

Application Management

A White Paper featuring the ProIT-Next Generation Network Management Platform

“Being the most plug & play network management product in its class, ProIT offers an almost immediate return on investment (ROI), accelerated productivity, and scalability to the demands of the enterprise.”

Introducing
ProIT Next Generation
Network Management Platform

1 Executive Summary

Applications are the core value of IT. We don't purchase servers, routers, and related infrastructure simply because of the brand name, features, or specifications. We buy these components to run our business applications reliably and at service levels that guarantee the quality of our customers' experience. Today's applications are more capable than ever, requiring near-perfect uptime and guaranteed performance levels. To deliver on these goals, the infrastructure needed has grown in scope and complexity without a good way to manage it...until now.

Applications now Services

A new lexicon has emerged to describe what used to be called "applications." No longer is an application merely "client-server" or "thin-client." Now, "applications" are referred to as *services*, because this better describes an end-to-end delivery across numerous hardware and software components that typically comprise today's business processes.

As an example, even e-mail should be considered a service. Consider the essential components of a typical corporate e-mail system:

- End-user PC e-mail client such as MS Outlook or Lotus Notes,
- User location's LAN gear, including hubs, switches, and wiring,
- WAN or Internet connection, including WAN routers and service providers,
- Corporate data center network gear, including firewall, VPN, and router devices,
- E-mail servers, gateways, spam and content filters.

A problem or fault in any one of the above components can cause an unacceptable service interruption for the corporate e-mail system. To proactively manage the email system, a robust set of tools must be put into service to assure the components are working together properly and to detect problems so they can be quickly remedied.

Service Component Monitoring

Using our e-mail example above, it becomes clear that the service management tool needs to be able to monitor each of the listed devices and components. But monitoring all of the individual pieces is just part of the solution. Let's call this *service component monitoring (SCM)*. By definition, SCM provides an alert if there is a problem with any individual component, a blessing for an IT staff. Where SCM falls short however, is that it does not provide a view of the end-user experience and may not provide a means to pinpoint a performance bottleneck across an end-to-end service.

Service Components

To properly monitor our e-mail service we need the following service components:

- Device availability for the numerous network devices (ping)
- Port availability between each device (TCP or UDP port monitoring. e.g., SMTP port 25)
- Application service availability (service / process monitoring on the e-mail server)
- Application performance metrics (e.g., WMI counters in Windows 2000)
- Application event logs (e.g., Windows 2000 Application Event Log)
- SNMP statistics from routers, switches, etc. (RMON, MIB2 data, e.g., bandwidth utilization)

- SNMP traps from devices (alerts)
- Device log files (e.g., syslogs on routers, switches, firewalls)

Service Level Monitoring

Service Level Monitoring (SLM) enhances SCM by enabling the measurement of service performance by comparison to agreed-upon baselines or even a true service level agreement. This technology is available today, and it is truly beyond the initial hype. Application vendors offer two types of SLMs: User Transaction Monitoring (UTM) and Service Level Agreement Monitoring (SLAM). They are distinctly different and should be clearly understood.

User Transaction Monitoring

UTM means that an SLM application will actually create authentic or “synthetic” user transactions, then monitor and measure them against acceptable thresholds. In our e-mail example, UTM means:

- Creating an e-mail packet
- Sending the e-mail via the corporate mail server over the network
- Accepting the e-mail at a destination point
- Assuring the e-mail is error free and was delivered within acceptable time
- Performing a likewise transaction for incoming e-mail

Another common example of UTM is for web-based applications, such as an online catalog or shopping cart. A typical UTM web-transaction might include several steps:

- Logging into a secure website
- Entering a search term in an input field
- Examining results of the search to assure success
- Select a catalog item
- Complete a shopping cart transaction

UTM would perform this transaction at specified intervals and be able to chart transaction time performance as well as provide alerts if any step fails or if the response time does not meet certain requirements.

SLA Monitoring & Reporting

While UTM incorporates SLA monitoring, SLA monitoring spans both UTM and SCM. For example, virtually all network management platforms (NMP) provide the standard five severity levels for events: Critical, Major, Minor, Warning, and Information. The problem with the “Five-Level” system is that it seems to be missing the category “SLA affecting.” Just because an event is marked Critical, does not necessarily mean an event is SLA-affecting. In most shops, Critical means “needs human attention right away.” A next generation NMP enables you to specify that an event is SLA-affecting. This has far reaching implications, most notably the ability to report on true SLA statistics. The adage “you cannot improve what you cannot measure” certainly applies.

Most NMP products provide more simplistic availability reporting mechanisms, often claimed to be SLA reports. This is generally far from true SLA reporting. Availability reporting is typically limited to service components (see SCM above) but not an overall end-to-end service availability or true SLA measurement.

ProIT SLA Monitoring provides a breakthrough technology that enables you to assign “SLA-affecting” weights to events, which in turn incur an SLA penalty if invoked. The way it works is as follows:

- You start by defining a “service group” within ProIT. This service group contains all of the components of your mission-critical application(s). There may be 5 components or hundreds of objects, any of them capable of causing an SLA violation (availability or performance).
- ProIT assigns a default SLA optimum score of 1,000 points (so as *not* to be confused with Availability)
- You can assign “SLA affecting” penalties by specifying the number of points for each SLA affecting event. Each time an SLA affecting event occurs, it deducts the point penalty from your optimum score of 1,000.
- You can now truly measure your SLA performance. If you achieve a score of 900 in one month, you can shoot for 950 next month.