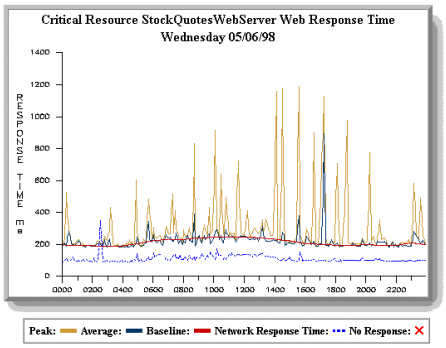
Is it the Network, the Firewall or the Application?



NAME	IP ADDRESS
StockQuotesWebServer	134.141.111.111

White Paper by

Atul Garg
Ronald Schmidt

ProactiveNet

2975 Bowers Avenue, Suite 300
Santa Clara, CA 95051-0955
(408) 450-7700
Fax: (408) 748-6947
www.ProactiveNet.com

Form SF298 Citation Data

Report Date *("DD MON YYYY")* 08011999	**Report Type** N/A	**Dates Covered (from... to)** *("DD MON YYYY")*

Title and Subtitle Is it the Network, the Firewall or the Application?	**Contract or Grant Number**
	Program Element Number
Authors	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) ProactiveNet 2975 Bowers Avenue, Suite 300 Santa Clara, CA 95051-0955	**Performing Organization Number(s)**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Monitoring Agency Acronym**
	Monitoring Agency Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract	
Subject Terms "IATAC COLLECTION"	
Document Classification unclassified	**Classification of SF298** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** unlimited
Number of Pages 11	

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		White Paper

4. TITLE AND SUBTITLE
Is it the Network, the Firewall or the Application

5. FUNDING NUMBERS

6. AUTHOR(S)
Atul Garg, Ronal Schmidt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

IATAC
Information Assurance Technology Analysis
Center
3190 Fairview Park Drive
Falls Church VA 22042

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Technical Information Center
DTIC-IA
8725 John J. Kingman Rd, Suite 944
Ft. Belvoir, VA 22060

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE
A

13. ABSTRACT ***(Maximum 200 Words)***

This White Paper entitled "Is it the Network, the Firewall, or the Application" was written by Atul Garg and Ronal Schmidt. In today's I.T. environment, the business relies on several critical application services. More and more of these applications are distributed client-server applications that rely on an increasingly switched network with a significant number of users accessing these application services via firewalls over the Wide Area Network (WAN). These application services can be broken into packaged or custom applications. Another view of applications is whether they are infrastructure or end-user applications.

14. SUBJECT TERMS
INFOSEC, Firewall

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	UNCLASSIFIED	UNCLASSIFIED	None

Introduction

In today's I.T. environment, the business relies on several critical application services. More and more of these applications are distributed client-server applications that rely on an increasingly switched network with a significant number of users accessing these application services via firewalls over the Wide Area Network (WAN). These application services can be broken into packaged or custom applications. Another view of applications is whether they are infrastructure or end-user applications.

	Package	Custom
End-User Business Applications	• SAP • PeopleSoft • Vantive • Baan •	
Infrastructure Applications	• Web • E-mail • dns • Security • Remote access	

Infrastructure applications. This category includes packaged applications such as email, firewalls, dns, ftp, nfs, etc. These applications are part of the corporate computing infrastructure. In addition, they also include custom infrastructure applications that perform such functions as factory control, fraud detection, etc.

End-user applications. These applications have several end users who are either in-house employees, customers or partners. They include packaged ERP applications from companies like SAP or PeopleSoft; packaged customer service applications such as Clarify and Vantive; e-commerce applications; etc.

If one of these client-server "network-dependent" applications goes down, it can bring critical services to their knees and dramatically affect the business. Because they tend to be host-centric, most traditional application tools do an inadequate job of monitoring the performance of these "network dependent" client-server applications.

Some of the unique challenges in monitoring these distributed applications include understanding the availability and response time of transactions from an end user perspective as well as being able to quickly isolate performance problems to the network, firewall or the application/server. The latter eliminates time-consuming finger pointing between the network department, the security department and other parts of IT. Further, correlating network-caused application degradation to "abnormalities" in the network or firewall assists IT departments in rapidly isolating the cause of application service degradation and fixing it.

Net Transaction Management™ for "Network Dependent" Applications
To do an adequate job of application performance management for "network dependent" applications, one has to understand how application transactions is performing from an end-users perspective and be able to assist in trouble shooting application degradation problems that are caused by performance bottlenecks. The end-user perspective is particularly important for end-user applications while a perspective from a single monitoring server is adequate for most infrastructure applications.

As such, the key measures of an application's performance are its transaction response time and availability as perceived by end-users. Once users start to experience degraded transaction performance, answering one or all of the following high level troubleshooting questions adds significant value to the network department, system department or both:

- Is it the network, the firewall or the application? (of interest to the network, security and system departments)
  - If the network, where in the network? (of interest to the network department)
  - If the firewall, where in the firewall? (of interest to the security department)
  - If the application, where in the application / server? (of interest to the system department)

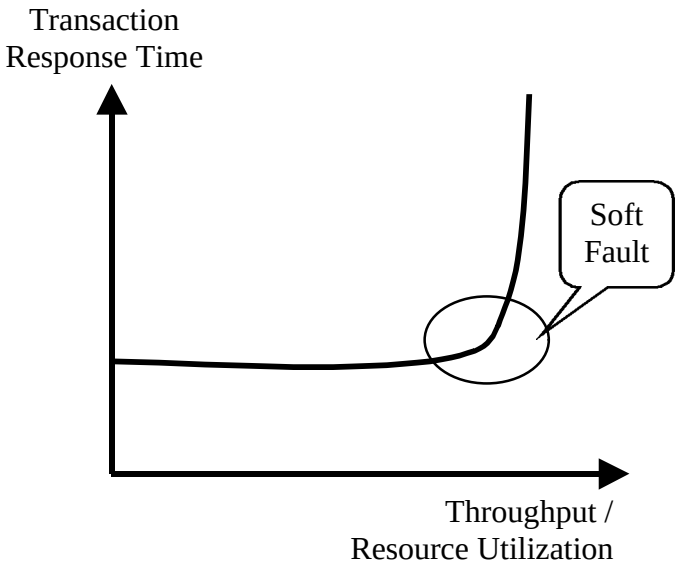


Figure 1

In adding value in the troubleshooting domain, it is important to understand the relationship between application transaction response time and resource utilization across the various resource pools on which that the application depends. As a resource starts to bottleneck – or "soft fault" - application transaction response time sharply degrades. Understanding in real time which resource is bottlenecking, or is about to bottleneck, can prevent application transaction degradation and rapidly restore service levels.

Measuring Application Transaction Response Time (Y-axis)

There are several approaches to measuring application response time. Broadly, they fall into four categories:

- Client-based "active" ghost transactions -- a client executes periodic active ghost application transactions against various application servers on the network and records the response times. Ghost transactions are representative of and specific to the application. Examples include measuring the time it takes to download a home page for the http application or performing an actual name lookup for the dns application Typically there is only one such active agent per key location on the network.

- Client-based "passive" agent measuring -- client workstations are equipped with agent software that clocks response times. These agents are light weight and clock actual application response times as users use the application. Although it is not required, this approach works best if all user desktops are instrumented with the light weight agent.

- Point-to-point packet inspection -- packets are monitored by probes as they travel between network points. The probes include smart software that measures response time between request and reply to application-specific packets.

- Application response time measurement (ARM) -- ARM is a set of application program interfaces that report performance data back to a management application. By using the APIs, an application can leave a trail of its activity and compliant software products can then determine the specific path taken by each request to get a read on response time.

Approach	**Pros**	**Cons**
Client-based "active" Ghost Transactions	• Active client can monitor several applications across the network and across several segments. • Lower cost of ownership - does not require maintenance of complex agent software on every desktop or probes. One per key network location suffices. • Periodic active ghost transactions lend themselves better to statistical analysis.	• Introduces extra active client transactions on the application server.

	• Isolates response time degradation to the network or the application/server. • Captures statistically representative users' perspectives. • Can align easier with SLA.	
Client-based "passive" agent	• Sees actual performance from a user perspective. • Can align easier with SLAs. • Isolates response time degradation to the network or the application / server. • Passive - does not introduce any transactions on the server.	• Higher cost of ownership - requires agents on several client stations. • Does not lend itself easily to statistical analysis – no data points when instrumented users are not active.
Packet Inspections	• Least intrusive of the approaches. • Attempts to re-construct user side of experience. • Greater detail on utilization by application.	• Complex and expensive -- requires one probe per critical segment. • Re-constructed user side experience – weaker credibility with System department when resolving finger pointing issues. • Cannot separate network from application. • Cannot monitor availability easily.
ARM	• Standards based	• Not many packaged applications support ARM standard

The first three approaches are all viable. The challenge with the ARM approach is that standards take a long time to take hold and consequently a real solution using ARM is still some time away.

While the first three approaches are all good, there are, however, some pragmatic tradeoffs. Probe-based packet-inspection is an old generation architecture that runs counter to the trend of switched networks and virtual private networks (VPNs). Ideally, each critical segment requires one dedicated probe, even though probe vendors offer excellent cost-saving recommendations on how to minimize the number of probes by placing them at strategic locations.

The other challenge with the probe approach is to get a true read on end-user response time. While probe vendors offer some clever technology that uses packet inspections to determine unique transaction times, the technology is less than perfect. Further, in a

situation where the data is going to be used as proof that the problem is in the network or the application / server, packet inspection technology is less intuitive and tends to have less credibility with the System department. The biggest advantage of the probe is a measure of utilization by application in the segment where it is installed.

Of the two client-based agent approaches, the passive agent requires more agents to be installed and is more suited for the System department that controls the desktops where these agents need to installed and maintained. The overhead associated with maintaining several hundred desktop agents may be substantial and impractical for the network department.

Vendors of this approach suggest that only select high-volume user desktops be instrumented to reduce the number of passive agents. The challenge there is that the system gathers no data when these "high-volume users" are not using the application. Consequently, the statistical accuracy of the data is suspect. The biggest advantage of the passive agent approach is that it captures the actual end-user experience of the application service rather than extrapolating performance from statistical samples as the active agent does.

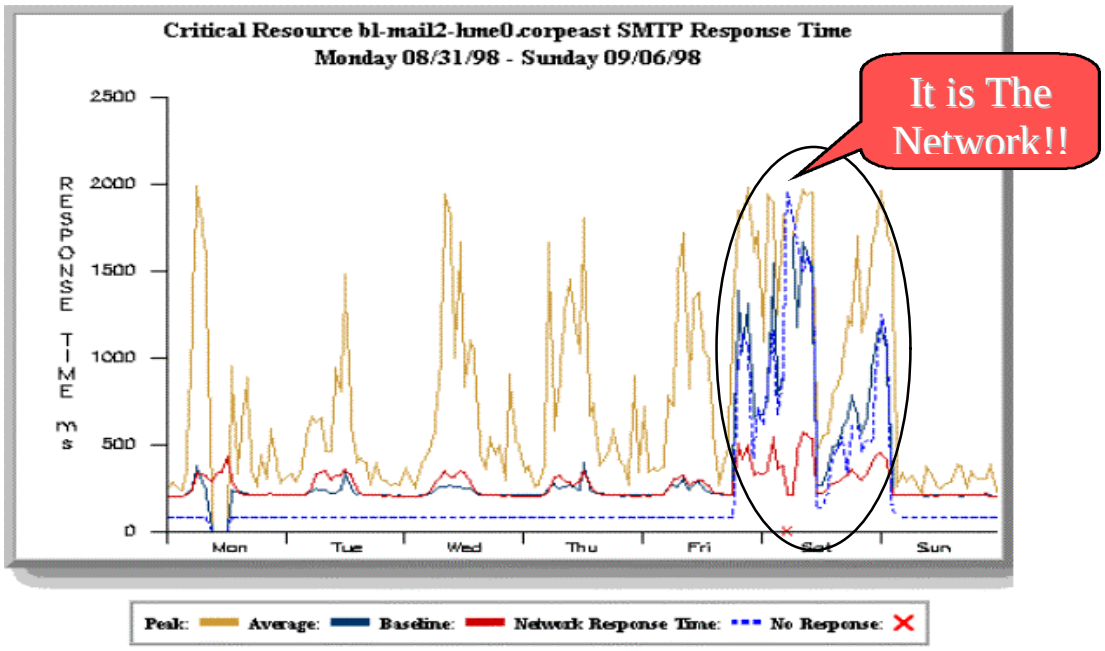
The active agent captures the end user experience using statistical techniques rather than measuring actual end user response time. The advantage of this approach over the passive agent is that it requires substantially fewer agents – a number that a network manager can control and realistically deploy.

Typically there is one active agent deployed for every major location in the network, as opposed to one per desktop or several "high-volume users" per major location, as is the case with the passive approach. Further, the active agent approach statistically captures data seven days a week, 24 hours a day.

One disadvantage of the active agent is that it injects some traffic into the system via its active ghost transactions. This traffic will typically be fairly minimal and most vendors allow network administrators to control the rate of active ghost transactions.

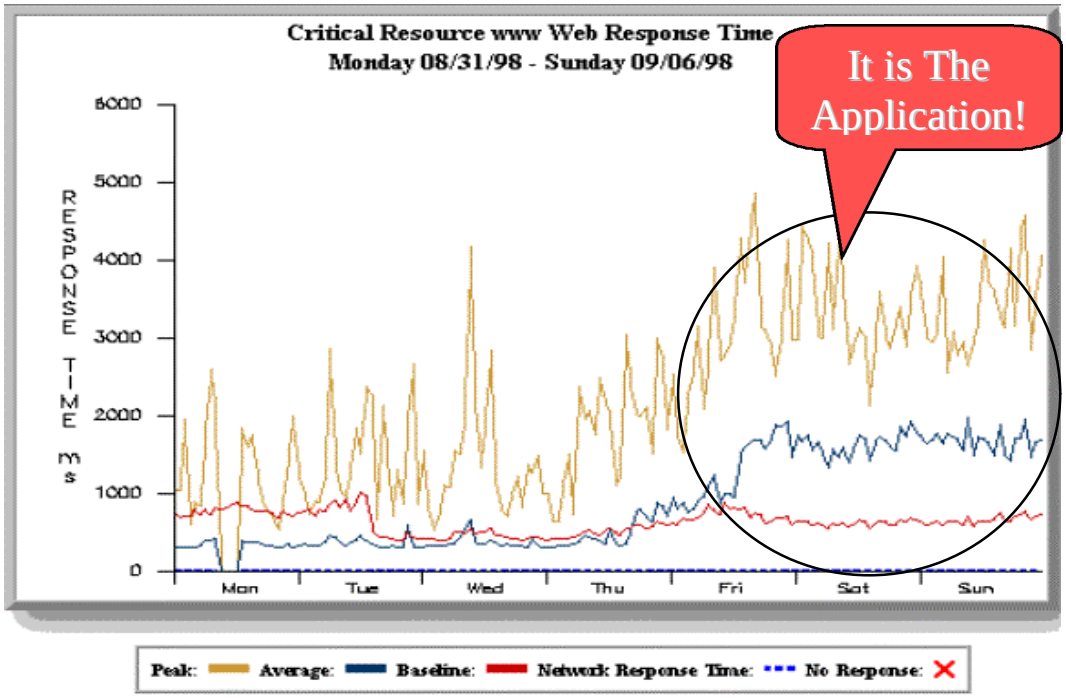
Both the active agent and passive agent approaches do an intuitive job of separating the network from the application. In most cases, vendors use a combination of network PINGs and TCP port connects to construct the network portion of the application response time. The packet inspection probe approach does not do an adequate job of separating the network from the application. The approach is less intuitive in this area.

Figure 2 shows an example where ProactiveNet Watch isolates the cause of a slow down in a mail server to slow response in the network. Notice, for the latter part of Friday and most of Saturday, the network response time (dotted line) has degraded significantly. Likewise Figure 3 shows a slow down in a web response time due to the application/server. Note the network response time (dotted line) is flat under 50 ms.



NAME	IP ADDRESS
bl-mail2-hme0.corpeast	132.245.135.83

Figure 2



NAME	IP ADDRESS
www	134.177.3.28

Figure 3

Real Time, Intelligent Baselining -- Linking Application Transaction Response (Y-axis) to Resource Utilization (X-axis)
The ultimate challenge in application performance management is linking application transaction response time to the utilization / throughput of the resources the application relies on to deliver service. Network resources can be monitored via snmp and server resources via either snmp or an agent. By doing this users will be able to pin point the probable cause of transaction service level degradation as well as predict service problems.

This is still relatively new territory for most vendors. The biggest challenge here is to deal with the mountain of data generated in monitoring all the resources. One effective technique involves the use of hourly baselines as a means of detecting abnormalities. Network administrators then use time and topology-based knowledge to link these abnormalities together to determine the probable cause of the disruption. ProactiveNet Watch technology pushes the state-of-the-art in this area.

Net Transaction Management – Around-the-Clock, Rapid Response Transactions
The core ProactiveNet technology is Net Transaction Management software that includes real time data collection, analysis, alarm notification and reporting. Network and server data is collected using snmp or light weight agents.

Single Server Architecture

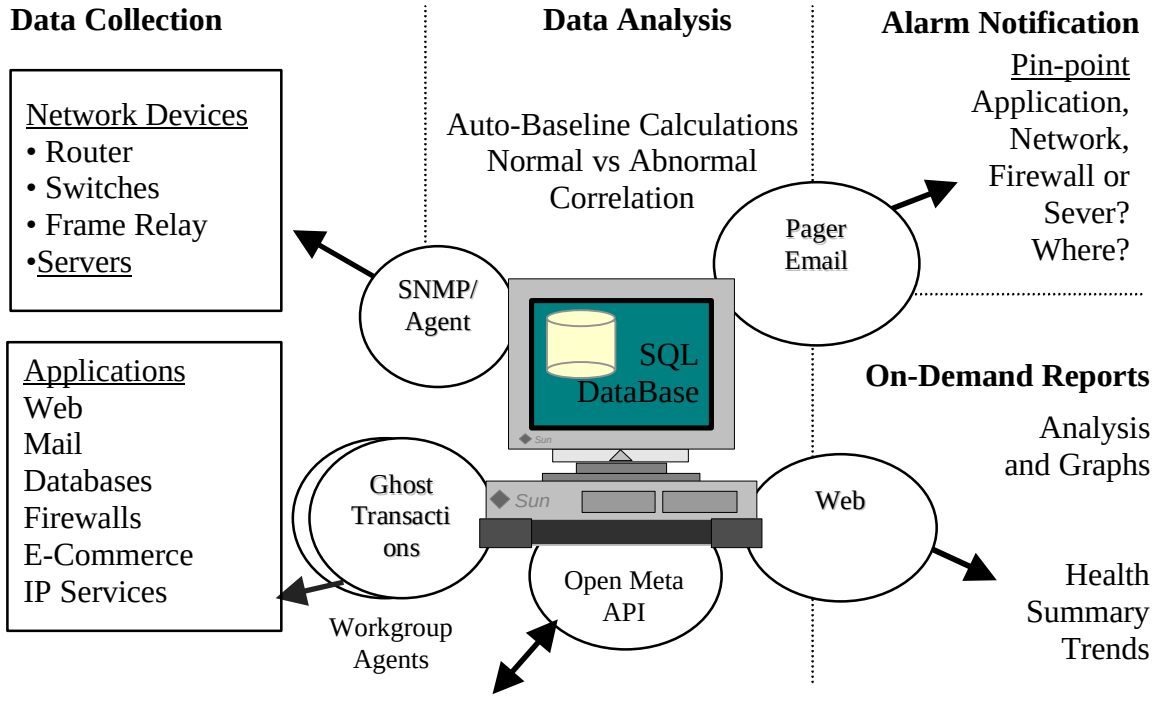


Figure 4

Application end-user response time data is collected using either an active or passive agent or probe based approach. Firewall performance data is collected using vendor specific APIs. In addition, custom application or other data can be fed into the system using ProactiveNet's Meta API (Figure 4).

ProactiveNet Watch uses a patented intelligent baselining technique called Proactive Signatures™ to create hourly baselines of the environment. These baselines are continuously evolving and grow and track with the environment. Proactive Signatures allow administrators to understand what is "normal" in their environment for any hour of day and day of week. ProactiveNet Watch uses statistical techniques along with the baseline to identify all significant above and below baseline abnormalities (Figure 5) and delivers Proactive Alarms™ via e-mail and pager.

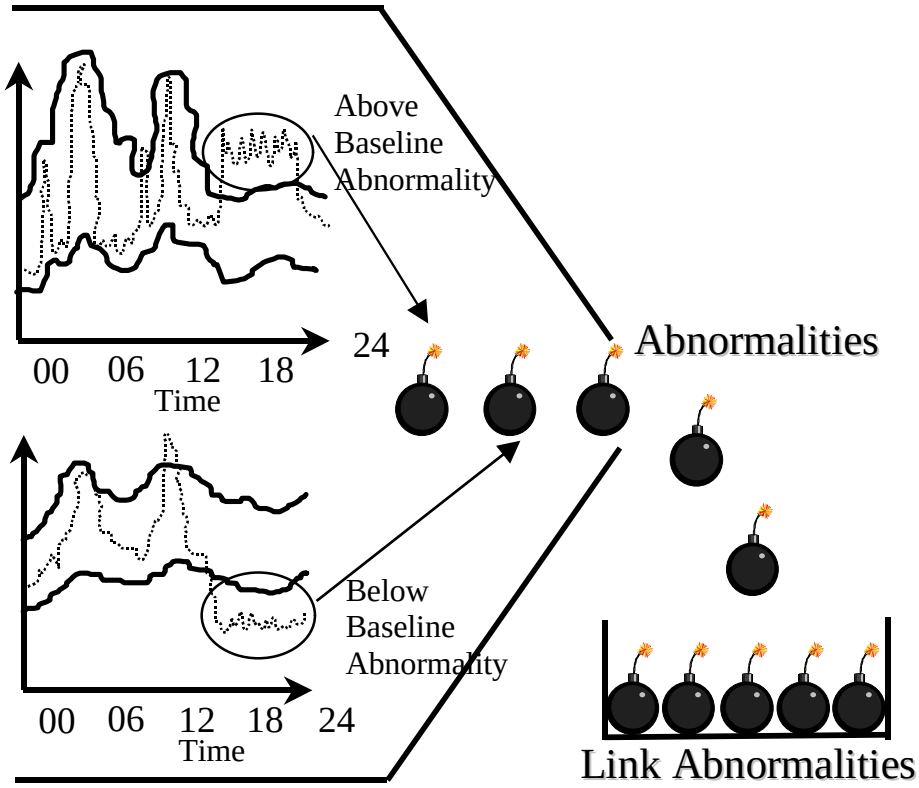


Figure 5

In another words, the system tracks any abnormalities with resource utilization and errors associated with all the resources being monitored that the service depends on for on-time delivery. It then uses time and topology based knowledge to assist the administrator in identifying which resource (network, firewall, server or application? where?) is bottlenecking and causing a service level problem.

About the Authors

Atul Garg, CTO, ProactiveNet, Inc.
Garg has over 18 years of experience in networking and network management at TCSI, Bay Networks and Hewlett-Packard. While at Bay, he developed the long term vision and architecture that led to the development of intelligent network trending and monitoring applications. Prior to Bay, he served as a project manager at Hewlett-Packard, where he initiated and led the development of key components of HP OpenView.

Ronald Schmidt, Board Member, ProactiveNet, Inc.
Ronald Schmidt is a member of the Proactive Networks' Board of Directors. He recently joined Lucent to start up Bell Labs Silicon Valley Research Laboratory in Palo Alto, California. Prior to that he was co-founder and CTO of SynOptics Communications and later Bay Networks.