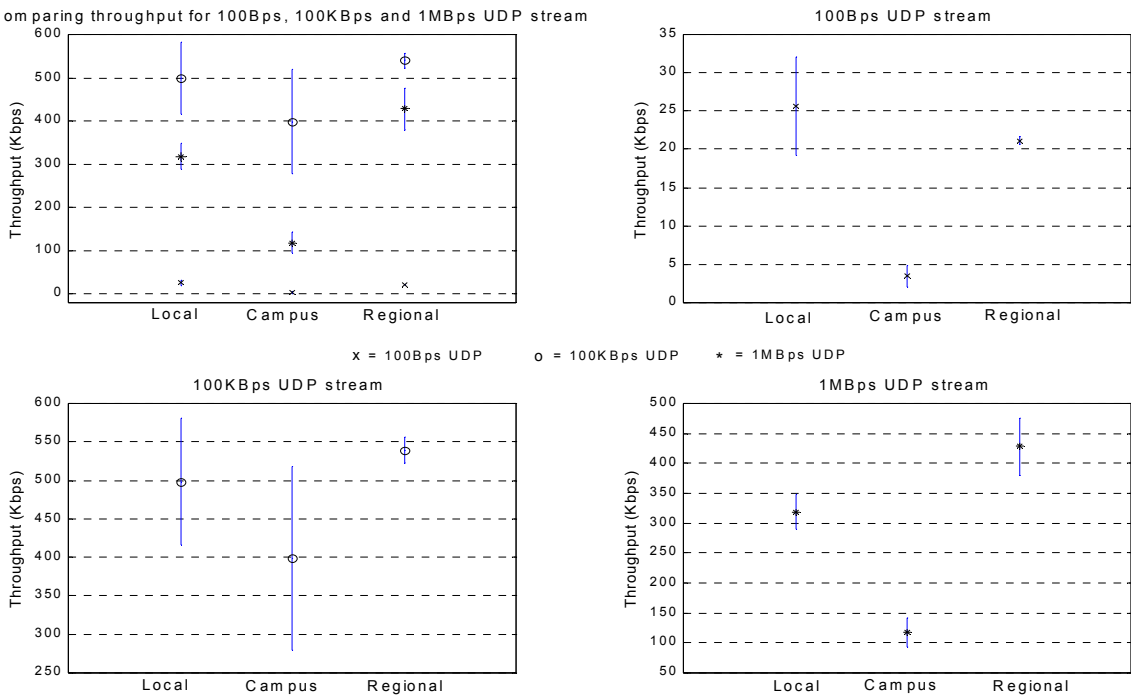


Figure 8. UDP Throughput (Mean Vs. Variance Plot)