Packetized Voice: It's The Software, Stupid!

Gary Audin

Sure, end users need to prepare for packetized voice and converged networks. But the vendors also need to change, particularly how they approach software issues.

Here's what some of the leading router vendors say about software: ■ At Juniper's first-ever worldwide analyst meeting, CTO Prodeep Sindu had this to say about high-reliability router software: "Technologically, we don't know how to do this. Software is complex. You cannot anticipate all future situations."

■ In an interview with Jim Duffy published in a recent "View from the Edge" newsletter (*Network World*, May 30, 2002), Roland Acra, VP and GM of Cisco's Internet Routing group, agreed with Sindu's perspective. "Software continues to be imperfect," Acra said. "Even if the software is rock solid, you want to minimize dependence [on a single system] and also to protect against human error. We're certainly shooting for not having the need for it [redundant systems], but we also want to be realistic."

Both gentlemen were referring to router software, the same routers that carry IP traffic and which have been in the field for years. So, what does this say about software for IP-PBXs, gatekeepers, gateways and softphones, which are much newer?

When designing software for telecommunications, four criteria rise to the top: reliability, stability, features and security. Software has to run smoothly, over long periods of time with fast recovery from problems. It has to have the features users and administrators require. Legacy phone switches can have security problems, but there is no problem with viruses or denial of service.

Other criteria for software design include:

- Required processor power.
- Memory size.
- Ease of maintenance and upgrade.
- User/administrator interface.
- Operating system and language.
- Reduced complexity.

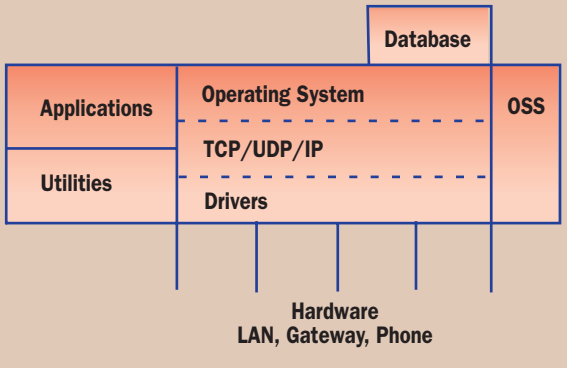
Software history has a way of recycling. Since programmers are human, they sometimes fail to look back and avoid the mistakes of the past. Many choose function over reliability unless customers require the reverse, as in mission-critical applications such as aircraft, manufacturing, appliances, mobile phones, process control, and so on. Poor reliability can lead to embarrassment, damage to the company and/or brand image, financial loss and/or mission setback.

What Do You Get In Software?

The software of an IP-PBX or converged (hybrid) PBX or IP-enabled PBX is a collection of components, some chosen by the customer, most inherited and mandatory by vendor design (Figure 1). Although the operating system (OS) for legacy PBXs has traditionally been proprietary and virtually unknown to the customer, it didn't matter because there was a single-source vendor for all hardware and software components. But as we migrate to the world of IP, the OS becomes visible to the user and may be a third-party product (see "OS on the Menu," p. 49).

The OS is a program that is loaded (booted) into the computer first. It manages and schedules all other programs. The OS provides services to other programs, called applications, through an Applications Program Interface (API). Users can interact with the OS through a command language or graphical user interface (GUI).

FIGURE 1 Call Server Software



Gary Audin is president of Delphi, Inc., an independent consulting and training firm. He has extensive experience in the planning, design, implementation and operation of a wide range of networks, and is an instructor for a number of BCR *seminars. He can be reached at delphi-inc @worldnet.att.net.*

OS On The Menu

■ **Unix:** Originally from Bell Labs (1969), it was the first OS to support the C language. Used in mainframes, mini-computers, workstation, servers and appliances, Unix was the first OS to evolve into a form of freeware. Rather than being owned by any individual vendor, there are many versions supported by different vendors. The C language and shell interfaces (user commands) for these versions are standardized by the IEEE as the Portable Operating System Interface (POSIX). Version 1 of the POSIX interface is known as Unix 95; version 2 is Unix 98. The trademarked Unix is owned by The Open Group standards organization.

■ **Linux:** A Unix-like OS that was designed for the PC, providing either a low cost or free OS. It is publicly owned and is extendable by any contributors. Designed for efficiency and fast performance, Linux is a complete OS with TCP/IP, X Window, EMACS editor, graphical user interface (GUI) and other components similar to Unix. There are some copyrights held by individual creators, but Linux is distributed freely. If a modified version is created, it must be made available to everyone else free of charge. Linux has grown in popularity as Intel, IBM, Avaya and other companies support it, and it may become a major competitor to Microsoft Windows.

■ **VxWorks:** An OS offered by Wind River Systems and others, it is designed for embedded systems—to work as the OS in other vendors' products, such as softswitches, routers, switches, DSLAMs and wireless devices, where high availability (read: 5–9s) service is required. It supports the C language and TCP/IP connectivity with a Unix-like OS for real-time operations as found in telecommunications. The latest version is VxWorks AE.

■ **Windows NT:** A Microsoft OS designed for business users. It is two products: Microsoft NT Workstation and Microsoft NT Server. The Workstation version is the successor to Windows 95 and Windows 98. The Server version is designed for networked computers that want to share common resources. The major competitor to NT has been Novell's Netware OS, although NT also competes with Unix-based systems. While NT has many good features, it was not designed for high-availability telecommunications applications and is therefore not used in routers, for example.

■ **Windows 2000:** Previously called Windows NT 5.0. It is designed to support larger business environments with more technical requirements than earlier NT versions.

■ **Windows XP**: The latest desktop successor to Windows 2000 improves reliability and security. It comes in Home and Professional versions, but there isn't a server version yet. About 90 percent of the XP code comes from Windows 2000□

We want stability in our software, but we don't usually get it

Unix, Linux and VxWorks are common OSs. Windows started as a PC operating system with a lot of user support (Win95/98) and gradually evolved to offer server support (Windows NT/2000—Table 1 p. 50.) Unix has been around, improving in various forms, for more than three decades. Interest in Linux for telecommunications, obviously, is more recent, but it's growing. In January 2002, at the Open Source Development Lab, a consortium devoted to open source software, the Carrier Grade Linux Working Group was formed by Alcatel, Cisco, HP, IBM, Intel, Nokia, MontaVista, RedHat and SuSE International.

VxWorks is a high-availability OS designed for embedded systems—i.e., it works in devices that are not recognized as computers, for example, microwave ovens, cars, aircraft and factory controllers. Wind River Systems is dominant among a field of more than 10 vendors.

The OS may also contain utilities providing usually non-essential services. Drivers or device drivers are programs that control a particular device. The OS usually comes with a set of drivers for operating printers, CD-ROM, disk drives, monitors, etc. In telecommunications, driver software is also required for line cards, trunk cards, Ethernet NICs, frame relay, ATM and other hardware interfaces.

Applications are the reason for the existence of the OS. An application performs a specific function—features, system administration, reports, user interface, database, file systems and even the protocol software (TCP/IP, signaling) are all applications. The network and hardware/software management of local and remote resources and operation support system (OSS) that manages service ordering, maintenance, provisioning, billing and repair are also applications.

A softphone is a software-based telephone, usually operating on a PC. The PC contains all the software elements of a server (OS, applications, utilities, drivers, etc.) as a call server, but it operates on a smaller scale—for one user. Microsoft's NetMeeting, running with Windows, is a softphone (Figure 2).

Is Change Good?

Webster's dictionary defines "stable, as "firmly established, not changing or fluctuating, permanent, enduring, steady in purpose," and

If new software isn't regression tested, it shouldn't be installed

that's certainly what we desire in our hardware and software. It's not what we get. Change is good, but continuous change often creates problems.

When a vendor issues a new version or release of software, you receive something new and/or fixes for something old. While in the legacy PBX world, installing the upgrade was optional, or at least delayable, some of the IP-PBX vendors who come out of the data industry feel that mandatory new software is the way to go. And the new software comes out often—sometimes every three to four weeks—and, in some cases, without notification to the customer or even to the vendor's own technical support personnel. The net effect: Customers can't plan, they can only react.

"They [Cisco] solve some serious issues [but] I cannot stand that they have releases where things do not work from previous software versions. Cisco will sometimes add features in new versions of its management software, or even IOS [router software], that are incompatible with older features. It really creates quite a problem," according to Bill Stevenson at New Century Mortgage in Irvine, CA (*from Network World Fusion* at www.nwfusion.com/news/2002/0603ciscowish.html).

From where I sit, if new software isn't regression tested it should not be installed. Regression testing insures that the new release/version/update works with previously successful features and functions. If Cisco did better regression testing, Bill Stevenson would not have a complaint. While legacy PBX vendors, like Nortel, appear slow bringing software to the market, at least they're serious about regression testing and will not release software prematurely.

As shown in Table 2, the size and frequency of releases/updates (as well as new versions) varies considerably. So far, the legacy vendors are following their traditional pattern of versions/releases/updates. Nortel is actually reducing its releases to once a year. Most have extended their legacy software platform of features to include IP support, while retaining much of the original software. This software is field proven and traffic stress-tested. The frequency of releases/versions/updates is relatively low, and so are the installation risks for the user because the feature set is already complete and the fixes are generally limited to the IP addition. Little or no operating system changes occur.

Many of the newer entrants into the PBX market come from a different position. Releases/versions/updates not only contain fixes, they contain a growing but limited set of new features. The software fixes have become an issue, as Bill Stevenson noted above, so in response, Cisco has promised to produce a major release containing only fixes by December of this year. What does

FIGURE 2 Softphone Software

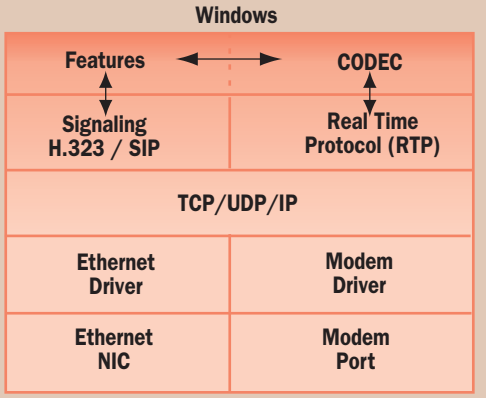


TABLE 1 Operating System Usage by product*

Vendor	Unix	Linux	VxWorks	Windows 2000 / NT	Other or Proprietary
3 Com			NBX		
Alcatel					Omni/PCX
Avaya		ECLIPS		IP600	Definity
Cisco				AVVID ICS/MCS	
Ericsson				MD110 Server	
Mitel		3100	3300		
NEC					NEAX 2400 NEAX IPX NEAX 2000
Nortel			Meridian Succession		
Siemens				HiCom150 HiPath3000 HiPath5000	HiCom300 HiPath4000
Shoreline			Shoreline3		
Vertical Networks				Instant Office	

* Reference: Alan Sulkin's "PBX Feature/Function Charts," published at VoiceCon, 2002

TABLE 2 VOIP-PBX Software Release/Update Rate

Vendor	Occurance	
	Release	Update
3Com	2 per year	every 2 to 3 months
Avaya	2 to 3 per year	every 6 to 12 months
Cisco	2 per year	every 3 to 4 weeks
Mitel	2 per year	every 3 to 4 months
NEC	2 per year	every 4 months
Nortel	2; going to 1 per year	every 6 to 12 months

this say to the existing customer base? Is this another "Trust me" marketing spin?

New software versions and releases present many issues:

1. New software can create new problems. In one case, Cisco distributed a release that terminated a previously operating feature.

2. Adding new software opens up the opportunity for mistakes or negligence, which causes operational problems until re-installed.

3. Whenever new software is added, it is an opportunity for introducing new security problems, either by negligence or on purpose.

4. Installation takes time. Cisco estimates that its version 12 release for the 7750 takes at least five hours to install—and the goal of "five-nines" thus takes another hit.

5. Users must verify what router and/or LAN switch release/version/update installation is required with VOIP product installation. Legacy (voice-centric) vendors are usually IP network-agnostic, but data-centric vendors may require IP/LAN network updates, too. The new VOIP release may not work until the routers and LAN switches have also been revised with new software. This includes troubleshooting and diagnostic software.

6. Vendors may not fix or even perform an audit of a problem if the customer has not installed all the latest releases in the VOIP product and routers and LAN switches.

The OS is part of the security equation

Feeling Secure

Beyond the reliability issues, security emerges. In the legacy world of PBXs, if you kept the maintenance port closed, you kept out hackers. There were no virus or denial-of-service situations. Interactive voice response (IVR) and voicemail systems were not as well protected, but security problems could be reduced and restricted by proper installation of software.

The operating system is part of the security equation, and the Windows world is more vulnerable to these problems. Windows is extremely widespread, making it a likely target for hackers and viruses. There is a large population of users who know the resources and design of Windows, useful information if one wants to circumvent security software.

These last two factors work in favor of the non-Windows operating system. Also, an

Software Questions For Your Vendor

1. What is the frequency of software versions? One, two or three times a year? And what is included (features and/or fixes)?

2. What is the frequency of releases? Are they simply fixes? Are they mandatory installations?

3. When there is either a new version or release, does the system need to be turned off? If so, for how long (hours, tens of hours)?

4. How well are new releases/versions/updates regression tested before delivery? Are they only tested for function, or is stress (traffic) also tested?

5. Does adding an IP phone, line card, trunk card or gateway cause a service interruption?

6. Can a faulty feature/application be suspended while the rest of the system (server, etc.) operates successfully?

7. What support is there for third-party applications and management? Is it standardized or proprietary? What are the APIs (Application Program Interfaces)?

8. Is the software new, was it ported from existing products or purchased from a third party and integrated by the vendor? What happens when the third party changes software, like a Windows upgrade? Who is responsible? Is it free? For how long?

9. What training is required for the support staff? Do I train data people in telephony or telephone people on servers and software? Does the staff require certification? If so, how much training, for how long, and at what cost? Is this a continuous process?

10. How do I train the users in the available software features? Are they trained by phone, manuals, classroom, website, traditional computer-based training (CBT) and/or Web-based CBT? Is it only text or video instruction? Do I pay once, or ongoing, or per user?

11. Does the software license contain a non-disclosure agreement regarding problems? If so, what is the vendor trying to hide?

12. Is the users' group independent of the vendor, allowing users to share information without restrictions?□

The same day XP was released, Microsoft issued two security patches for the OS. Only one worked

ISO 9000

The International Organization of Standardization (which uses the acronym ISO) created a set of quality management standards in 1987. These standards were updated in 1994 and again in 2000. The latest versions are ISO 9001:2000, which describes requirements, and ISO 9000:2000 and ISO 9004:2000, which describe guidelines. All these versions are lumped together and called ISO 9000. Specifically, these are management process standards, not product standards. They apply to all kinds of areas, including hardware, software and consulting in communications.

ISO 9000 systematizes procedures, policies, record-keeping, technologies, structures and resources. A company or organization can be certified or registered; both terms mean the same thing, but different terms are used in various countries.

Being certified/registered is not the same thing as being compliant. Certified/ registered means that an independent registrar has audited the quality system and has certified that it satisfies the ISO 9000 requirements (1994 or 2000 version).“Compliant” means that the company declares it has met the ISO 9000 requirements but it has not been audited by an independent registrar. It is a self-certification.

Whether certified/registered or compliant, this means the management process that generates products and services meets ISO 9000 requirements. A vendor may not have revised its use of the standards to the 2000 version and may be using the older versions (ISO 9001:1994, ISO 9002:1984 and/or ISO 9003:1994). That does not mean that the product or service itself meets ISO 9000 requirements□

operating system such as VxWorks, which can function when some software elements are failing, helps isolate security problems. In addition, security of applications software should include steps for preventing toll fraud, abuse, call monitoring and negligence.

An article by Dan Richman, which appeared in *The Seattle Post-Intelligencer*, June 25, 2002, announced Microsoft’s intention to improve software security in a multiyear effort. The design includes both new hardware and new software. The program, called Palladium, is similar to the program and efforts of the Trusted Computing Platform Alliance.

“This goes beyond the function of an operating system. A lot of software has to be written,” according to Ira Winkler, president of the Internet Security Advisors Group. IBM has already shipped security chips in some of their laptops. Intel and Advanced Micro Devices have signed on to Palladium. The software piece of Palladium will be part of Windows in 2004 or later.

What do you do now to protect your VOIP environment when it is clear that software protection is incomplete?

1. An IP-PBX on a separate LAN or VLAN using legacy trunking puts up a barrier that cannot be crossed by external IP intruders.
2. IP trunking connected to a legacy PBX may create gateway problems, but the PBX is protected from IP intruders.
3. Place a firewall between the VOIP network and the rest of the IP network.
4. Create and *enforce* policies and procedures to improve security.
5. Periodically verify the phone number/IP directory and the status of all VOIP devices to ensure that no illegal devices have been added to the network.
6. Monitor moves, adds and changes, and check against previous configurations for validity.
7. Limit the number of persons with knowledge of and access to the VOIP devices.
8. Schedule, limit and control the installation of new software versions and releases. Software installation is a time of security vulnerability.
9. Ensure regression testing includes software security.

Good News/Bad News

“Why Software is So Bad,” written by Charles C. Mann and published in the July/August 2002 edition MIT’s *Technology Review* (www.technologyreview.com) concludes that the software industry is learning how to improve the reliability of software—good news. The bad news is that the learning is far from done.

On the same day that Microsoft issued Windows XP (Oct. 25, 2001) it also posted 18 *megabytes* of patches on its website, including bug fixes, enhancements and compatibility updates. Two patches were issued for security holes; one worked, one didn’t. To play it safe, Microsoft advised users of the home version to back-up files. But there was no way, if things went bad, to restore these backup files. The XP release did not work for backup.

Software is critical to nearly all the products and services we use today. Hitting “Restart” or waiting for a telecommunications patch are not solutions.

A perfect example of this is a Cisco update (not released) that came out in March 2001. When the update was installed, it terminated the operation of “Call Pickup” on the CallManager.

According to Watts S. Humphrey of Carnegie Mellon University’s Software Engineering

Institute, "Software's simply terrible today, and it's getting worse all the time." Good software would be "usable, reliable, defect free, cost effective and maintainable. And software is none of these things. You can't take something out of the box and know it's going to work." Not everyone shares this dire view, but all agree that software needs to be better.

Calling The Doctor About Bad Code

There are many ways to reduce software problems. The operating system you choose makes a big difference. VxWorks is designed for high availability; it is used in aircraft, cell phones and thousands of devices where reboot is not an option. Both Linux and Unix have improved their reliability over the years, but reliability and security appear to have been low on the list of goals for Windows until recently.

One solution is for the vendors to better manage how software is written, tested and revised. The management standard, ISO 9000, defines processes used to develop and support products and services (see "ISO 9000," pp. 52). Is your hardware and software vendor certified by an independent registrar or compliant (self-certified)? Does the vendor even support ISO 9000?

Additional measures can be taken by vendors, although customers may have a hard time verifying if some of these steps actually have been implemented:

1. Software developers should have much more input from system administrators, staff charged with software maintenance and end users. In short, we—customers—need to speak up, loudly and continuously, to the vendor.

2. Stability in specification equals stability in product. Is the software based upon a stable specification or have many people put in their two cents after software developments started?

3. Are programming tools, or languages like Ada that prevent errors and/or weed out common problems, used in the development of software? The tools eliminate or prevent the compilation of bad code. What does your vendor do?

4. The vendor as well as the user should track revisions so that significantly rewritten sections—which usually include many errors—are thoroughly inspected and regression tested with each issue before installation.

Is Software A "Culture" Or A Technology?

"Software is different," is the mantra for some. Maybe, but we shouldn't have to lower our requirements to meet what the software can deliver. You design stability and reliability into the version/release/update, you don't attach it later.

I worked on a system designed for the Federal Bank of New York, which had a requirement for a one-minute restart time for all software, all transactions in process and all accounts. Our worst case was a 55-second restart, and we did that in 1976! We designed recovery as we designed the applications, not afterwards.

In some cases, software providers reportedly prevent public discussion of their product stability and reliability. According to the *Technology Review* article cited above, Oracle used its license terms to block the results of benchmark tests performed by *PC Magazine*. Customers must promise not to publish comments or reviews of Network Associates (NA) McAfee Virus Scan software without prior consent, a requirement that plaintiffs before the New York State Court call an "illegal…restrictive covenant" that "chills free speech." It has not yet come to trial. The conclusion of the *Technology Review* is that "the lawsuits will eventually come, and when the costs of litigation go up enough, companies will be motivated to bulletproof their code." When a non-disclosure restriction is part of the software license, we have to wonder what the vendor is hiding. It's the culture, not really the technology.

What can you do? Push, press, hound, complain to your vendor, especially to the marketing and sales personnel. The message will get back to the designers. You may even refuse to buy more of their product. Go public with the information, or at least threaten to do so. Vendors will improve their code once the public relations gun is put to their head□

Stability and reliability should be designed into a product—not attached later

Companies Mentioned In This Article

3Com (www.3com.com)
Advanced Micro Devices (www.amd.com)
Alcatel (www.ind.alcatel.com)
Avaya (www.avaya.com)
Carnegie Mellon University's Software Engineering Institute (www.sei.cmu.edu)
Cisco (www.cisco.com)
HP (www.hp.com)
IBM (www.ibm.com)
Intel (www.intel.com)
Juniper (www.juniper.net)
Microsoft (www.microsoft.com)
Mitel (www.mitel.com)
MontaVista (www.mvista.com)
NEC (www.cng.nec.com)
Network Associates (www.nai.com)
New Century Mortgage (www.newcenturymortgage.com)
Nokia (www.nokia.com)
Nortel (www.nortelnetworks.com)
Novell (www.novell.com)
Open Source Development Lab (www.osdl.org)
RedHat (www.redhat.com)
SuSE International (www.suse.com)
Wind River Systems (www.windriver.com)