“Can we talk?”

L. Bernstein
AT&T Bell Laboratories
184 Liberty Corner Road
Warren, NJ 07059 USA
fax 908-580-4580
lbernstein@attmail.com

C.M. Yuhas
Freelance Writer
4 Marion Ave.
Short Hills, NJ 07078 USA

Abstract

The successful integration of Services, System and Network Management depends on teamwork among technicians who have trouble understanding each other. The special problems of system configuration and software reliability are addressed in the context of their implications on Services Management. A vision for the future management of complex distributed services is offered.

Keywords

Software, network management, systems management, services management, configuration

INTRODUCTION

The concept of integrated network management sounds like such a good idea. There is the illusion that such a thing exists--some people even come to symposiums to discuss it. Actually, our industry is on the brink of combining 3 types of management--system, network and services-- to achieve a totally integrated product. I've used Joan Rivers' classic line, "Can we talk?," as my title because I hoped the directness and honesty of her delivery would resonate with you. The answer to "Can we talk?" for our systems, for now, is no. Computer management systems, physical networks and service objectives use the same vocabulary to mean different things and to pursue different goals. Acknowledging the problem areas is the beginning of the solution.

Let's listen to a few of the major voices in this field. Here is Arno Penzias, Bell Labs Nobel Prize winner: "If a customer can't use it, it might as well be broken--it might as well not exist!" Clearly that is a statement of the ultimate service objective. It is the goal (albeit negatively stated) of all our work.

BITS VS BYTES

Now here is Charlotte Dennenberg, a Southern New England Telephone vice president: "The most beautiful network is one that is about to break from overuse." She is speaking from the viewpoint of telecommunications, a discipline devoted to getting bits from point A to point B reliably and efficiently. These people measure costs and manage networks to optimize bit-delivery performance. They worry about security and billing. They assume the bytes will make sense of themselves if the bits arrive intact. They juggle 4 networks: one carrying message bits, a second carrying alarms and measurements, a third to operate the network and a fourth bringing management data "outband" so that if the network fails, they can restore it or route around the failure.

Then there is Clarke Ryan, a Bell Labs vice president in network operations, who observes, "Every platform is a weak alternative to an optimal solution." His is a wide-ranging data networking perspective, focused on applications. People like Ryan see the other side. They create local area networks and establish client/server hierarchies to transfer bytes from terminal to application. They worry about balances between server and network performance. They have little enthusiasm for discussions of point-to-point switching and performance because the assumption is that of course the bits will certainly get there. In these terms, the computer system is up if one terminal can send to one printer, even if the other 999 terminals are down. To these folks, network management is "inband," with systems detecting errors for other people to isolate and correct.

Each of these viewpoints is absolutely valid, yet none is sufficient in itself. Is it any wonder that Vinton Cerf, the inventor of TCP, could remark, "Most applications have no idea what they need in network resources or how they need to be managed."? This difficulty is the main reason that large firms have not embraced distributed computing.

COMMUNICATION DIFFICULTIES

For an industry whose stock in trade is communication, we have extraordinary difficulty in talking to each other. Network managers, computer system managers and service managers each feel like Captain Picard on a Star Trek diplomatic mission to alien races.

Terms

The three management schools use the same terms to mean different things. When "disk full" is reported to a system manager, it means time to reorganize the disk use. To the network manager, it means overflow for network data. Systems managers use "fault management" to describe application anomalies. When a network manager says this, it means a box problem needs to be tracked down. Event notification, alarm distribution and logging are carried out differently.

Security

Security management presents real problems. The issues of authentication, authorization, security alarm reporting and audit trails need attention. We need to know when security is compromised. A hot topic today is the possibility of digital cash to pay for all these great system features. The problem is how to send such payment over a broadcast network without having one's credit card stolen. We cannot entirely prevent security breeches, so we had better be very good at detecting and tracking them.

Things will get even more complicated with SONET and SDH (Synchronous Digital Hierarchy) when we multiplex the control and network management data on separate channels within the same physical circuits as the messages and signals. After we understand the use of these isolated channels, we will be ready to use ATM (Asynchronous Transfer Mode) to multiplex all of these together on the same transport links, trusting decoders which will be built into routers to sort things out.

Congestion

Dr. Harry Heffes of Steven's Institute of Technology points out that traffic prediction techniques will be required to reserve bandwidth because the networks will not be able to react to surges quickly enough. Traffic jams could become horrible on the broadband networks. The "byte folks" from computer networks consider congestion management a passing cloud, while the telephony "bit folks" are consumed with avoiding it. Congestion is inadequately addressed in systems management [Vaughan94]. Messages describing component failures induced by congestion will quickly exhaust SNMP. Scalability is poor and "byte folks" need hands-on use of protocol analyzers to find and fix problems. This in itself is not bad, except for the time it takes and the custom-built arrangements required to get to the message paths so that protocol analysis can happen. This becomes a challenge for the "bit folks" to get the message streams nimbly to protocol analyzers.

Protocols

Let's examine the protocols used to convey the network management data and commands. Telephony people spend lots of effort standardizing on OSI agents and their managed objects. Client/server people use SNMP[Barbier94], but their managed objects are different. Since we have two types of network management servers and two types of networks, we will need four interfaces. Now add signaling and message networks of several varieties and watch the complexity grow. Unfortunately, we can't pick one way. The OSI base is just too expensive for simple networks and the SNMPv2 upgrade is not totally backward compatible. SNMPv2 adds security features, but they are too complicated and bulk retrieval of agent data can cause congestion.

MAJOR ISSUES

Configuration Management

Configuration management is a major issue. Reliable, cost effective and easy-to-use backup and restoral systems are needed. When it comes to software, we need to "pack it and track it" as it goes among servers, clients, managers and agents. The same holds true

for data. *Data*, not *database*, management will be the issue for these complex networks. How will we trigger work to begin once we install a new feature or recover from an outage? In 1985 when client/server systems were young, we did not know we were actually doing data management. The problem we faced was to broadcast work status information from a Unisys mainframe to thirty NCR tower clients throughout the day. A work center manager who wanted to know how much work was left could ask the local client. This was a nice feature most of the day, but became critical at 3 PM when all the managers wanted to know if they had to schedule overtime to close out the day. Before we had the client/server solution, they would jam the mainframe and networks with report requests, generations and transmittals. And each wanted only that work related to their technicians. To keep the clients in step with the mainframe server, we provided an initial report whenever the client came on-line. This obviated the need for a separate record of the state of each client because we relied on the clients to customize the one comprehensive report for each work center. The server would broadcast all changes to all clients. By using this approach, we did not have to resort to complex startup and recovery procedures that cost network and server capacity. Today, better solutions elude us except for client/server systems that do not grow too fast in size or capability. Static mapping of data models or of software executables will not be good enough to handle future applications. Who will be charged with keeping this complex of systems operating sanely?

Auto-discovery of clients on a TCP/IP network has been wonderful, but this feature does not scale and has been used sparingly in telephony applications. A generalized auto-discovery feature is needed which embraces the concept that the network is the data base[Caruso1994]. It is also needed at the services level. Recently, Hewlett-Packard added centralized software to its Open View platform which manages configurations across networked systems. Instead of making multiple changes every time a user is added to the network, the administrator can use one command to configure the user's password, e-mail account and downloaded software. HP relies on a "synchronization" function to true-up the actual state of the networked applications and computers with the administrator's databases. The problem of mixing the network's physical inventory with logical data in UNIX databases is a tough one. For example, when we built a prototype to extract information from a network element and write it to a relational data base, the hardest part was getting the client protocol stack just right, especially in its interaction with the server's relational database. The database demanded versions of the protocol stack which could not be purchased for the client. The devil is in the details! Once we got the specific configuration to work, it worked well, but it is not robust to changes in the client or the server.

Software

Software may be the toughest problem to solve in building systems that manage other systems. Software has the awful propensity to fail with no warning. One manager of my acquaintance issued a memo stating, "There will be no more software bugs!" The trouble was he meant it; no joke. Even after we find and fix a bug, how do we restore the software to a known state, one where we have tested its operation? For most systems, this is impossible except with lots of custom design which is itself error-prone[Ross94].

One new idea is software rejuvenation. It is special software that gracefully terminates an application and immediately restarts it at a known, clean, internal state. It precedes failure, anticipates it and avoids it. It transforms non-stationary, random processes into stationary ones. Instead of running for a year, with all the mysteries that untried time expanses can harbor, a system is run for one day, 364 times. It is re-initialized each day, process by process, while the system continues to operate. Increasing the rejuvenation rate reduces the cost of downtime. Two years of operation have passed with no reported outages for one system set at a weekly rejuvenation interval. In one of my laboratories, a 16,000 line C program with notoriously leaky memory failed after 52 iterations. We added seven lines of rejuvenation code with the period set at 15 and the program ran flawlessly. Rejuvenation does not remove bugs, it avoids them.

SYNTHESIS

How will we bring all this together? We need to adopt a policy of "agreeing to agree before we disagree" and drop our current practice of "agreeing to disagree before we agree." Cooperation and harmony among practitioners in this field are the only hope. With customers demanding that the power of these services be unleashed, we can either make it happen together or watch others without our know-how create a new industry of service managers under our noses. Professor Ed Richter of the University of Southern California points out, "The more technically competent an organization, the harder it is to accept new technology." If we are not careful, novices will capture our industry while we squabble. We need to understand that working together improves each one's lot. There are technologies to help if we will only use them.

Photonics

We can take advantage of photonics to combine service testing and surveillance. Since photons at different frequencies do not interfere with one another as electrons do, we can send test signals at one color while customer data is being carried at another color and measure performance or monitor a system for alarms. This will lead to true non-interference testing. Len Cohen of AT&T Bell Labs Research is developing a photonic chip that will make this easy to implement[Bernstein94].

Human Factors Design

IBM seems to be leading us to friendlier systems. They will deploy standard object based technology to bind together current products and add new functions in distributed systems management encompassing network, configuration, change, performance, and storage. Their approach differs from the TMN (Telecommunications Management Network) standard by including system administration functions and by giving attention to system usability. The TMN standard is not natural for people. I have seen network managers need to combine and partition TMN functions to solve a problem. We invented "quick switch" in the 1970's to break the straight jacket of hierarchical application system design. We need something similar between TMN functions today. Will they set yet another de facto standard as we argue about the de jure ones? Bev Little commented in the April 25, 1994 issue of Forbes that "Today's systems have good facilities for humans to take over,

the ones coming down the line are supposed to handle higher traffic flows, and it is not obvious that humans could intervene." Whether IBM's approach will solve this problem and be scalable remains an open question.

Who Manages Whom

The question of whether system management is part of network management or vice versa will be hotly debated in the coming year. My view is that the network extends from my finger tip to the application executable code, so that even queue management and caching control inside the servers are right and proper network management functions. As client/server systems get deployed, savings are not as great as expected because they are much more expensive to operate and evolve than expected. The "administrator to box" ratio remains constant because of the poor systems and network management design which is slowing down progress. Planners are not faced with a "buy or build?" decision since no solution solves the existing problems and none can be built fast enough to keep up with the demand. Administrators are often faced with the problem of integrating solutions from different suppliers. People are beginning to use the INTERNET for remote management to good effect. But, this is a "send and pray" network that cannot be counted on under stress to isolate and fix problems. It may prove to be perfectly fine for detecting problems, but will it be adequate to force test messages on defined paths, or to monitor major network events, or to isolate focused overloads, or to convey commands to network elements?

State Transition Theory

A theory of state transitions for systems and network management needs to be articulated to define how the network states will be re-established after the inevitable hang or crash. With a workable theory, the state transitions will become predictable and, more importantly, testable. One set of these states will be the orderly downloading of programs and data from servers to client with appropriate recovery strategies.

ATM Data

Current network management is inadequate for the coming flood of ATM data because of incompatible network management from different suppliers [Wilson94]. SNMP cannot scale well to handle the volumes required, such as automatic topology features with huge network domains. Protocol analyzers will not be able to get to the troubled transport in time. Data base "sniffers" will be incompatible with network analysis software. With computer communications and voice communications sharing the same network, we see a threefold increase in complexity. With the emerging multimedia, we are seeing ten times this complexity. A 30:1 increase in complexity is on the horizon for our overworked system administrators. Today AT&T's Accu-Ring Service Centers provide customers with a single point of contact for SNET dual fiber ring networks [Robinson94]. These networks span customer premises, local exchange carrier central offices, competitive access providers control centers and AT&T central offices. The service center coordinates repairs and manages installation, sectionalization, and alarm monitoring. Eventually this network management service will be extended beyond communications to the actual applications on the network.

THE FUTURE

Even with all these problems, I have hope--even enthusiasm--for the future. I envision a Services Management System that detects anomalies from any source. Rather than dump a glut of raw bits, this system digests the data and transforms it to information, presenting the network manager with only what's needed to resolve the problem. Picture a manager who would sit at a display well designed for easy human use. One could see all elements of service operation from server queues to transport error seconds. The manager would extend that famous finger tip and point to any service, system or network element. Up would come graphics of decoded messages being passed or filling a buffer. The manager would take performance measurements, then instruct "service" to make changes in its operation. Finally, without stirring from the display, the manager would inject a test message into the application and trace it through the distributed system, either by allowing it to choose its path or by forcing one. This will lead to integration of boxes, links, elements, queues, utility software and data managers in a friendly, easy way.

REFERENCES

- Barbier, Susan "Systems Management in the 1990s," AT&T Technical Journal, ..July/August 1994 Vol. 73 No. 4, ISSN 8756-2324 pp. 82-97
- Bernstein, Lawrence "Innovative Technologies for Preventing Network Outages" AT&T ..Technical Journal, July/August 1993 Vol. 72 No. 4, ISSN 8756-2324 pp. 9-10
- Caruso, Jeff(1994) , "UB Networks Embeds Network Management In The Hub," ..Network Management Systems & Strategies Vol. 6, No. 21, ISSN 1043-1217
- Robinson, Michael J., Werner, Thomas S., Ward Joseph A(1994)., "Solving the Access ..Outage Problem," AT&T Technology Vol. 9 No. Fall, ISSN 0889-8979, pg. 21
- Ross, Philip E.(1994), "The day the software crashed," Forbes Magazine April 25, pp. ..142-156
- Vaughan, Nick(1994), "Networks and Systems Still Seeking Suitable Frameworks," ..Software Magazine October pp. 53-58
- Wilson, Tim(1994), "Current Net Mgm't Inadequate for ATM," Communications Week ..November 7, No. 530 A CMP publication

BIOGRAPHIES

Lawrence Bernstein is Chief Technical Officer of the Operations Systems Business Unit at AT&T Bell Laboratories. He holds a BEE from RPI and a MEE from NYU. He has contributed to the evolution of Network Management and is a Fellow of ACM, IEEE, and Ball State. He is listed in Who's Who in America.

C. M. Yuhas is a freelance writer who has published articles on network management in IEEE Journal on Selected Areas in Communication, IEEE Network and International Journal of Systems and Network Management. She has a Bachelor's in English from Douglas and a Master's in Communications from NYU.