

Top Ten Network Management Lies and Deceptions

A free vendor-agnostic whitepaper courtesy of PerformanceIT, Inc.

#1 Simple Network Management Protocol (SNMP) is Simple

This ranks as the number one deception in the industry. SNMP is by no means simple for even a seasoned network engineer. In theory, it is a simple protocol that governs how management data is retrieved and processed from network devices by a Network Management System (NMS) and how network devices send management information back to an NMS. Getting useful and meaningful data for your NMS via SNMP is the real challenge and is where most vendors distort their capabilities and deceive their potential clients.

Most network management vendors claim to be SNMP-capable. What they don't tell you is that what they really offer is simple SNMP GET requests which ask a network device to return the current value of a specific object identifier (OID) such as "packets IN." This simplistic polling yields data such as "packets in = 5,992." What can you do with this information? Not much, of course.

A true NMS assembles appropriate OID values into meaningful *statistics* such as "Bandwidth Utilization." Most vendors won't be honest and explain their SNMP limitations accurately. The statistic Bandwidth Utilization is not a single OID object, as many vendors will have you believe. The actual statistic is created by using the formula: $((\text{IN pkts in Octets} + \text{OUT pkts in Octets}) * 8) / 1024$. There are two OIDs with appropriate mathematics built-in. Many vendors will say they can provide statistics like Bandwidth Utilization, but in truth, they leave it up to you to figure out which OIDs you need, and then how to perform the appropriate mathematics. Be sure to see this functionality demonstrated.

Further, ask your vendor to provide vendor-specific statistics that are important in your infrastructure. If you don't know what is important, it may be that you never implement this powerful monitoring capability. Unless you are a programmer or SNMP expert, be sure to look for an NMS that has the *statistics* you are looking for built-in.

The difference between SNMP polling and SNMP monitoring is as follows: SNMP polling means that a network management station (NMS) polls (sends SNMP GET requests) to the remote device. In other words, it asks the device for information and the device responds with a value. SNMP monitoring means that the network device is configured to send SNMP traps (messages) to an NMS without a specific request. Again, the problem is, as with most network monitoring information, there may be many, many trap messages and they are usually cryptic in raw form.

Example Raw SNMP Trap:

```
"9/13/2002 9:39:8 : grtse38;grtse38;enterprises.16.1.1.200;6;.9002; enterprises.16.1.1.200.1001.0 = 0
enterprises.16.1.1.200.1003.0 = 0 enterprises.16.1.1.200.1004.0 = 1
enterprises.16.1.1.200.1005.0 = "ONLINE" enterprises.16.1.1.200.1006.0 = "FAILED"
```

This message, by itself, does not produce the useful information that is actionable by the network administrator. The network management product must accept this message, simultaneously link its contents to a Management Information Base (MIB) definition file, then convert it to a useful message such as "Server1 PowerSupply 1 FAILED." To say that you can monitor SNMP traps without doing MIB lookups and message conversion wouldn't qualify as a feature for most users. Further, you may also desire specific integration with some of the most common and useful SNMP trap producers such as Compaq (HP) Insight Manager, Dell OpenManage, and IBM Director.

#2 Root Cause Analysis

This “feature” means different things to different vendors and places #2 on our list. Delivering true root cause analysis means that your NMS can filter through hundreds (or thousands) of events and point you directly to the root cause of either an outage or performance degrading event, a very tall order.

To separate fact from fiction in this area, we generally like to separate monitoring capabilities into three categories: Component Monitoring, Transaction Monitoring, and Event Correlation. True Component Monitoring means that your NMS is capable of three distinct capabilities:

1. Monitoring all of the individual components (“moving parts”) of your application across all of the devices and software on which it depends;
2. Be able to understand and group the relationship of these components; and
3. Represent those components as a single view of your application.

Many NMSes boast their number of “monitors” and can monitor the various devices within your network, claiming that they can then provide Component Monitoring and therefore “root cause” analysis capability. Unfortunately, without #2 and #3 above, the closest you will come to a root cause identification is several individual component alarms that are not correlated to other events or to your specific application. Often, you won’t know which event came first and what chain reactions were caused.

Transaction Monitoring means that your NMS has the ability to create actual or synthetic user transactions that exercise every component of an application end-to-end. While Component Monitoring is powerful and valuable, Transaction Monitoring provides a more holistic view from the user’s perspective and ensures that even unknown components are exercised. Two examples would be Email and Web Transactions Probes. An Email Probe creates, sends, receives, and measures email packets out-of and back into your network, assuring that Email is flowing properly. This mechanism is sure-fire and enables you to rest easier knowing this critical app is functioning properly. If a problem does occur, a true Transaction Monitor can provide accurate root cause by specifying which component failed or is causing a performance degradation.

Event Correlation means being able to intelligently correlate storms of events with the outcome being the display of the true root cause. Many times, this functionality is designed to suppress the “noise” so that you can see the root cause more clearly. Event Correlation can be achieved via any of the following mechanisms:

- By Service Group. For example, if a Critical alarm occurs on any item in a service group, all following alarms can be correlated into a single event.
- By Topology. Alarms can be correlated by manually or auto-discovered Layer 2 or Layer 3 network topology or via Dependency Maps
- By Host. Multiple alarms from a single device are consolidated into a single event.
- By Domain. Multiple alarms from a logical domain are consolidated into a single event.

Above all, one of the newest breakthroughs in root-cause analysis is to provide real-time anomaly detection. In order to detect anomalies, your NMS has to have collected sufficient data profiling your network’s behavior. Deviations and anomalies in that behavior are then automatically detected and displayed via for rapid-root cause identification.

#3 Availability Monitoring & Reporting

Ranking #3 on our list of network management lies is Availability Monitoring and Reporting. Most NMS products claim to monitor for availability. This is a blanket statement that must be carefully inspected. Today's applications and networks are more complex and determining application availability is almost always more involved than the mechanism they claim is monitoring it. The primary mechanism used by most NMS products to determine *device* availability is the trusty ICMP ping. If a device responds to ping, it is assumed that the *device* is therefore available.

There are many problems with this assumption. First, a device may be responding to ping, but the *application* may very well be down for any number of reasons. Simple examples include: application services stopped on a server (server is not down but *is* responding to ping), application ports are blocked by a firewall (pings are allowed, a specific port is not). So, if your NMS is monitoring for availability using ICMP ping as its sole mechanism, be prepared for disappointment.

A true NMS solves the complex issue of availability monitoring and accuracy by building in the ability to specify which events are *availability affecting*. This is a very powerful capability because it transcends the traditional event category limitations of Critical, Major, Minor, Warning, and Information. In most NMSes, Critical simply means "needs immediate human intervention." Some Critical events may be *availability affecting*, others not. For example, if one power supply in a server fails and the redundant power supply takes over, this may generate a Critical event even though it has not yet affected availability. However, if *both* power supplies fail, then you have a definitive *availability affecting* event.

Availability Reporting suffers the same problem in most NMSes, reporting either only whether or not a device has responded to ping. Some NMSes use the "uptime" feature built into operating systems to provide their availability reports. This method is flawed as well because it doesn't account for planned or scheduled downtimes such as voluntary reboots.

#4 SLA Monitoring, Measurement, and Reporting

True SLA management is indeed an advanced and sophisticated subject. In order to truly manage SLAs, your NMS must have the ability to establish performance baselines, deliver accurate availability results, and a way to measure improvements or results against the SLA.

In order to manage in accordance with an SLA, your NMS must enable users to organize *services* as they are defined within the SLA. If you are confined to a single view or a device centric view, it may not be possible to organize your monitored objects in a way that matches your SLA.

Next, your NMS must have the ability to set an *SLA score*. This would be your perfect score if you indeed met your SLA requirements. Any events that are *SLA affecting* would then be subtracted from your *SLA score*, providing you with real measurements of SLA compliance. Using this real SLA management mechanism, IT managers can honestly measure and improve their results against real world SLAs.

#5 Capacity Planning and Analysis

In order to provide capacity planning and analysis, your NMS has to have the following two capabilities:

- Ability to analyze historical data, and
- Ability to use historical data to provide accurate forecasts and predictions of future behavior.

The first requirement assures that you need to be collecting data in a relational database. If the proposed NMS does not use a relational database, it is certain that you will have minimal data collection, storage, and reporting capability.

The second requirement is more advanced. Once data is stored in a database, what your NMS can do with the data is where you find true value. Raw data is useless, *information* is valuable. A good NMS turns your raw data into valuable, time saving, and reliable information. Better NMSes use sophisticated mathematical probability and statistical algorithms to provide accurate trend analysis for reliable forecasts and planning.

#6 Easy To Use

All vendors claim their products are easy to use and configure. They all look good during the demo and all seem like they were easy to implement. Of course, this is why Easy To Use ranks #6 on our list. There are many different areas where this claim usually becomes fictitious. Be on the lookout for these “features” as signs that the product will *not* be easy to use or configure:

- 100% Agentless*. While this one sounds good, it really is a dishonest claim. Agentless implies that there is no software to install and therefore it is easier to deploy, manage, and maintain. Sounds reasonable. Of course, the devil is the truth. There is no true “agentless” product, the proposed NMS uses the “agents” that come with another vendor’s product instead, such as Windows’ Host MIB. Host MIB is not enabled by default in Windows 200x; you have to manually configure it (over 30 steps). So, you perhaps don’t have to install a proprietary agent, but you do have to turn on and implement someone else’s. This can cause more problems than the ones they are purported to solve.
- Monitors or Agents for Specific Applications*. Having monitors or agents for specific apps usually means that you have to install one or more agents on a remote host. This *can* create a management and maintenance headache and impose performance degradation on the remote hosts. Look for an NMS that combines the best of both agent and agentless approaches and has a low-impact host agent.
- Central Event Store or Consolidated Event Viewer*. This means that your NMS collects *all* raw events (98% of which you don’t need) into a centralized database, causing unnecessary network traffic overload and processing problems. You are then left to learn and write complex rules and filters to find useful events.

#7 Service Management

One of the hot buzzwords in IT today is “service management.” What this means is managing applications and infrastructure as end-to-end *services* from the client’s perspective and moving away from the device-centric approach. Today’s complex applications rely upon a multitude of protocols and devices which have to work seamlessly together. Then they have to do so in a world of increasingly tight security, adding a whole new dimension in complexity and potential troubleshooting pitfalls. True service management means that you can view an application as its packets flow through your network, almost like watching water flow through a sprinkler system. The challenge in networks is that the pipes are carrying a diverse payload consisting of several different fluids.

Your NMS must enable you to create multiple views of your infrastructure and one that accurately represents your services. For example, your core router carries *all* of your traffic. To accurately represent your *CRM service*, you would need to create a view that monitors only your CRM traffic flowing through your router and nothing else. This can’t be an exclusive feature. You must be able to set this up for all of your services and be able to view them accordingly. To continue the example, your Service View might include the application services on a particular set of windows servers, TCP ports on your firewall, and perhaps objects in a specific database instance.

#8 Plug And Play Installation

Apparently, Plug and Play means different things to different NMS vendors. To most of us, this means that a product we buy is up and running with minimal configuration in a matter of minutes or hours. If we have to hire expensive consultants and spend weeks or months implementing and learning the product, it is *not* plug and play. Ask your vendor for references on how long it took to implement the product, how much daily consultants cost, and what they mean by “plug and play.” Top warning signs that an NMS product can’t be Plug and Play:

- More than 200 systems integration companies exist to implement and configure their product
- Daily consulting rates for their product are higher than any normal consulting rate, typically 2X or 3X normal consulting rates
- Numerous third-party products exist to *enhance* the functionality

#9 Web Based

Although there should be truth-in-advertising laws to cover such claims, unfortunately there is no way to enforce this oft-exaggerated claim. If only 2% of your product is available via a browser, the vendor is likely to claim it is “web-based.” Being fully web-based has both advantages *and* disadvantages. Be sure to size up how web based a particular NMS really is. Some offer reports only, others provide limited configuration capabilities, and any combination in between. Disadvantages include the loss of some “windows-style” comforts such as right-click options and drag & drop capabilities. A fully web-based NMS will provide 100% of its features and functionality in a browser, with no management client software to install, including java applets. This provides true anywhere access.

#10 Best Practices

Ahh, the best saved for last. Whose best-practices are they talking about anyway? The ones concocted in the vendor's lab? Best practices should mean those agreed upon by industry standards bodies or trusted user groups. Unfortunately for all of us, no such standards really exist for network & systems management products. Best practices can be anything a vendor claims them to be. Lately, some vendors tout best practices from the IT Infrastructure Library (ITIL), set of standards originally developed by the British government. Suffice it to say that one's best practices may be another's poison and that there's no one size fits all for monitoring an infrastructure effectively. What we can all agree on is that there exists a base list of known events and performance metrics you should be monitoring. Buyers of NSM products should not have to figure out these known items on their own. These events are "best practices" and should be built-in and have appropriate thresholds pre-determined. If you are lucky, the NSM's "best practices" will be based upon OEM recommendations and published sources.