AN OBJECT-ORIENTED FRAMEWORK FOR THE INTEGRATION OF NETWORK CONFIGURATION OPERATIONS

Richard B. Clendenning

Indiana State University,
USA

Abstract

Many current network management systems are highly capable, but are vendor-specific and are not capable of configuring network devices from other vendors. For smaller networks, the purchase price of multiple highly capable management systems can be very prohibitive. This paper presents a framework for a network management system that provides for basic device configuration and that can be expanded to interoperate with devices and operating systems from different vendors. Initially, the system supports basic Windows Active Directory and New Technology File System (NTFS) operations as well as Dynamic Host Control Protocol and Cisco switch configuration tasks. However, the object-oriented nature of this open source framework allows new modules to be added easily for configuration of additional devices and operating systems. The system reduces the time for common configuration tasks, reduces configuration errors, and saves the cost of expensive management systems.

I. INTRODUCTION

As networks in all types of organizations grow and become more capable, the complexity of managing these networks rises dramatically. This complexity has prompted the development of management systems that provide a centralized interface to manage many network devices or hosts that may be geographically dispersed. Many of these management systems are provided by the vendor of the devices or operating systems being managed. For example, Microsoft offers System Center Configuration Manager 2007 (formerly Systems Management Server), a highly capable product that helps administrators to utilize and manage the many features of Windows operating systems and companion features (Microsoft Corporation, 2008). Another example is Cisco's CiscoWorks, which provides centralized management of a Cisco internetwork (Cisco Systems, Inc., 2007). However, these highly capable systems cannot configure operating systems or devices from another vendor since the features that each vendor offers, and how these features work, can vary greatly. The result is that for configuration tasks the administrator must either purchase and work with a set of management systems, or work without management systems and configure devices directly.

Large networks, and especially those which are being used in unusual ways, may well need the complete feature sets that highly capable vendor-specific management systems provide. However, for many typical smaller to mid-sized networks, the network administrator cannot afford an expensive management system for each set of products in the network. Yet the administrator still needs some affordable management capabilities to perform common configuration tasks and to query for configuration information. Ideally these capabilities would be provided via one integrated system that can interact with different server and host operating systems as well as network devices from different vendors, so as to avoid the trouble of setting up, maintaining, and learning multiple management systems.

This paper sets forth an Integrated Network Configuration System (INCS) that in its initial phase will only perform basic configuration tasks related to Microsoft Windows domain users and groups, NTFS permissions, Dynamic Host Control Protocol (DHCP) leases, and Cisco switch ports. Although its functionality is limited compared to that of a combination of expensive, vendor-specific management systems, these basic functions may still cover most of the network configuration that is performed on a regular basis for a small to medium-sized network. A major design goal of INCS is extensibility, so that new devices and operating systems can be added with a minimum of programming effort. INCS is not intended to be a complete solution for any particular network. It is a beginning template that starts with basic Windows- and Cisco-related capabilities that can be easily extended to perform most of the commonly performed tasks in any small to medium-sized network. With INCS, an administrator can perform these tasks while remaining in the INCS system, without the need to access multiple expensive management packages that do not interact with each other.

II. OVERVIEW OF PRIOR RESEARCH

Although automation of system and network configuration processes has been a major focus of research, much of this research has focused on administering UNIX-based systems rather than on configuring and monitoring local area networks (LANs) with Windows-based hosts. For example, Libes (1990) developed the *expect* program to automate tasks that would otherwise require interactive use of UNIX utilities such as passwd or fsck. Nevertheless, some elements of INCS derive from concepts pioneered from these earlier systems. Anderson (1993) describes using a centralized database to store machine configuration information so that new Solaris machines can be brought online more quickly and easily, and to ensure that policies (which are maintained in the configurations) are being enforced. Burgess (1995) created cfengine to automate frequently-performed system administration tasks. cfengine is a language-based system that stores the configuration of all networked hosts in a central data file, and it runs on a number of UNIX platforms. While cfengine "hides the differences between different operating systems" (Burgess, 1995), only UNIX-like operating systems are intended. Other configuration systems have attempted to deal with a wider variety of systems, such as the Accountworks system (Arnold, 1998) which was developed at Sybase, Inc. so that new hires could automatically be granted "a personal account in SQL, Notes, NT, and UNIX administrative domains". However, Accountworks did not address network configuration and it is not freely available.

While a lot of research has focused on systems management, some researchers have concentrated on the network itself. Agrawal et al. (2007) propose a system for monitoring converged networks, but they do not address configuration and they are chiefly concerned with service provider networks and not local area networks. The need for system integration and an increased level of automation is the basis of the work of Joshi, Riley, Schneider, and Tan (2007), who propose integrating a number of IBM Process Managers (PMs) to improve process flow for large networks. By contrast, INCS aims to do away with the need for expensive management systems such as the IBM PMs and instead provide direct management of the network devices and systems for small to medium-sized networks.

III. THE INCS SYSTEM

High-Level Description

INCS is a modular system consisting of a user interface, a database, a main distribution module and sub-modules that provide interfaces to the database and to the network hosts and devices that are being configured. Fig. 1 provides a high-level block diagram illustrating the major components of the system.

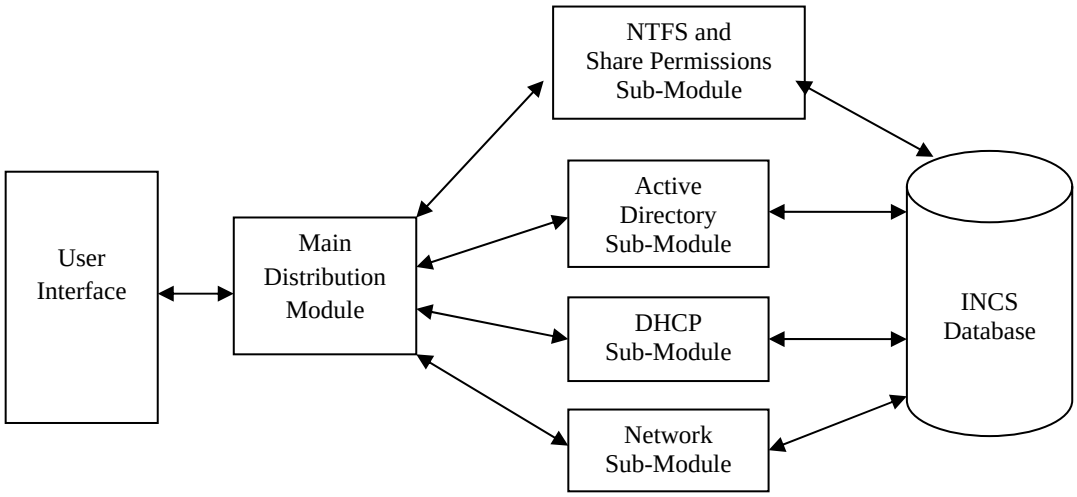

Fig. 1. High-level functional block diagram of INCS

The user interacts with well-designed windows and menus that provide access to the various functions of the system. The Main Distribution Module accepts requests and distributes them to the appropriate sub-module, or in some cases, to multiple sub-modules. For example, a request to turn on a switch port and assign it to a virtual LAN (VLAN) would be assigned to the Network Sub-Module. The sub-module communicates with the database to retrieve additional information needed to fulfill the request. For example, if the Main Distribution Module instructs the Network Module to turn on a particular host's switch port, the Network Sub-Module queries the database to discover which port belongs to that host.

The sub-module then passes the command, with the additional information, to a lower-level module (not shown in Figure 1), which may fulfill the request or pass the command to a still lower-level module. The highly modular nature of the code

ensures that adding a device or function requires a minimum of additional code. The additional code to handle a new device is provided in the form of a low-level module, similar in concept to a device driver in a computer system. Extending the system in this modular fashion helps ensure that additional code does not jeopardize the existing system. The lowest level modules communicate the specific configuration commands to the devices and handle the responses.

INCS Sub-Modules

The four INCS sub-modules are briefly described below:

– The Network Sub-Module is responsible for enabling and disabling user ports, assigning ports to Virtual Local Area Networks (VLANs), and other network functions (such as port security) that must be carried out by network switches.

– The Active Directory Sub-Module is responsible for creating and deleting Windows users and groups, assigning users to groups and removing them from groups, and other Active Directory functions.

– The NTFS and Share Permissions Sub-Module is responsible for reporting on the NTFS permissions and share permissions that are currently in place for a particular file or folder, as well as assigning file/folder permissions to groups and removing these permissions.

– The DHCP Sub-Module is responsible for reporting on information related to IP addresses in use on the network, such as identifying the computer and user to which an IP address has been assigned. This module is also used to assign static IP addresses to computers and to make IP address reservations.

INCS Functionality Illustrated

Although adding a new user to the network is simple in concept, in practice this task can involve many sub-tasks and quite a number of different systems. To provide a brief illustration of the functionality of INCS, three common and related sub-tasks are considered in this section: adding a new user to Windows Active Directory, creating a DHCP IP address reservation for the new user's computer, and configuring the new user's switch port. In a secure network, all managed switch ports should be turned off until they are needed, and when they are turned on, they should be configured with appropriate security measures, such as only allowing the user's computer and IP phone (if applicable) to access the network.

Without INCS, these tasks require accessing three different systems (and perhaps additional systems in order to look up information such as the computer's MAC address for the DHCP reservation and switch port configuration). However, INCS stores all the information needed so that the INCS user can accomplish all of these tasks within the INCS system. When adding a new user directly in Active Directory (without INCS), there are three different dialog boxes to traverse, the new user's password must be entered twice, and no options are selected by default. This is despite the fact that the initial password assigned a new user is generally a standard password and the option to "Require password change at first logon" is essential to ensuring that the user changes the standard password at first opportunity. By contrast, using INCS a new user is completely configured on one "New User" screen, and INCS allows an administrator to set default options so that a typical new user can be added only by typing in the user's full name and user ID, and accepting all the defaults (such as the standard password and "Require password change at first logon" option). Thus, not only is the process easier and faster, but it reduces the potential for configuration errors and avoids the security consequences of these errors.

Using INCS, the administrator makes the DHCP reservation for the IP address of the new user's computer on the "New User" screen, where the new IP address can be selected from a drop down list of available addresses. Without INCS, the administrator would need to access the Windows DHCP Manager, examine the addresses available for distribution, those excluded from distribution, and the reservations that have already been made in order to choose an appropriate IP address for the new reservation. The administrator would also need to look up and enter the computer's MAC address (this is needed for the reservation), while if INCS were used, the MAC address would already be stored by INCS. This illustrates that while it is easy to perform many tasks in INCS, there is some amount of initial setup required.

The third sub-task is turning on the user's switch port and enabling security features on the port. Without INCS, the administrator must identify the switch port, log into the network switch, and enter the configuration commands needed to configure the port. Using INCS, after the administrator associates the computer with a switch port in the INCS database, only one check box is used to apply all the default settings (of course, the default settings can be changed if the administrator has allowed them to be).

In summary, without INCS an administrator must enter three different systems to accomplish these tasks, and perhaps additional systems if information such as MAC addresses must be looked up in another database. By contrast, all of these tasks are accomplished within the INCS system, as the administrator enters a minimal amount of information and INCS in turn configures parameters in Active Directory, the DHCP Manager, and network devices. By providing for default settings, INCS also allows an administrator to limit the options available to entry-level staff so that they can safely perform standard operations, while advanced operations require the intervention of a more senior administrator.

IV. INCS System Testing

Test Configuration

Although the INCS system is still under development and testing has not begun, a rough outline of the test plan has been conceptualized and is presented in this section. The goal of the test plan is to confirm the proper functioning of INCS as well as to demonstrate the advantages of the system. The network configuration for testing is shown in Fig. below.

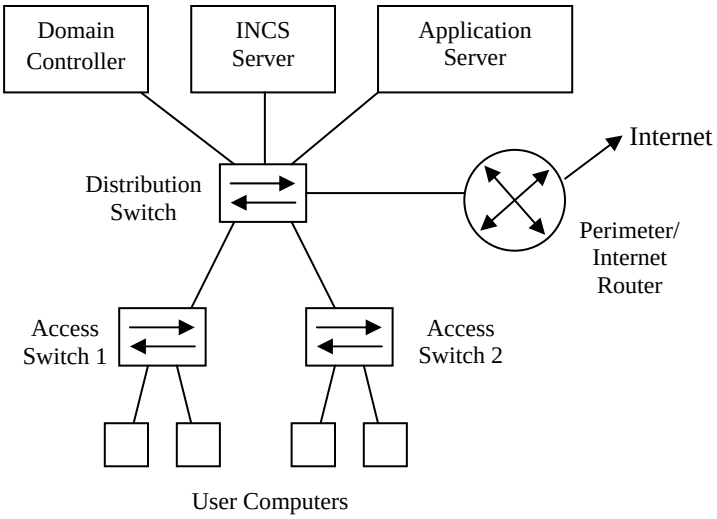

Fig. 2. Block diagram of the INCS test network

Two access layer switches are used to provide ports for user computers. These switches have a trunk link to the distribution layer switch. The distribution layer switch connects the server farm (represented by one Windows application server), the Windows domain controller, and the INCS system. The domain controller hosts the Active Directory database and the DHCP server, while the application server contains files and folders for testing the NTFS and share permissions aspects of INCS. The router provides simulated access to the Internet.

During the testing, 50 users and 50 computers will be entered into the INCS system, although only four physical computers will be employed. These four computers will migrate to various access switch ports as needed during the testing. Use cases will be designed for each of the functions of INCS. For example, adding a new user to INCS, assigning a machine to the user, and assigning a machine to a port will all be use cases. The testing will be completed when all of the use cases are successfully performed by the system.

INCS Efficiency

To illustrate the advantages that INCS provides, each use case will not only be performed using the INCS system, but also without the INCS system. The number of steps that the user must perform to accomplish the use case task will be recorded in both cases. If INCS is successful, the number of steps used to perform the tasks using INCS will be significantly fewer than the number of steps required without INCS. Much of the expected efficiency gain will be attributable to the fact that with INCS, functions involving multiple systems are performed through one user interface without needing to access multiple systems. Some of the gain will be due to the fact that the interface will concentrate on the most commonly performed network tasks, thereby avoiding much of the complexity inherent in a system whose interface is burdened with many rarely-used features.

V. CONCLUSION

The INCS system has much to offer as a network configuration system that focuses on commonly performed tasks. The three key advantages of the INCS system are that a) it presents a single user interface to accomplish common network tasks and access network information, thereby avoiding the need for the user to access different systems, b) it concentrates on the most commonly used network tasks and queries so that tasks are easily accomplished (especially when configurable default settings are utilized), and c) it is easily expandable to interoperate with any network devices or management systems that have an open interface or command set. In addition, the open source nature of INCS makes it much more cost-effective than multiple vendor-specific network management software packages.

REFERENCES

[1] Agrawal, S. Kanthi, C. N., Naidu, K. V. M., Ramamritham, J., Rastogi, R., Satkin, S., et al. (2007). Monitoring infrastructure for converged networks and services. *Bell Labs Technical Journal 12*(2), 63-78.

[2] Anderson, P. (1994). Towards a high level machine configuration system. *Proceedings of the Eighth Systems*

Administration Conference (LISA VIII) (p. 19). Berkely, CA: USENIX Association.

[3] Arnold, B. (1998). Accountworks: Users create accounts on SQL, Notes, NT, and UNIX. *Proceedings of the Twelfth Systems Administration Conference (LISA XII)* (p. 49). Berkely, CA: USENIX Association.

[4] Burgess, M. (1995). A site configuration engine. *Computing Systems, 8*(1), 309.

[5] Cisco Systems, Inc. (2007). *CiscoWorks Lan Management Solution*. Retrieved February 20, 2008, from http://www.cisco.com/en/US/products/sw/cscowork/ps2425/index.html

[6] Joshi, N., Riley, W., Schneider, J., Tan, Y. S. (2007). Integration of domain-specific IT processes and tools in IBM service management. *IBM Systems Journal 46*(3), 497-511.

[7] Libes, D. (1990). Using expect to automate system administration tasks. *Proceedings of the Fourth Large Installation System Administrator's Conference (LISA IV)* (p. 107). Berkely, CA: USENIX Association.

[8] Microsoft Corporation. (2008). *Systems Management Server Home*. Retrieved February 20, 2008, from http://www.microsoft.com/smserver/default.mspx