Remote Outsourcing Services for Multiple Branch Offices and Small Businesses via the Internet

Dejan Diklic, Venkatesh Velayutham, Steve Welch, Roger Williams
– IBM Almaden Research Center

ABSTRACT

Maintaining a reliable computing infrastructure encompassing servers, clients, printers and Internet connections is one of the major problems faced by branch office managers and small business owners. The cost associated with good systems administration is too high for either small businesses or branch offices. At the same time large service corporations don't see this market as attractive due to the high startup and infrastructure cost. We at IBM Almaden Research Center developed a research prototype that enables remote outsourcing services for multiple small offices from a central location via the Internet using the already available commercial software. This technology provides a solution for both small business owners (branch office managers) and service providers. We use the term small business for any company with one to 100 client machines. For our purposes branch offices can be categorized as small businesses. By combining our newly developed systems management technologies with an existing Internet connectivity server, small office LANs can be remotely monitored and managed via the Internet. At the same time the need for expensive leased lines and dedicated routers for secure connectivity is removed since both routing and security are provided through software. In this paper we will describe the approach taken as well as utilized software and hardware components.

Now that we described the target environment we introduce all of the components of the solution. In the second section we describe implementation details. Short summary of the work is given at the end of the article.

Introduction

Most of readily available network management applications such as Tivoli ITDirector and HP Open View [1, 2] provide all usual computer management functions such as: remote control, software/hardware inventory, task scheduling, sending alerts, logging events and software distribution. When combined together these functions represent a set of tools that is used by various corporations to manage their internal network with thousands and tens of thousands of clients [4, 7]. These management applications are installed on a server which is on the same network as the client machines they manage [5]. To put it simply, "if you can ping the client you can manage it." While this works for internal networks of large corporations it is not usable if the client machine is connected on a different subnet and/or hidden behind the firewall inside a small business or branch office. The main goal of our research project was developing software extensions to regular off-the-shelf network management applications which would enable management of client machines over untrusted networks such as the internet.

The approach taken with the purpose of addressing these issues started from a very simple idea. If we can use one server located at the central network operations center (NOC) of the service provider to manage multiple small businesses (without a server at the customer site) the cost for managing each business will be much lower then when using a regular management model described above. Another issues addressed by this model are the ease-of-use for the customer and the ability to outsource system management. By utilizing this model management of IP-disjoint pools becomes possible. This approach applies to both small businesses and large corporations that use firewalls on the Intranet to separate networks. Another advantage of the new approach is that a service organization will not have to own or maintain the server and server software at customer premises but only at the central location. Most small businesses can not afford the cost of a management server, but they can afford the clients and associated software.

In Figure 1 we present the view of the network architecture as is typically seen by the network management providers [6, 8]. Management server which is located on the side of corporate Intranet is usually a very powerful machine. On the other side of the network (Internet) are multiple small businesses and branch offices that need management services. Small businesses or branch offices are typically setup to run their own networks using non registered private IP addresses. They are connected to the Internet through an Internet gateway which can be anything from a regular PC to a small business server such as IBM Whistle InterJet II, Cobalt Qube, etc. We assume that both the service provider and the small businesses are isolated from the Internet by a firewall.

This architecture makes it almost impossible to remotely outsource to small businesses and branch offices [11]. Since most of the SMBs are using the same private IP address range, the management application has to not only traverse two firewalls, but has to access machines hidden on the private network. As part of this project, we set out to develop a new network protocol with an advanced addressing schema which would allow us to traverse multiple networks, firewalls and duplicate IP addresses while at the same time keeping the system secure.

We attacked several issues in order to solve the problems described above. The first issue was that of addressing different machines with the same IP address on different private networks. To solve it we developed a novel form of network address which enables us to both access the machine at the private network behind the internet gateway and at the same time keep the network infrastructure secure from outside tampering. In addition to that packets are encrypted across the untrusted environment and delivered through a firewall to the addressed machine without compromising the small businesses network security. The next challenge was related to changing as little as possible of the original management software. Due to the architecture of the software used (ITDirector) this was rather easy to achieve. Another requirement was that new system has to use existing holes in the firewall and existing hardware at remote sites (no new hardware required).

Solution

In the next three chapters we describe the main idea behind the solution in more detail. First we describe the extended network addresses needed to address all of the machines in disjoint IP pools. Then we describe newly developed (java-based) proxy which runs on the Internet gateway machines. The last chapter is dedicated to security problems inherent in a system like this.

Network Address

The first part of the solution involved changing the network management protocol so that more complex network addressing schemes could be used. To maintain the simplicity of the SMBs LAN, private dynamic addressing schemes were widely used. Therefore, regular four byte IP address can not be used to identify a particular machines in the private addressing space. We devised the way to address the target machine by incorporating the IP address of the Internet gateway.

`$CH(`	`A.B.C.D`	`)X.Y.Z.Q`

Figure 2: New network address format.

While the four byte IP address will suffice once the packet is delivered to the internet gateway, we need a way to address at the same time both the internet gateway and the client machine. The new format of a new network address is shown in Figure 2.

The regular IP address of a machine on the private network (X.Y.Z.Q) is combined with the external

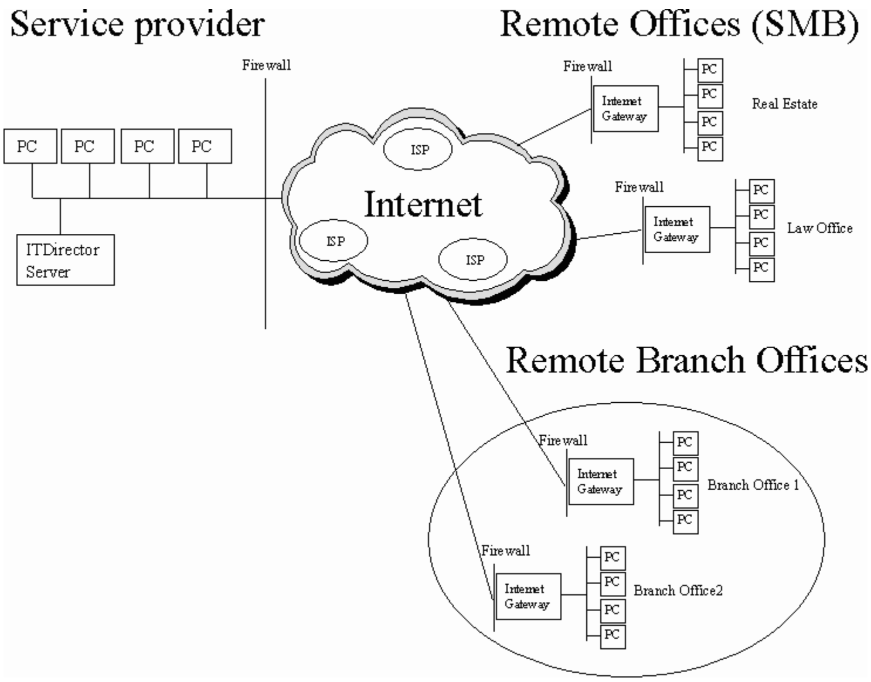

Figure 1: Typical network architecture.

registered address of the Internet gateway (A.B.C.D). This way, the packets are first delivered to the gateway with address A.B.C.D. Then they are forwarded to the private network to the machine with the address X.Y.Z.Q. While the address of the from shown in Figure 2 is user readable, it is not used in such a form in the network protocol to reduce overhead. We encode the addresses as an eight byte array at the end of the network packet. This minimized the overhead associated with the sending of every packet when compared to using the address shown above. At the same time the processing time required for packet forwarding is minimized. When this new complex address form is introduced, it becomes possible to address machines that are more then one network hop away from the server. Any extensions of this type of network addresses are easily done. If we need to deliver a packet to a machine which is three hops away from the server, extending the protocol would be just a matter of attaching four more bytes to the end of the package.

Gateway Proxy

Having solved the problem of addressing client machines behind the Internet gateway machine, the problem of delivering packets to a client machine behind the firewall needs to be attacked. There are several options available. The fist one that comes to mind if also the simplest one. It requires a hole to be opened in the firewall of the Internet gateway machine. Then a proxy application can be installed on the gateway machine that will accept packets coming from the newly opened hole in the firewall. Both the packets originating at the server and at the client will be directed to the same port and then consequently forwarded to their destination by the proxy application running on the gateway. Most of the small businesses are however not willing to expose themselves any more then necessary and are highly critical of opening holes in the firewall. Therefore we developed an alternative method to traverse firewalls. We made several assumptions about small businesses. We assume that they are connected to the Internet and that the gateway machines come installed with a web server capable of running CGI scripts [7, 12]. In order to pass the packet through the firewall we decided to pass the packets through the web server using our novel CGI script. While this way of distributing packets is rather slow, when compared to the dedicated application sitting on a dedicated open port, it enables us to solve this problem without needing customers permission to open an additional hole in the firewall. Any communication between the server and CGI application and client and CGI application is performed using standard GET and POST methods. To pass network packets we developed an application that registers itself to a predefined URL such as http://A.B.C.D/my_cgi_app where A.B.C.D is the address of the Internet gateway device. The CGI application will receive packets coming from the Internet (server) and then based on the address of the private machine which is read from the network packet itself, forward the packet accordingly. The end-clients address is stripped off from the packet before it is forwarded so the client can process it. This proxy has several different functions. The most obvious one, which is forwarding packets from the server to the client and vice versa was described above. The second function is described below.

The Internet gateway machine which is running the proxy application has to be detected from the management server for the system to be fully usable. The detection mechanism built in network management applications can be rather complex. We tried several different approaches to implementing the detection mechanism in the proxy application. The first approach was to try using hard coded responses to discovery packets. This was too complicated and resulted in unreliably functioning proxy. The second approach yielded much better results. Since the code for the java based agents is available, it is possible to add the forwarding and complex network address thereby effectively creating a detectable proxy that forwards packets and understands the long network address.

Security

One of the problems faced by service providers is ensuring security and confidentiality of data while remotely outsourcing the customers devices across an untrusted network (Internet). We added security to remote outsourcing by establishing an encrypted session (Virtual Private Network) [5, 9, 10] between the service provider and the internet gateway. We use SSL and client side certificates to make sure that nobody else but authorized users can connect to the gateway proxy [3, 14]. Since the Internet is not a trusted network, the remote session management data and protocol is sent through the VPN [13, 15]. After the packet is delivered to the proxy machine it is decrypted and forwarded across the local area network to the end-client(s). Since the local area network of the small business or the branch office is trusted, the data is sent unencrypted. Our design also facilitates the packet sent encrypted all the way to the end client. This addition to the system is essential since remote outsourcing could result in having confidential data exposed to the outside world and prove damaging to most businesses.

The biggest problem when dealing with encryption is the need for powerful processors that can complete encryption and decryption of large amounts of data in a very short time. If the processor is slow the encryption/decryption will take a long time and the whole remote management process is endangered since the operator can't perform tasks in the real time. A delay of several seconds is not acceptable when performing base operations (like clicking the mouse button, or opening the window). There are several ways this problem can be solved. The easiest is to make sure that the processing capability of the Internet gateway is sufficient. This is highly expensive and it increases overhead added to each device. Another way is to add encryption acceleration hardware to the device which is a lot cheaper and equally effective. Additional area that could be easily addressed is the option of specifying tasks that should be performed using encryption and tasks that should be performed without encryption. For example if you are making a backup of financial data, you would want to be as secure as possible, but if you are just distributing the latest version of some software no encryption is necessary. The same holds true for regular everyday monitoring of system events. Encrypting data such as "Disk out of space" messages is in most cases not essential.

All the regular safety measures provided by ITDirector are still present in the new protocol. Users need to be registered to access all of the functions of the server and clients can be controlled only if the proper authentication is provided to the client (once the client is locked).

Implementation

Since there are several systems management applications on the market at the beginning of the project we debated pros and cons of IBM Tivoli ITDirector and HP Open View [1, 2], as basis for our prototype. We selected ITDirector since it has a rich set of utilities and functions to manage all aspects of the PC environment and is used and developed by IBM. ITDirector enables us to utilize the rich set of APIs and extend network protocol that allows us to connect to clients over the internet using the provided tool kit. ITDirector is shipped with various agents for various operating systems such as Windows NT/2000, 9X, Linux, AIX, OS/2 etc.

The agents for Linux and AIX are available in Java which makes it simple to use as basis for our proxy code. Since the proxy will run on a Internet gateway machine we prototyped our system on several different hardware platforms. Our first prototype was developed on a regular PC that is running Linux and is used as Internet gateway. This significantly simplified the development when compared to developing on a dedicated Internet gateway.

Once the first version was available we integrated and tested it with IBM's Whistle InterJet II and Cobalt Qube2. Whistle InterJet II is a thin server offered by IBM as an Internet gateway. InterJet II is a small footprint, easy to use and simple to administer server based on FreeBSD operating system which connects small businesses to the Internet and provides all of the functions of a file, web, mail, DHCP and DNS servers. Cobalt Qube2 provides the same functionality as IBM's device. To utilize InterJet II or Cobalt Qube2 we had to develop all the pieces needed to incorporate them into the ITDirector addressing and management schema.

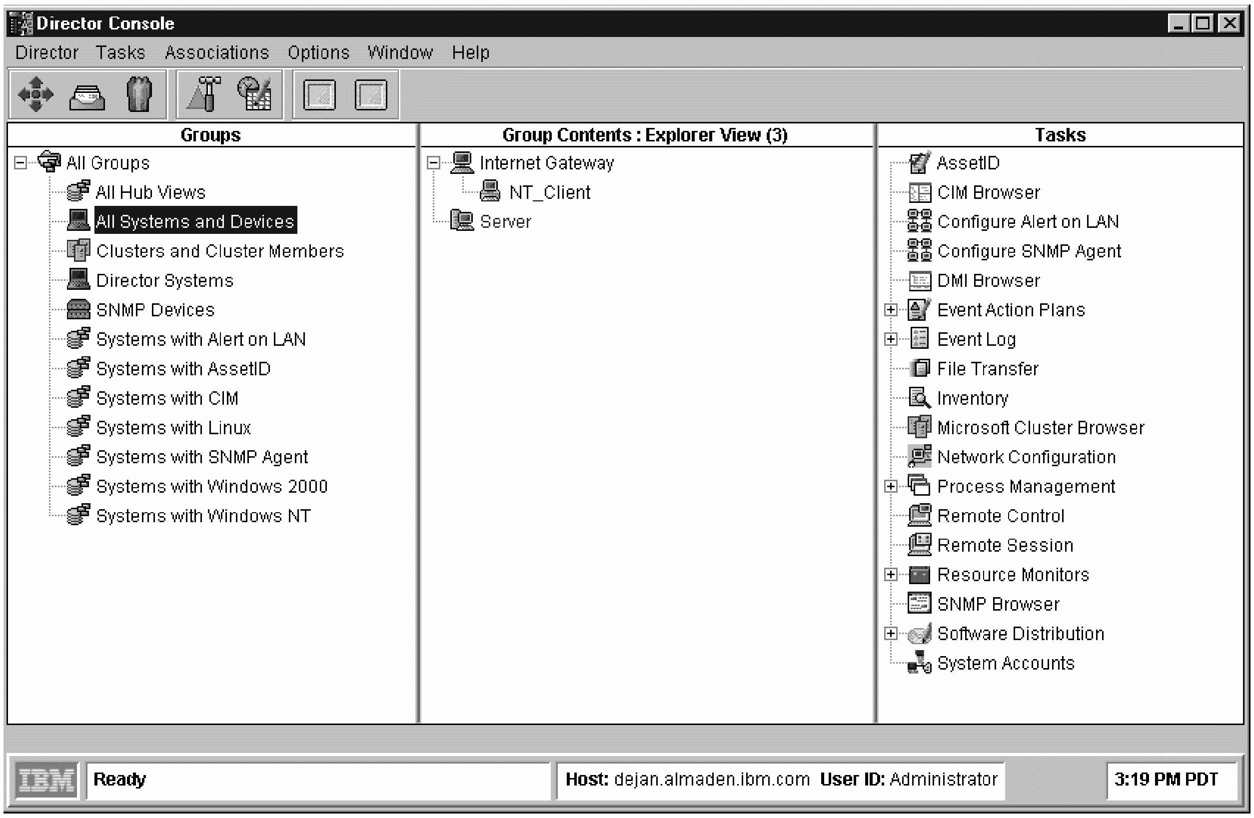

Figure 3: Tivoli TIDirector showing server side extensions.

The addition of the complex network addressing schema and security layer to the network protocol changes the complete network protocol layer of the regular Tivoli ITDirector. This rather complex and significant changes can be done due to the architecture of the product, which makes it possible to just change tcpip.net file which defines the complete network protocol. Security layer was added to the protocol through usage of OpenSSL. The java based Linux agent was the origin of the CGI proxy. Adding forwarding to it made it usable on the Internet. We added security to the java agents by using jsse-ssl. Once the whole system was tested to perform the desired tasks, several interesting observations and measurements were made.

During the performance testing phase we noticed that simply adding one more hop in the network protocol decreased the performance by less then 5%. Adding firewall traversal to the system increased the performance penalty to 8%. The biggest drop in performance however comes when using encryption over SSL. The drop is performance is almost 40%. This means that encryption should be aided by hardware acceleration on the Internet gateway. In Figure 3 is a screen capture of the Tivoli ITDirector screen which shows our extensions to the server side. In the middle column we see only three machines, the Tivoli server, the Whistle gateway, and the client. The presentation of machines on the screen was modified to include hierarchical drawings of associated gateway and clients. In Figure 4. we see the new address of the client machine in the complex form. In Figure 5 we show the grouping of all hubs (Internet gateway machines). This view is very useful since it displays only proxy gateway machines and the clients behind them.

Future Work

At the moment there are several interesting developments going on with this project. The most important one that we are pursuing is software distribution to multiple machines from a central fan-out site (Internet gateway). The idea behind this is that you really need to only once distribute the software packet to the Internet gateway and it can then redistribute the packet to as many clients on the local area network as necessary. This speeds up the process considerably since the big packet is transferred only once over the slow Internet (application specific compression of data). At the same time we are working on speeding up the packet transfer and minimizing the forwarding (processing) time of every packet on the proxy gateway.

Summary

In this paper we presented an extension to Tivoli ITDirector that enables service providers to use ITDirector as the application of choice for managing multiple small and medium businesses from a centralized location. We presented the technical enhancements made to the ITDirector network protocol and the newly developed java proxy for Whistle InterJet II. By

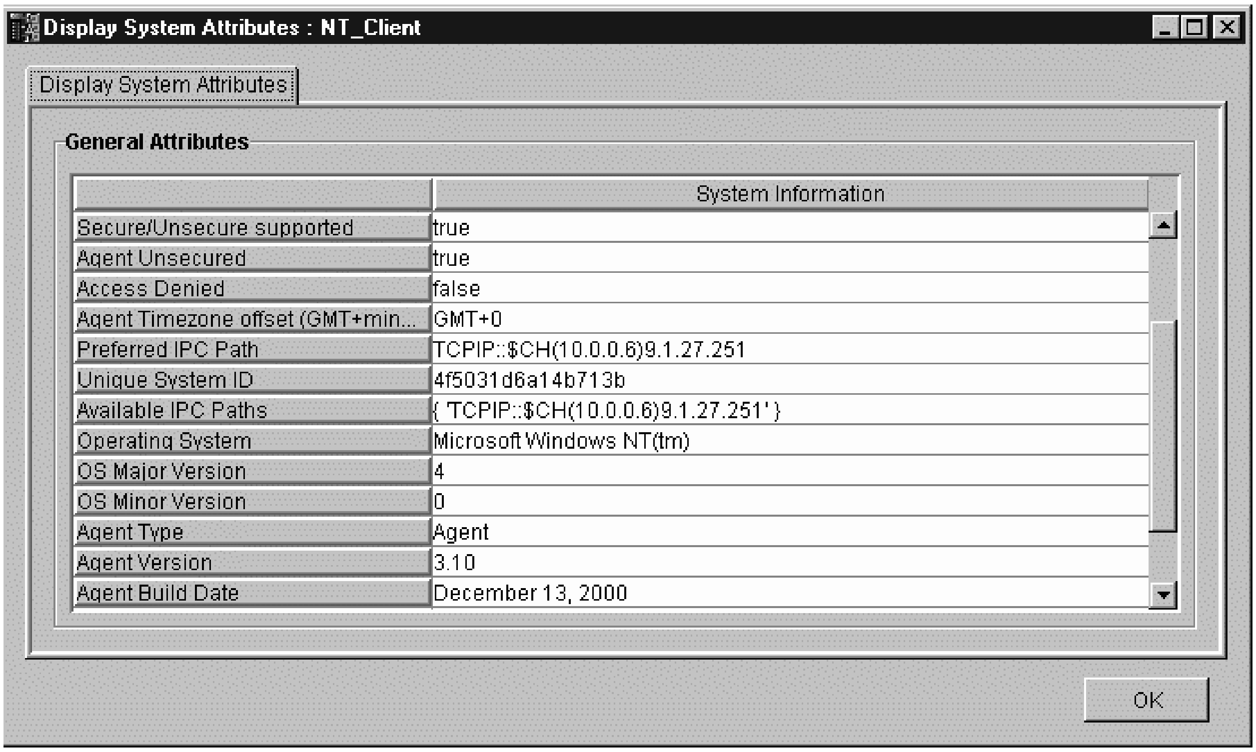

Figure 4: New address of client machine in complex form.

modifying the network protocol by adding one more step in network packet distribution as well as security in form of client side certificates and encryption we proved that the technology for managing multiple small and medium business from a central location can be achieved.

Author Information

Dejan Diklic has a MS in physics from University of Bremen, Germany and MS in computer science from Santa Clara University, USA. He is currently employed at IBM Almaden Research Center where he works on various problems related to systems management and storage. He can be reached at ddiklic@almaden.ibm.com .

Benjamin Reed has a PhD in Computer Science from the University of California, Santa Cruz. He works at IBM Almaden Research Center in the area of network attached storage, pervasive computer systems, and system management. He can be reached at breed@almaden.ibm.com .

Venkatesh Velayutham has a BS in Mechanical Engineering from Anna University, Chennai, India. He is currently employed at IBM Almaden Research Center where he works on various problems related to network management and storage. He can be reached at vvelayutham@almaden.ibm.com .

Steve Welch completed BS degree in CS from Cal-Poly San Luis Obispo. He is a manager of Cyberspace Technologies at IBM Almaden Research Center focusing on development of system management and Internet technologies. He can be reached at swelch@almaden.ibm.com .

References

[1] Tivoli Web site, http://www.tivoli.com/products/index/it-director/ .

[2] HP Open View web site, http://managementsoftware.hp.com .

[3] D.Zeltserman, G.Puoplo, *Building Network Management Tools with Tcl/Tk,* Prentice Hall, 1999.

[4] Mark Burgess, *Principles of Network and System Administration*, John Wiley & Sons, Chichester, 2000.

[5] S. Brown, *Implementing Virtual Private Networks*, McGraw Hill, 1999.

[6] M. Subramanian, *Network Management: Principles and Practice*, Addison Wesley, 2000.

[7] T. C. Mann-Rubinson, K. Terplan, *Network Design: Management and Technical Perspectives*, CRC Press, 1998.

[8] K. Terplan, S. Zamir, *Web-Based Systems and Network Management*, CRC Press, 1999.

[9] Peter D. Rhodes, *Building a Network: How to Specify, Design, Procure, and Install a Corporate LAN*, McGraw-Hill.

[10] H. Hegering, S. Abeck, B. Neumair, *Integrated Management of Networked Systems: Concepts,*

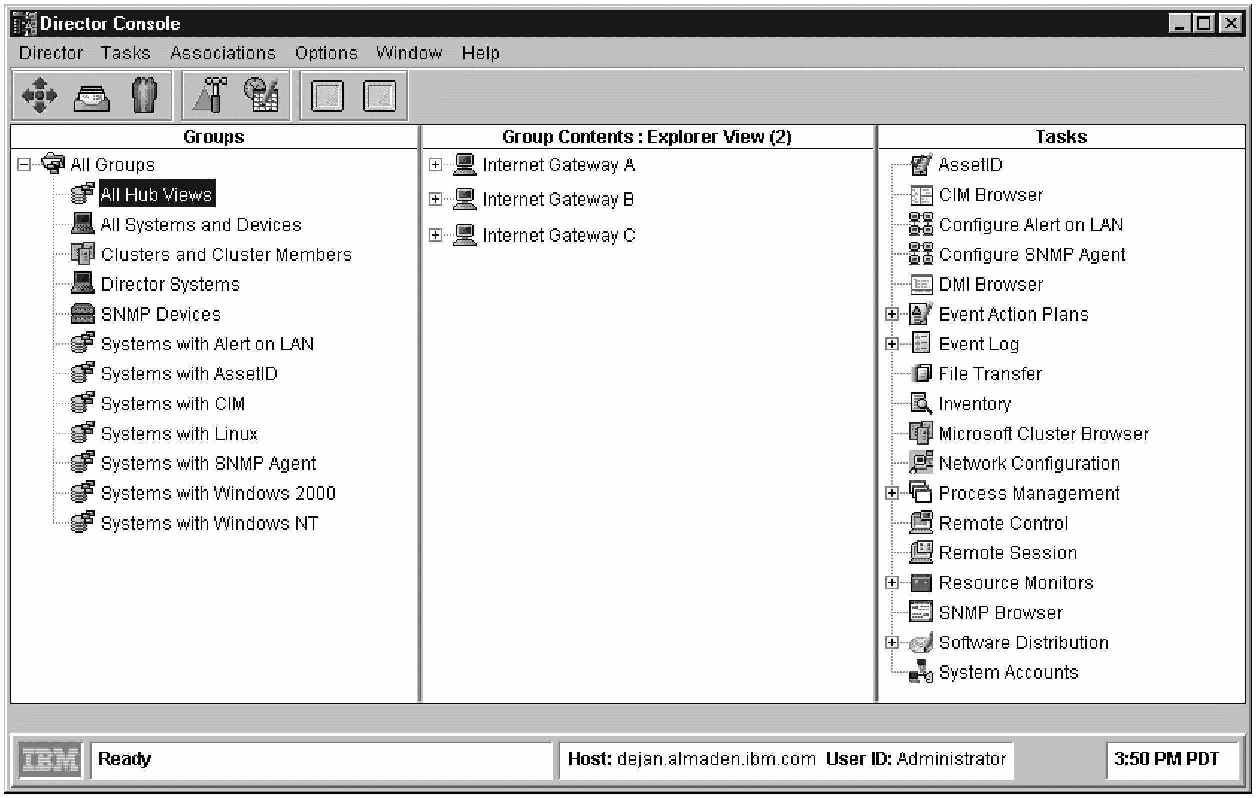

Figure 5: Grouping of all hubs.

Architectures, and Their Operational Application.

[11] M. Subramanian, *Network Management: Principles and Practice*, Addison Wesley, 2000.

[12] K. Terplan, S. Zamir, *Web-Based Systems and Network Management*, CRC Press, 1999.

[13] *The Basics Book of OSI and Network Management by Motorola Codex*, Addison Wesley, 1999.

[14] G. Held, *Managing TCP/IP Networks: Techniques, Tools and Security*, John Willey & Sons, 2000.

[15] A. Leinwand, K. Fang-Conroy, T. Stone, *Network Management*, Addison Wesley, 1996.