Network Working Group
draft-ietf-pier-applications-00.txt
Internet Draft
Expire in six months

Philip J. Nessser II
Nesser & Nesser Consulting

March 1996

IP Addresses in Applications

Status of this Memo

1.0 Abstract

The Procedures for Internet/Enterprise Renumbering (PIER) Working Group of the Internet Engineering Task Force (IETF) has been tasked with the creation of documents to aid renumbering efforts. This document defines a series of classes of IP address locations. Each class will be described in a general sense, while specific examples are provided as possible.

2.0 Introduction

In an effort to aid organizations in the efforts of renumbering the PIER group is producing a series of informational documents about renumbering. While much press has been given to the special case of required renumbering when changing Internet Service Providers (ISPs), there are many reasons which necessitate renumbering. For a very detailed discussion of the reasons for renumbering the reader is refered to RFC XXXX. A few of those reasons are given below:

- o Changing ISP's;
- o Company Splitting into smaller subdivisions;
- o Two companies merging;
- o Moving facilities whose physical layout require

  topological changes;
- Changes in network topology;
- Moving from a bridged to a routed network;

When developing a renumbering plan the administrator typically identifies the individual elements on the network and tries to group those devices into like groups. A relatively natural set of groupings is presented below:

- Routers;
- Infrastructure Devices: Bridges, Terminal Servers, Gateways, Firewalls;
- Applications Servers: DNS, Mailhubs, News Servers, FTP Servers, WWW Servers, Network Management Systems;
- End User Systems.

A complete guide to router renumbering has been published as RFC XXXX, and hence will not be covered in this document. This documents will attempt to cover all of the remaining devices. There will not be a linear mapping between the groups above and the the classes presented below. For example, when renumbering a WWW server, it will be necessary to consider both the underlying system and the WWW server software.

3.0 Basics

When approaching a renumbering project there are several preliminary steps which should be addressed before proceding with the project. Some of the steps may seem unnecessary or overcautious but in the end they almost always save substantial time and wasted effort.

3.1 Identify the Scope of the Renumbering Project.

It is not sufficient to decide to renumber a series of networks. Each network, each device on that network, and each network to which it connects must be inventoried and designated, since there is interaction between all three groups. The last group is especially important in cases of firewalls, access control lists, etc.

3.2 Identify the Numeric Boundries of the Renumbering Plan.

It is important to know the ranges of IP addresses which are in use and will be in use after the renumbering plan is complete. The simplest case is renumbering one block of addresses to another block of addresses. A more complex case might involve renumbering a block of addresses into the same block of addresses with a different network toplogy. In this case, the order of implementation is critical to a sucessful project.

3.3 Identify the New Network Topology.

While many cases of renumbering are simply a change of prefix, many involve a network topology change as well. For example, moving to a different building may require a shift in both addresses and topology. In the case of a changing topology, the new plan should be in place before any changes are made.

3.4 Identify the Components.

Once the above steps are completed the basic inventory of devices identified above can be fleshed out. Each operating system and hardware platform has distinctions which may aid or hinder its ability to renumber gracefully. This RFC attempts to identify those strenghts and weaknesses. For example, some routers and host operating systems allow multiple IP addresses to be assigned to the same interface. This is a great boon when renumbering since the old and the new addresses can both be functioning at the same time.

3.5 Create a Map.

It will be greatly beneficial to create a IP address map between the old addresses and the new addresses. This will aid in the migration in countless ways, as well as serve as a bridge backwards to any need to translate historical data to the current topology.

3.6 Make the Most of the Pain.

There is no doubt that renumbering IP networks can be a painful process for all concerned. Since you have to endure the pain, it is clear that it would be desireable to make the most of the situation. Try and take this opportunity to put as much planning and resources into the project as possible. Updating operating systems, hardware firmware and BIOS'es, new cabling, new equipement, new address assignment plans, etc. are all items which many be necessary when renumbering, so place careful emphasis on designing a new system which is flexible to change and well designed.

4.0 Classes

Each of the following sections will be devoted to one "Class" of renumbering problem. These will roughly correspond to a set of locations that contain IP addresses, and some basic ideas on how to sucessfully renumber those addresses. Wherever possible a set of very specific examples will be given. Every effort has been made to provide as many examples as possible, but they are definitely not exhaustive.

4.1 Firewalls (Including Filtering Routers, Proxy Servers, Application Layer Gateways (ALGs), and Network

Address Translators (NATs)

To Be Done

4.2 Network Management Stations (NMS)

NMS's are typically dedicated machines which runs a SNMP application used to both monitor systems, routers, and other infrastructure, but also to collect, store, manage and track that data. In all known cases, NMS's store this data based on the IP address of the monitored device or interface. Because of the typical volume of this data, it is typically stored in a proprietary database format to save space, and speed access.

To Be Finished

4.3 Software License Servers

To Be Done

4.4 Names Systems (Including Domain Name System (DNS), Windows Internet Naming Service (WINS), Network Information Services (NIS) and NIS Plus)

4.5 DHCP/BOOTP Servers

To Be Done

4.6 Client Configurations (Unix varieties, Windows 95, Windows NT, Mac OS 7.5)

The purpose of all of the infrastructure is to allow operation between host/client computers. In this context all machines are clients, in the sense that all systems underlying operating systems (OS's) need to be renumbered, even if additional steps might be necessary to renumber higher layer applications.

Part of the difficulty in this problem is the extreme success of the TCP/IP protocol. The ability of IP to run over such a large variety of level two protocol media has encouraged adoption on so many hardware platforms and operating systems. The combinations of hardware, operating system, operating system version, network software implementation, network software implementation version, layer two protocol and layer two media provide a vast, if not unlimited number of possibilities. This discussion will center on the likely locations of hard coded IP addresses in the OS. Some specific examples will be provided for some of the most common hardware platforms and OS's.

Operating systems typically have IP addresses in a minimum of three locations. First make the assumption that this machine does not use one of the dynamic address protocols (bootp, dhcp, etc). The first is the location of the IP address for machine itself. The second location is the default gateway for the network to which the machine is connected. The third location is an IP address for a machine which provides name to IP address mapping (there of course are often multiple name to IP address servers for redundency). Many popular implementations do group all three of these IP addresses in a single location, making changes straightforward. This is perhaps the simplest configuration for a machine. In many sites a large majority of machines to be renumbered will fall into this category. The observation that 90% of the machines to be renumbered will only take 10% of the time and effort, while the last 10% of the machines will take over 90% of the time and effort. These simple client machines will typically be the bulk of the *number* of machines to renumber but are typically the easiest.

There are, of course, other locations in operating systems where IP addresses can be found. These additional locations can be grouped into two categories. The first is a local cache of hostnames mapped to IP addresses, while the second group is usually dependent on additional applications running on the system. The use of IP addresses for both of these purposes followed the philosophy that addressed changed so rarely that it was more network efficient to hard code addresses for local hosts that were accessed often, as well as, addresses of machines that provided well know services, thus saving many unneccesary name server lookups. This is not the case in the Internet of the 1990's, and should therefore no longer be practiced.

In the first category, for example, the local time sharing machines, or mail hub, or news server, etc. may be stored locally to avoid "wasted" nameserver lookups. In the second category, for example, there are several time syncronization protocols which allow clients to sync their time and date with a server somewhere on the network. There is typically a startup file which identifies the time server's address. This is perfect example of a situation where an IP address is not needed, and where a hostname would be better suited.

In the past, it was common for IP addresses to be used in situations like this where remote servers rarely changed addresses, thus it was unnecessary to "waste" namesever lookups in these cases. In the "new" Internet with its quicky growing infrastructure it is an unwise decision to hardcode such addresses. In fact, as "smarter" application servers are deployed which learn network topology and provide the location to the "best" server for an application, there may be significant problems if hard coded addresses are used.

Other examples of possible IP address locations which fall into the second category include:

- Configuration of remote time syncronization (as above)
- Configuration of remote printers
- Configuration of remote filesystem connections
- Non-default network to subnet mask mapping

o Configuration of remote font servers

4.6.1 Examples -- Unix --

4.7 Applications (Web Servers)

To Be Done

4.8 Mail Systems

To Be Done

4.9 Netbios over TCP

To Be Done

4.10 System Security Tools (TCP Wrappers, Socks, Xinetd)

To Be Done

4.11 Documentation (Online and Offline)

Every system has some sort of online documentation. At the simplist level it may be self referential documentation (i.e. hosts file), or it may be elaborite documentation covering each network node and its associated network interfaces and services, or it may fall somewhere in the middle. Nevertheless, it is vital to update this information as part of the renumbering process. In some cases, where the documents are either in ASCII or stored in a database, it will be relatively easy to automatically convert the information using a mapping table from old to new addresses.

However, there are many places where IP addresses are embedded in a binary document, say a word processor or spreadsheet file, where significant manual intervention may be required.

An often over looked location when renumbering is your organizations off-line documentation. Over the years most organziations have developed numerous offline documents which could contain IP numbers. This may be a good time to do a complete scan of all offline documentation. Below is a list of examples which should be checked for hardcoded IP addresses.

- System setup information
- Disaster recovery plans
- End user documentation
- Network maps
- Dialin instructions
- Numbering schemes

- List of network resources (DNS servers, gateways, Database servers, etc.)

5.0 Security Considerations

Security issues are not discussed in this memo.

6.0 Authors' Addresses

Philip J. Nesser II
Nesser & Nesser Consulting
13501 100th Ave NE, Suite 5202
Kirkland, WA 98034
USA

Phone: (206)481-4303
Email: pjnesser@martigny.ai.mit.edu

7.0 References