Addressing

In order for packets (the physical data units traveling across the media, which correspond to the frames created by the data link layer) to find their way across a network, the source and destinations need to be identified in some way. We refer to this identifying name as an address. Different layers of the protocol stack require different addressing mechanisms, as determined by the task to be accomplished at that layer.

Address Types

MAC Addresses
MAC (Media Access Layer) Addresses are also known as ‘physical’ or ‘hardware’ addresses. These addresses uniquely identify the devices to which physical media is directly connected. Most commonly these devices are Network Interface Cards (NICs) designed according to the Ethernet standard. The MAC Addresses are hard-coded into each card. A host might have several such cards installed, and each of them will have a unique MAC Address.

MAC Addresses are 48-bit binary numbers, specified for human use as six pairs of hexadecimal digits separated by hyphens. For instance, A3-BF-12-55-DD-E4 might be the MAC Address for some device. MAC Addresses are assigned by the IEEE; the first (high-order) 24 bits (3 bytes or 6 hex digits) identify the manufacturer of the device. The manufacturer then assigns the remaining portion of the address to each of the devices they manufacturer. For each high-order identifier a manufacturer has $2^{24} = 16$ Million$^1$ individual addresses it can distribute.

MAC Addresses are used by the Data Link Layer (Layer 2) of the OSI Model and are frequently referred to as ‘layer 2 addresses.’

Host Names
Host Addressing uniquely identifies a specific host on a network or on the Internet. It may be hierarchical or non-hierarchical.

Non-hierarchical addresses are used on LANs which do not connect to the Internet. As discussed earlier, NetBIOS names are an important non-hierarchical naming mechanism. They consist of 15 characters followed by a 1-byte control code. The hostname assigned to a computer when networking is

---

$^1$As usual, a million = $2^{20}$, not $10^6$. 

nos_text_addressing.wpd

NTC 12/20/04
 installed is generally also the NetBIOS name.

**Domain Names**

Hierarchical Addresses allow hosts to be found by an iterative search through a hierarchy, based on the format of the host name. We are most familiar with World Wide Web domain names, which generally have at least three parts separated by periods, such as


In this name, org is the **top-level domain (TLD) name**. There are a relatively small number of top-level domain names, the most well-known of which include

- com commercial or business domains
- edu educational institutions
- gov governmental institutions
- mil military institutions
- net network providers, such as ISPs (Internet Service Providers)
- org non-profit organizations

There are a few others not mentioned here. In addition there has been some progress in defining some brand-new top level domains such as .name and .biz.

In addition to the TLDs listed above, there is a 2-character TLD assigned to every country in the world. Thus, the TLD for Japan is jp, for the United Kingdom it is uk, etc. For instance, the domain name for the BBC is bbc.co.uk.

**Second-level domain names** are the names immediately preceding the TLD. In our example, Tatooine would be the second-level domain name. These names are unique to an organization and we usually just refer to the whole construct ‘Tatooine.org’ as the organization’s domain name. This is the name that you can purchase (register) and have exclusive use of for your web site, etc.

The first part of the name is the **hostname** of the name of the host machine which is being referenced. This is also the NetBIOS name on a windows LAN. If this machine is running a server process which is being accessed, than this part of the name, Luke in our example, is also known as the **server**. In the early days of the internet most web servers were called www.

Once an organization has registered its domain name it can create additional hierarchies within its organization, called **subdomains**. Our Tatooine.org
domain might elect to create a subdomain called Skywalker, for instance. The resulting domain name in our example would be


**Host Addresses (IP Addresses)**

Host addresses, usually called **IP Addresses** and sometimes Logical Addresses are binary strings which uniquely identify a host machine on the Internet. Every device on the Internet which may be the destination of a packet needs an IP Address.

IP Addresses (IP Version 4, or IPv4) consist of 32 bits, broken up into four bytes or octets. Again, for human consumption, the convention is to represent these binary numbers as four decimal integers separated by periods. For example, 192.168.10.1 might be a valid IP Address. Note that since each decimal number is actually the decimal equivalent of an 8-bit binary number, they can only take on the values 0 through 255. Thus, 192.268.10.1 is not a valid IP Address since the second number is greater than 255. We are rapidly running out of available IP Addresses. To rectify this situation a new version, IPv6, is in the process of being deployed. IPv6 addresses have 128 bits, providing enough addresses to assign one to every particle in the universe.

IP Addresses are controlled and distributed by the Internet Corporation for Assigned Names and Numbers (ICANN). Numerous companies have been authorized to act as registrars for ICANN and IP Addresses can be purchased through these registrars. How these Addresses are assigned and used will be discussed later.

Every Domain Name has a corresponding IP Address. The master registry of all domain names and IP Addresses is maintained by **Network Solutions, Inc**.

IP Addresses are used by the Network Layer (Layer 3) of the OSI Model and are frequently referred to as ‘layer 3 addresses.’

**Port Addresses**

Port Addresses (or Port Numbers) are integers assigned to individual applications, or processes, running on a host. When a packet is routed through the Internet (by the Internet Protocol) it is delivered to the designated destination, but it must now be routed to the appropriate process running on that host. Each such process is assigned a unique (to that host) number for identification purposes, called a port address. The range of allowed port numbers is from 0
through $2^{16} - 1$ (65535).

When a client process, such as a web browser, makes a request it is assigned a port number by TCP so that when the response (the web page) comes back it can be forwarded to that browser and not, say, an e-mail client or another browser. Such client port numbers are assigned randomly by TCP.

On the other hand, when a web browser makes a request, it has to specify that web server software on the destination host is the intended recipient. In this case it is clearly not helpful for the web server to have a random port address. In fact there are a set of well-known port addresses reserved for use by such servers as web servers, ftp servers, mail servers, etc. The reserved ports are 0 through 1023 ($2^{10} - 1$); port numbers above 1023 are generally used for random assignment to client processes. Some well-known port addresses are 80 (web servers) 25 (mail servers) and 21 (ftp servers)

Port Addresses are used by the Transport Layer (Layer 4) of the OSI Model and may be referred to as ‘layer 2 addresses’, although this is not common.

**Sockets**

A socket identifies a specific interface on a specific host. It consists simply of the host’s IP Address catenated with the processes port address, separated by a colon. For example, 192.168.10.103:2055 might be a valid socket interface.

**URLs**

URL is an acronym for Universal Resource Locator. It is the addressed passed to a layer 7 (Application layer) process by the user whenever a resource (such as a web page, or a file being accessed remotely by, say, FTP) is requested from a remote server. The generic form of a URL is given by

```
server process://domain name:port/path
```

The **server process** is the name of the protocol running on the server machine which is providing the service. We are most familiar with **HTTP** which indicates a web server, but others are often used including **FTP** for remote file transfers, and **NNTP** for newsgroup servers, among others.

The **domain name** is just what we have been discussing above. It includes the hostname, any subdomains, and the second and top-level domain names.

The **port** is usually omitted because any server process is usually listening on
the appropriate well-known ports.

<table>
<thead>
<tr>
<th>Process</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>80</td>
</tr>
<tr>
<td>FTP</td>
<td>21, 20</td>
</tr>
<tr>
<td>SMTP</td>
<td>25</td>
</tr>
<tr>
<td>NNTP</td>
<td>119</td>
</tr>
<tr>
<td>Telnet</td>
<td>23</td>
</tr>
</tbody>
</table>

However, there are occasions when a server is assigned to listen on a port which is not well-known. For instance, a web server may be set up for the use of a restricted group of users. It would be configured with a non-well-known port address and only those users who are permitted access would know what that port address is. They are then required to explicitly include it in the URL.

The path may be a simply a file name or a more complex path specification (see the chapter on File Systems) to the desired file. It is also optional in an URL. The domain name on a particular host actually specifies a directory on that host which has been identified as the root document directory which is the top of the subtree on the host which contains the pages which may be accessed by the server for delivery to requesting hosts. Usually this root document directory contains a default page (such as index.html) which the server sends to a requestor if no page or path is specified. However, if the name of a particular file is known it may be specified in the URL. If the desired file is not in the root document directory, but somewhere lower down in the subtree, then a full path needs to be specified.

The following is an example of an URL containing all of these elements.

http://harry.Gryffindor.hogwarts.edu:3055/owls/mail.htm

Can you identify the server process, server host, subdomain, domain name, port, path and document being requested?