A Network Management Software Based on Secure Shell (SSH) Channels and Java Universal Network Graph (JUNG)

Students: Simran Khalsa, Gerald Castaneda, Ruben Rivera. Mentor: Dr. Jorge Crichigno
Department of Engineering, Northern New Mexico College
{simran_khalsa, gerald_c_castaneda, ruben_m_rivera, jcrichigno}@nnmc.edu

Abstract

This project presents a Network Management Software (NMS) implementation based on Secure Shell (SSH) channels and Java Universal Network Graph (JUNG). Using secure SSH channels, the NMS extracts the topology of any computer network using Open Shortest Path (OSPF) as the routing protocol. NMS consists of two subsystems: the Topology Visualization Module, and the Control Module. The first one provides a visual interface that permits dynamic interaction between the network manager and devices. The latter implements control and optimization algorithms for automatic control of the network. An example application of the Control Module is routing optimization, where the routing of traffic is dynamically adjusted to avoid congested areas or hot spots. NMS is able to handle real-time updates in the network, such as link and node failures.
Introduction

Computer networks consists of network devices and communications links. Network devices include specialized computers such as routers, which are the core of the Internet, and end users such as Laptops and mobile phones used by humans. Today, the Internet has grown into a production communication system that reaches all populated countries of the world and its use has grown exponentially [1]. A high-level overview of the Internet is shown in Figure 1 [2], where end users may be connected to the Internet by 3G providers, cable and DSL modems, and other services.

The advent and utility of networking has created dramatic economic shifts. To provide Information Technology (IT) services, any current middle and large-size Internet Service Provider (ISP) and enterprise must manage an important number of routers for proper operation. As a result, an entire industry, network management, has emerged to develop technologies, services, and products to facilitate the management and administration of networks.

Figure 1. Overview of the Internet architecture [2].
Efforts to improve network management in current and future networks includes the Software-Defined Networking (SDN) initiative [3]. SDN is an emerging architecture that separates the forwarding plane from the control plane in network devices. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

While the SDN architecture is still a work-in-progress, this project presents a Network Management Software (NMS) that proposes to control, in software, the behavior of network devices. NMS is based on Secure Shell (SSH) [4] channels and Java Universal Network Graph (JUNG) [5]. The project focuses on the management of routers using the secure SSH protocol which implements cryptographic algorithms to provide for authentication and confidentiality. Using secure SSH channels, the NMS extracts the topology of any computer networks that use Open Shortest Path (OSPF) [6] as the routing protocol. OSPF is the most widely used routing protocol.

NMS consists of two subsystems: the Topology Visualization Module, and the Control Module (Figure 2). The first one provides a visual interface that permits dynamic interaction between the network manager and devices. The latter implements control and optimization algorithms for automatic control of the network. An example application of the Control Module is routing optimization, where the routing of traffic is dynamically adjusted to avoid congested areas or hot spots. NMS is able to handle real-time updates in the network, such as link and node failures.
With NMS, routers can be manually or automatically operated according to the needs of the network manager. NMS also makes user access and troubleshooting more convenient, and will provide a feedback control system for traffic engineering [7] and network security.

**Methods**

Given a single IP node and SSH passwords, NMS probes the entire network for all existing layer 3 devices. The Topology Visualization Module (Figure 2) generates a visual representation of the network (Figure 3). By clicking on a given router, an SSH channel between NMS and the router is open for management purposes.

![Network Management Software System](image)

**Figure 2. Network Management Software System.**

Our future work includes the Control Module (Figure 2), which will permit managers to automate security policies and traffic engineering. The latter refers to the ability of routers to route traffic optimally using optimization algorithms such as Linear Programming and Dijkstra.

Java SE is the programming environment used to build this software. Java Universal Network Graph (JUNG) and Visual Library are used for drawing the network. The test-bed used
for developing and testing the NMS is composed of Cisco routers, family model 2900, used in the industry (Figure 4). Cisco routers 2900 implement SSH channels, OSPF, and several other protocols. The routers used at Northern New Mexico College (NNMC) have 2 serial synchronous interfaces and 2 Fast Ethernet interfaces. All software development is performed using the test-bed (i.e., real equipment) rather than simulation software.

![Network Topology Diagram](image)

**Figure 3. Topology Visualization output of a real network.**

**Further Discussion**

The NMS presented is built for network topologies using OSPF. In OSPF there are three type of networks: stub, point-to-point, and transit. A stub network is also known as Local Area Network (LAN). This is a network composed of end users that utilize a router to communicate with other LANs. A point-to-point network is a network in which two routers are serially connected. A transit network is a network in which two or more routers are connected through a LAN.
Figure 4. Test-bed used at NNMC for developing and testing NMS.

**Conclusion**

The first prototype of NMS has already been created and tested. The Topology Visualization Module is able to accurately build the topology of a real network. The module also permits the network manager to visually interact with any device. Current efforts include further testing of the Topology Visualization Module, integration of a Syslog [8] server for real-time updates, and the implementation of the Control Module. Future work includes the expansion of NMS to manage network devices in wireless networks such as wireless mesh [9] and sensor networks.

**References**


