Monitoring Nortel IP Telephony with AppManager

White Paper
March 2005

Each IP telephony deployment is different, but generally a Nortel IP Telephony deployment includes a Nortel CS 1000 system (Call Server, Signaling Servers, Media Gateways) and/or a Nortel BCM system; and a CallPilot voice mail server, routers, L2/L3 switches, IP phones and other applications.

This paper highlights the components that you should monitor to ensure a successful IP telephony deployment.
Why Monitor IPT Components?

Nortel’s Business Communications Manager (BCM) and Communication Server 1000 (CS 1000) both exemplify high-reliability IP telephony (IPT), but their reliability is dependent on the proper configuration and operation of dozens of associated components.

Nortel BCM is an IP-enabled communication system that integrates sophisticated voice and data capabilities, VoIP gateway functions and quality of service (QoS) data-routing features into a single system for small sites, multi-sites and branch offices. Nortel BCM provides a variety of desktop communications options and offers a broad choice of voice and data software applications, such as Unified Messaging and Multimedia Call Center, optimized for the small-to-medium business environment.

Solutions for Business Communications Manager

AppManager® is designed to help you gain easy access to Nortel BCM data, and to help you analyze and manage that data. The AppManager for Nortel IP Telephony solution minimizes the cost of maintaining Nortel BCM services and functions, aids in capacity planning and helps prevent downtime.

With AppManager for Nortel IP Telephony, administrators gain access to a new set of tools they can leverage to gather a wide range of diagnostic and management data, which can help prevent outages and keep things running smoothly.

AppManager for Nortel IP Telephony includes Knowledge Scripts® that monitor the health, availability and performance of key Nortel BCM features. These scripts allow you to monitor and manage crucial services at a depth unparalleled by any other solution. Each Knowledge Script can be configured to send an alert, collect data for reporting and perform automated problem management when an event occurs.

Solutions for Communication Server 1000

Nortel CS 1000 is designed to provide a full-featured IP Telephony solution for the enterprise environment. It is a robust, survivable, IP-based platform, capable of being distributed across IP Wide Area Networks (WANs) and delivering the full range of proven telephony applications from Nortel Networks. The Nortel CS 1000 system consists of a number of key components that are illustrated in the Nortel CS 1000 Infrastructure diagram on the following page.

AppManager for Nortel IP Telephony is part of the Proactive Voice Quality Management (PVQM) strategy Nortel Networks has co-developed with NetIQ to give network managers the capability to ensure the overall quality of IP Telephony deployments. PVQM continuously and passively measures the users’ quality of experience (QoE) for all IP Telephony communications, conducts system health checks for IP Telephony servers and provides troubleshooting and resolution for any performance degradation or fault conditions.

Monitoring Nortel BCM and CS 1000 with AppManager for Nortel IP Telephony enhances performance, cost effectiveness and reliability, and simplifies the management of your IPT network. A comprehensive management solution is vitally important to the success and reliability of your IP telephony implementation.
Figure 1. Nortel CS 1000 Infrastructure. A typical Nortel CS 1000 solution comprises a Communication Server 1000 (CS 1000) system and a Media Gateway 1000T (MG 1000T) platform.

About AppManager

AppManager is the most-comprehensive and most reliable VoIP management solution on the market. AppManager works to ensure the availability and performance of VoIP systems and networks through the use of Knowledge Scripts, which implement network management rules. Depending on how a Knowledge Script is configured, it can collect performance data (for example, about how many calls have been attempted today), monitor a system for simple or complex events (for example, call quality is poor or a service is down) and respond with one or more actions (such as sending a page or email when there’s a problem, or restarting a service automatically). Much of the monitoring involves collecting data from SNMP MIBs and/or other management data sources.
Nortel BCM Server Health

AppManager checks CPU and memory utilization for Nortel BCM processes on each server you choose to monitor and raises an event when a process exceeds its utilization threshold, indicating reduced performance or an increased risk of a failure. It tracks average CPU and memory usage over time and gives you access to a list of the processes that are consuming the most CPU resources. You can use the data it collects to generate graphs and charts from the AppManager Operator Console and as the basis for AppManager and AppManager Analysis Center reports.

Here’s a list of the most important things to monitor right from the start:

- **CPU usage.** Run the `BCM_SystemUsage` script to monitor Nortel BCM CPU usage and total CPU usage. Run this script every two minutes for a week or two. Then run `Report_SystemUsage` or create a chart to compile the data that you’ve collected. The maximum and average data streams provide valuable trending information.

  Look for spikes in CPU usage. Spikes in excess of 80 percent may indicate that your system can’t handle any new functions or that the Nortel BCM server might start dropping calls. Consider adding another server or moving phones to balance the loads carried by all your servers.

  In addition, run `Sys_CpuByProcess` to verify which application is actually causing the spike in CPU usage.

  Other AppManager scripts can provide more information about CPU usage:
  - `Sys_CpuLoaded`. Monitors CPU usage and queue length.

- **Physical memory.** Run the `BCM_SystemUsage` script to monitor Nortel BCM memory usage and total memory usage. Run this script every 12 hours for a week or two. Then run `Report_SystemUsage` or create a chart to compile the data that you’ve collected. The minimum, maximum and average data streams provide valuable trending information.

  Other AppManager scripts can provide more information about memory usage:
  - `Sys_MemUtil`. Monitors utilization of physical memory, virtual memory and paging files.
  - `Sys_TopMemoryProcs`. Monitors the working-set memory usage of all processes and identifies the top consumers.

- **Disk space usage.** Run the `Sys_LogicalDiskSpace` script to monitor the usage of disk space of your Nortel BCM servers. You can avoid many problems altogether if you take a proactive stance toward managing log file sizes. Run the script every 12 hours. Disk space usage above 75 percent or free space of less than one GB is a signal to delete temporary files and archive and delete log files.

- **Virtual memory.** Run `Sys_MemUtil` every two minutes to monitor the usage of virtual memory, physical memory and paging files. A spike in excess of 75 percent could indicate heavy usage or a more serious issue, such as a virus or denial of service attack. Run the `ReportAM_AvgValueByHr` report script to summarize the data you’ve collected on an hourly basis. Look at the Minimum and Average memory data streams to help you set event thresholds and establish growth needs. Look at the maximum memory data stream to help detect memory leaks.
• **Memory leaks.** A memory leak occurs when a process requests memory for temporary usage, but does not release the memory when the process no longer needs it. This accumulation of memory by a process can starve other processes that need memory, leaving your system unstable or degraded.

Run `BCM_SystemUsage` to monitor physical memory usage and memory pages/sec for the Nortel BCM computer. Run this script for a week or two and then create a chart or run `Report_SystemUsage` to compile the data that you’ve collected.

Graph the daily (or 12-hour period) maximum free memory values to identify possible memory leak conditions at the system-level; you can identify a potential memory-leak condition by maximum free memory values that continuously diminish over time (assuming other parameters, such as the number of registered devices, remain somewhat constant).

Other AppManager scripts can provide more memory-related data:

– `Sys_MemUtil`. Monitors utilization of physical memory, virtual memory and paging files.
– `Sys_TopMemoryProcs`. Monitors the working-set memory usage of all processes and identifies the top consumers.

---

**Nortel CS Administration**

The `NortelCS_GetOMReport` Knowledge Script must be run prior to any of the other `NortelCS` Knowledge Scripts. This script is used to obtain the latest Operational Measurement (OM) report from the signaling server or voice gateway media card. OM reports contain hourly statistics for QoS (jitter, latency, lost packets, R-Factor) and various totals (including voice time, registrations, unregistrations, calls attempted and calls completed). The statistics are aggregated for each type of phone. The other Knowledge Scripts process this report information for the management information.

---

**Nortel CS Server Health**

AppManager checks the health of the CS 1000 Call Server, Media Gateway, Signaling Server and Voice Gateway Media Card (VGMC). An event is raised if the server is not responsive or is in an abnormal state. Use the `NortelCS_HealthCheck` Knowledge Script to monitor the server health.

There are three different attributes to server health and multiple possible states for each attribute (the state considered normal is shown in italics):

• **Card State:** active, standby, unknown
  
• **Application State:** normal, maintenance, test
  
• **M1 State:** enabled, disabled, unequipped, unknown

Following are two common error messages, accompanied by explanations, likely causes and any operator actions that may be needed.

**Error Message:** `<Call Server | Media Gateway | Signaling Server | VGMC> unresponsive to SNMP:<IP address>`
• Explanation: The Call Server, Media Gateway, Signaling Server or VGMC is not responding to SNMP.
• Likely cause: Normal message when item is unresponsive.
• Operator action: Restart the Call Server, Media Gateway, Signaling Server or VGMC.

**Error Message:** <Call Server | Media Gateway | Signaling Server | VGMC> in abnormal state:IP address>

• Explanation: The Call Server, Media Gateway, Signaling Server or VGMC is not running in the proper state.
• Likely cause: Normal message when state is abnormal.
• Operator action: Change the state of the Call Server, Media Gateway, Signaling Server or VGMC.

---

**Nortel CS Functionality**

Once you’ve taken care of the essential monitoring tasks outlined in the sections above, you’re ready to extend coverage to the functionality of CS 1000 itself.

Nortel CS 1000 Call Servers, Signaling Servers, Media Gateways and VGMCs send alarms to the proxy agent using SNMP traps. AppManager monitors the Nortel CS 1000 proxy agent for Nortel CS 1000 alarms using the *NortelCS_Alarms* script. Once Nortel CS 1000 has been configured to send SNMP traps to AppManager, you can choose which alarm categories you wish to monitor and can select particular alarms to include or exclude. Alarm categories include: critical to monitor, QoS, critical, major, minor, warning, info, cleared and indeterminate.

Additionally, for each signaling server, H.323 or SIP virtual trunk and VGMC, AppManager collects the following call data:

- **Phone registration attempts.** You’ll want to be informed any time the number of registration attempts and failures exceeds your threshold. Run *NortelCS_SS_Registration* every hour. Data can be collected for charting purposes to show the average and maximum registration attempts, registration failures and un-registration attempts.

- **Incoming voice calls.** It’s important to monitor the number of attempted incoming calls for call failures. Run the *NortelCS_SS_H323Stats* and/or *NortelCS_SS_SIPStats* Knowledge Script to monitor whether the number of incomplete and attempted incoming calls is greater than the threshold you set.

- **Outgoing voice calls.** It’s important to monitor the number of attempted outgoing calls for call failures. Run the *NortelCS_SS_H323Stats* and/or *NortelCS_SS_SIPStats* Knowledge Script to monitor whether the number of incomplete outgoing calls is greater than the threshold you set.

- **Incoming fax calls.** If you have an environment with a fax solution, run the *NortelCS_SS_H323Stats* and/or *NortelCS_SS_SIPStats* Knowledge Script to monitor the number of completed and attempted incoming fax calls.
• **Outgoing fax calls.** If you have an environment with a fax solution, run the `NortelCS_SS_H323Stats` and/or `NortelCS_SS_SIPStats` Knowledge Script to monitor the number of completed and attempted outgoing fax calls.

• **Available bandwidth.** Voice traffic requires specific bandwidth based on the codec employed. G.711 requires about 64 Kbps for each direction of a bi-directional call. G.723 and G.729 require significantly less bandwidth due to compression, but congestion can severely impact call quality. Each time you add a new application to the mix on your network, you risk the oversubscription of certain links. Congestion will almost certainly affect overall call performance, particularly if data loss or excess latency occurs. Voice quality is susceptible to catastrophic degradation under conditions of network oversubscription.

To ensure that you have adequate bandwidth that you know when bandwidth availability is low, run the following Knowledge Scripts every five minutes:

- `NetworkDevice_SingleWANLink_Util`. Monitors a single WAN (serial, T1, or T3) link on a network device.
- `NetworkDevice_WANLink_Util`. Monitors WAN (serial, T1, or T3) links on a network device.
- `NetworkDevice_Device_Uptime`. Monitors the number of hours that a network device has been operational since its last reboot, allowing you to see when a device went down and to graph its availability over time.

---

**IP Gateway Health**

We recommend that you constantly monitor VoIP gateways for availability, CPU statistics, memory usage and link utilization. Run the following Knowledge Scripts to gather the relevant data:

- `NetworkDevice_Chassis_Usage`. Monitors the physical chassis of a network device.
- `NetworkDevice_Interface_Health`. Monitors the interfaces on a network device.
- `NetworkDevice_LANLink_Util`. Monitors the LAN links on a network device.
- `NetworkDevice_WANLink_Util`. Monitors the WAN (serial, T1, or T3) links on a network device.
- `NetworkDevice_Device_Uptime`. Monitors the number of hours that a network device has been operational since its last reboot.

---

**QoS Monitoring**

In order for VoIP users to receive an acceptable level of voice quality, VoIP traffic must be given priority over other kinds of network traffic, such as data. The main goal of QoS monitoring is to ensure that VoIP traffic receives the preferential treatment it needs, thereby reducing or eliminating the delay of voice packets that travel across a network.

You should monitor the following aspects of VoIP call quality:

- **Delay.** The end-to-end delay, or latency, as measured between endpoints is a key factor in determining VoIP call quality.
• **Jitter.** Jitter is a call quality factor known to adversely affect call quality. Jitter is also called delay variation, and it indicates the variance of the arrival rate of datagrams sent during a simulated VoIP call.

• **Jitter buffer loss.** Jitter buffer loss is the amount of data that is lost when jitter exceeds the jitter buffer capacity. Jitter buffer loss affects call clarity, which affects the overall Mean Opinion Score (MOS).

• **Packet loss.** When a datagram is lost during a VoIP transmission, you can lose an entire syllable or word in a conversation. Obviously, data loss can severely impair call quality.

• **R-value.** Defined by International Telecommunication Union (ITU) recommendation G.107, the E-model is a complex calculation, the output of which is a single R-value derived from delays and equipment impairment factors. An R-value can be mapped to an estimated MOS. R-values range from 100 (excellent) to 0 (poor). As shown below, an estimated MOS can be directly calculated from an R-value:

```
<table>
<thead>
<tr>
<th>R-value</th>
<th>User Satisfaction</th>
<th>MOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Very satisfied</td>
<td>4.4</td>
</tr>
<tr>
<td>94</td>
<td>Satisfied</td>
<td>4.3</td>
</tr>
<tr>
<td>90</td>
<td>Some users dissatisfied</td>
<td>4.0</td>
</tr>
<tr>
<td>80</td>
<td>Many users dissatisfied</td>
<td>3.6</td>
</tr>
<tr>
<td>70</td>
<td>Nearly all users dissatisfied</td>
<td>3.1</td>
</tr>
<tr>
<td>50</td>
<td>Not recommended</td>
<td>2.6</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>1.0</td>
</tr>
</tbody>
</table>
```

• **MOS.** By comparing your real network data with the subjective MOS, you can understand which network factor is clearly affecting voice quality. The MOS is an overall score representing the quality of a call. The MOS is a number between 1 and 5. An MOS of 5 is excellent; an MOS of 1 is unacceptably bad.

Several AppManager Knowledge Scripts simulate a VoIP call between performance endpoints. After simulating a call, the scripts can gather data about some or all of the QoS metrics as they relate to your network:

• **VoIPQuality_CallPerf_G711a.** Simulates a VoIP call between endpoints using the G.711a codec, which is the ITU standard for H.323-compliant codecs. Uses the A-law for companding, a popular standard in Europe.

• **VoIPQuality_CallPerf_G711u.** Simulates a VoIP call between endpoints using the G.711u codec, which is the ITU standard for H.323-compliant codecs. Uses the U-law for companding, the most frequently used method in North America.

• **VoIPQuality_CallPerf_G723.1-ACELP.** Simulates a VoIP call between endpoints using the G.723.1-ACELP codec, which uses the conjugate structure algebraic code excited linear predictive (ACELP) compression algorithm.
- **VoIPQuality_CallPerf_G723.1-MPMLQ.** Simulates a VoIP call between endpoints using the G.723.1-MPMLQ codec, which uses the multipulse maximum likelihood quantization (MPMLQ) compression algorithm.

- **VoIPQuality_CallPerf_G726.** Simulates a VoIP call between endpoints using the G.726 codec, which is a waveform codec that uses Adaptive Differential Pulse Code Modulation (ADPCM). ADPCM is a variation of pulse code modulation (PCM), which only sends the difference between two adjacent samples, producing a lower bit rate.

- **VoIPQuality_CallPerf_G729.** Simulates a VoIP call between endpoints using the G.729 codec, which is a high-performing codec that offers compression with high quality.

- **VoIPQuality_CallPerf_G729A.** Simulates a VoIP call between endpoints using the G.729A codec, which is a reduced-complexity version of the G.729 codec, developed for simultaneous voice and data applications for which the G.729 codec was too complex. Speech quality is virtually indistinguishable between G.729 and G.729A.

Monitoring voice quality is a real-time process. You need to know about obstacles that could hinder call quality before your users are affected. You need to know immediately if voice quality has deteriorated below an acceptable level. For a BCM environment, the following Knowledge Scripts help you monitor and collect information about Quality of Service:

- **BCM_QoSInterfaceActivity.** When a packet is dropped during a VoIP transmission, you can lose an entire syllable or word in a conversation. Obviously, data loss can severely impair call quality. This script will notify you if your BCM is dropping too many packets. When you set the parameters for the QoSInterfaceActivity script, use the **Threshold percent for dropped packets** parameter to determine how many packets is “too many.”

- **BCM_QoSLog.** If you use VoIP trunking, run this script to gather information from the QoS Monitor log in order to verify that QoS between target BCMs is maintaining an acceptable MOS. Run this Knowledge Script to monitor the MOS scores for the following codecs in transmit and receive directions: G.711a, G.711u, G.723-5.3 kbps, G.723-6.3 kbps and G.729. This script raises an event for every line in the Nortel QoS Monitor log that contains an MOS score that exceeds a threshold. To review historical MOS information, run **Report_QoSSummary**, which displays the average QoS for a BCM group.

- **BCM_QoSPrioritySessions.** If a priority session requests to use the Premium queue 1, but that session is not served on the Premium queue, then the associated call can be subject to jitter or even failure. This script notifies you if too many sessions are not being served. When you set the parameters for the QoSPrioritySessions script, use the **Threshold for sessions not served** parameter to indicate the most “not served” sessions that your users can tolerate.

AppManager monitors the Nortel CS 1000 proxy agent for QoS alarms that are sent to the proxy agent as SNMP traps from the Call Servers, Signaling Servers, Media Gateways and VGMCs. QoS threshold levels must be manually configured in Nortel CS 1000 Element Manager. Call Quality metrics that fall outside of the thresholds will be identified by the **NortelCS_Alarms** Knowledge Script.

The Knowledge Scripts **NortelCS_SS_CallQuality** and **NortelCS_VGMC_CallQuality** are also available to monitor call quality for a Nortel CS 1000 environment. For each signaling server, H.323 or SIP virtual trunk and VGMC, AppManager collects the following call data from the OM report:
- **Lost packets.** The total number of data packets that have been lost (on the signaling server or VGMC) since the beginning of reception. This number is defined as the number of packets that were expected minus the number of packets that were actually received. The number of packets received includes those that were late or duplicates. Packets that arrive late are not counted as lost; the presence of duplicate packets could result in a negative lost data amount. AppManager computes a weighted average percentage of lost packets for all phone types you choose to monitor, as well as the maximum for all the phone types you choose to monitor.

- **Jitter.** An estimate of the statistical variance of the RTP data packet interarrival time, measured in milliseconds and expressed as an unsigned integer. Interarrival jitter is the mean deviation (smoothed absolute value) of the difference in packet spacing at the receiver compared to the sender for a pair of packets. AppManager provides both maximum and average jitter values for the signaling server or VGMC. AppManager computes a weighted average amount of jitter for all phone types you choose to monitor, as well as the maximum for all the phone types you choose to monitor.

- **Latency.** An estimate of the network latency, expressed in milliseconds. Latency is the average value of the difference between the time stamp indicated by the senders of the messages and the timestamp of the receivers, measured when the messages are received. The average is obtained by adding all of the estimates, then dividing by the number of messages that have been received. AppManager provides both maximum and average latency values for the signaling server or VGMC. AppManager computes a weighted average amount of latency for all phone types you choose to monitor, as well as the maximum for all the phone types you choose to monitor.

For Nortel CS 1000 version 4.0 and higher, the *NortelCS_SS_CallQuality* Knowledge Script also collects the following data:

- **R-value.** Defined by ITU recommendation G.107, the E-model is a complex calculation, the output of which is a single R-value derived from delays and equipment impairment factors. AppManager computes a weighted average of the R-factor values for all phone types you choose to monitor, as well as the minimum for all the phone types you choose to monitor.

- **MOS.** The MOS is an overall score representing the quality of a call. The MOS is a number between 1 and 5. An MOS of 5 is excellent; an MOS of 1 is unacceptably bad. AppManager converts the average and minimum R-factor values into an MOS using a conversion formula provided by the ITU.

### Layer 2 and 3 Switches

We highly recommend that you continually monitor Layer 2 and Layer 3 switches for switch failures, card failures (such as reboots, crashes), memory utilization, CPU utilization, power supply status, temperature status, fan status, QoS parameters and IP phone port status.

The following AppManager Knowledge Scripts provide the monitoring capability you need:

- **NetworkDevice_Chassis_Usage.** Monitors the physical chassis of a network device.

- **NetworkDevice_Interface_Health.** Monitors the interfaces on a network device.

- **NetworkDevice_LANLink_Util.** Monitors the LAN links on a network device.
• *NetworkDevice_Device_Uptime*. Monitors the number of hours that a network device has been operational since its last reboot.

---

**Reporting**

AppManager collects data about the performance of IP telephony and stores it in the AppManager repository, a SQL Server database. You can access this data in real-time or historically for your reporting needs.

**Real-time**

The AppManager Chart Console lets you generate and view charts of data streams generated by Knowledge Script jobs. As the jobs run, the data streams in the charts are continually updated with new information. The Chart Console provides key data that you can use instantly to manage and troubleshoot your Nortel IPT environment.

You can use the AppManager GUI- or web-based Chart Console to view collected data in real-time at regular intervals as low as one minute. Viewing of data can be organized and segmented by data stream and access to charts can be restricted by AppManager user login.

Any data displayed in charts can also be viewed using AppManager Report scripts and selecting the desired data streams.

In addition, AppManager includes the *ReportAM_Chart2HTML* Knowledge Script, which allows you to easily convert charts to reports.

**Historical**

We recommend that you collect trending information whenever and wherever possible. Trending information should contain at least maximum and average values, which can then be used to define “above average” and “peak” thresholds for the different parameters. If possible, the threshold should be defined using the average and maximum values observed during the busy hour of the day in order to avoid unnecessary alerts.

**Compiling the Collected Data**

AppManager reports are generated using Report Knowledge Scripts. AppManager ships with dozens of Report scripts to generate HTML reports based on any type of collected data. Access to reports can be restricted using IIS web site directory security.

The following is a list of frequently used generic Report Knowledge Scripts:

- *ReportAM_AggValueHistory*. Generates a report from data in the archive and aggregate tables.
- *ReportAM_AvgMaxMinValue*. Displays the average, maximum and minimum values of the data stream(s) collected by a Knowledge Script within a specified time frame.
- *ReportAM_AvgValueByDay*. Details the average daily value of data streams collected by Knowledge Script jobs.
- *ReportAM_AvgValueByHr*. Displays the average values by hour of the data stream(s) collected by a Knowledge Script within a time range.
• *ReportAM_AvgValueByMin*. Displays the average values by minute of the data stream(s) collected by a Knowledge Script within a time range.

### Conclusion

Nortel BCM and CS 1000 are excellent choices for your IP telephony implementation. As with any sophisticated system, though, there may be a few hurdles along the way to your goal of a VoIP network with five-nines of reliability; network hardware and links go down; software, services and processes consume limited CPU and memory resources; intruders interfere with administrative files and records.

Nortel Networks has worked hard to ensure that your Nortel BCM and CS 1000 system will be as reliable as the telephone networks we all take for granted. But keeping the network running perfectly all the time requires proactive management and a good understanding of the various system components—including the operating systems, databases and servers—that support Nortel BCM and CS 1000.

Intelligent deployment and ongoing monitoring are required to keep Nortel BCM and CS 1000 and their associated software and hardware operational, efficient and reliable. The necessary monitoring tasks you need to perform are numerous and can be quite time-consuming. Fortunately, AppManager and its Knowledge Scripts can perform them for you.

For additional information please see the following:

## Appendix: Supported Environments

AppManager modules support the following platforms:

<table>
<thead>
<tr>
<th>Module:</th>
<th>Supported platforms:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nortel IP Telephony</td>
<td>Nortel Networks CS 1000 R3.0, R4.0</td>
</tr>
<tr>
<td></td>
<td>Nortel Business Communications Manager 3.6 or later</td>
</tr>
<tr>
<td>H.323 Call Setup</td>
<td>Microsoft Windows 2000 SP2, or Windows NT 4.0 SP6a</td>
</tr>
<tr>
<td>Microsoft Exchange Unified</td>
<td>Exchange Server 5.0, 5.5 and Exchange 2000 Server</td>
</tr>
<tr>
<td>Messaging</td>
<td></td>
</tr>
<tr>
<td>Network Devices</td>
<td>All known Cisco Systems switches, routers and gateways</td>
</tr>
<tr>
<td></td>
<td>Nortel BayStack switches, models 460 and above</td>
</tr>
<tr>
<td></td>
<td>Nortel Networks routers, BayRS v14 and above</td>
</tr>
<tr>
<td></td>
<td>Nortel Access Stack Node (ASN) Series</td>
</tr>
<tr>
<td></td>
<td>Nortel Backbone Concentrator Node (BCN) Series</td>
</tr>
<tr>
<td></td>
<td>Nortel Backbone Link Node (BLN) Series</td>
</tr>
<tr>
<td></td>
<td>Nortel Backbone Node (BN) Series</td>
</tr>
<tr>
<td></td>
<td>Nortel Passport Advanced Remote Node (ARN) Series</td>
</tr>
<tr>
<td></td>
<td>Nortel Passport Series, including 8600 series</td>
</tr>
<tr>
<td></td>
<td>Alcatel OmniSwitch/Router 6000 and 7000 Series</td>
</tr>
<tr>
<td></td>
<td>Extreme Networks switches using ExtremeWare v6.1.8 and above</td>
</tr>
<tr>
<td>SIP Call Setup</td>
<td>Microsoft Windows 2000 SP2</td>
</tr>
<tr>
<td>VoIP Quality (Call</td>
<td>Microsoft Windows 2000 SP2, Windows NT 4.0 SP6a</td>
</tr>
<tr>
<td>Performance)</td>
<td>Linux for x86</td>
</tr>
<tr>
<td></td>
<td>Sun Solaris (x86 and SPARC)</td>
</tr>
<tr>
<td>Windows</td>
<td>Microsoft Windows 2000 SP2, or Windows NT 4.0 SP6a</td>
</tr>
</tbody>
</table>

For the most current information on supported products please visit:

http://www.netiq.com/support/am/supportedproducts.asp