A (very) brief introduction to TCP/IP

By R Hart

Overview

The aim of this short article is to explain how a machine on a network uses TCP/IP to communicate with other hosts. It is not meant to be an in-depth discussion, as there are many articles to be found online which delve deeper into the topic.

Introduction

Internet Protocol version 4 (or IPv4) has been around for over 30 years. I believe that it was first defined in RFC 791. RFC's (or Request for Comments) are documents published by the IETF (Internet Engineering Task Force) containing notes and technical descriptions of the various Internet protocols and technologies. At the time of writing this guide, there are over 7000 RFC's!

In order to communicate with other hosts on a network, IPv4 enabled machines require three things:

1. IP Address
2. Subnet Mask
3. Default Gateway

We will discuss each of these in a little more detail below.

IP Address

An IPv4 IP Address is a 32 bit number commonly written in what is known as the dotted notation, where each group of numbers consists of 1 byte (or 8 bits) and is then separated by a dot. Although IP addresses can be written in other notations, the dotted notation is a lot more convenient than trying to write it in binary or hex.

IP addresses should be unique for the network which the machine is attached to. If not, conflicts will occur and this can cause problems.

Subnet Mask

A Subnet Mask is used to distinguish the network part of the IP Address, from the host part of the IP Address. This is achieved by creating classes of networks, of which there are five. These networks can be further subdivided by the use of variable-length subnet masks (VLSM) but that's a discussion for another time.

Default Gateway

The Default Gateway is a host on the network which routes traffic from machines on one network to machines on other networks. If a machine sending a packet of data doesn't have a route to the destination address in its routing table, it forwards the packet to the IP Address of the Default Gateway. The Default Gateway will then route the packet accordingly.

Method of Operation

Let us assume for this example that there are four hosts (or nodes) configured with IPv4 addresses as follows:

- HRMACHINE – 192.168.0.100
- FILESERVER – 192.168.0.10
- PROCSERVER (on MS Azure platform) - 94.245.117.129
- ROUTER1 (default gateway) – 192.168.0.254

If HRMACHINE wants to communicate with FILESERVER, HRMACHINE will use the Address Resolution Protocol (ARP). ARP is used to convert an IP Address to an Ethernet Address (also called a MAC, or Media Access Control address).

Before an ARP request is sent out, HRMACHINE will check its local ARP cache to see if a record exists, and if not, a request will be sent.

The ARP request is broadcast to the IP Address of 255.255.255.255 which all nodes on the network will respond to. The ARP request basically says 'Who has got IP address x.x.x.x? Send me the MAC address of the node'.

In this example, FILESERVER will respond to HRMACHINE with its MAC address and IPv4 address. The response will then be cached in the ARP table on HRMACHINE and the message can be sent.

What happens though if HRMACHINE wants to communicate with PROCSERVER which is running on Microsoft Azure?

In this case, HRMACHINE will check its routing table and see that there is no route for the 94.245.117.129 address, so it will forward the packet of data to ROUTER1 which is its default gateway.

ROUTER1 will then check its routing table. If it finds a match for the destination address, it will forward the packet out of the appropriate network interface, but if not, the router will forward the packet out of its own default gateway.

Conclusion

If this short guide has piqued your interest in IP communication, I would strongly suggest seeking out further information online. There are many resources available which provide an in-depth look into the world of IP networking, many of which are easily found with a Bing search.

If you would like to actually see the packets of information as they are going from one network node to another, a protocol analyser like WireShark is invaluable.

Thank you for taking the time to read this article, I hope you found the content of interest.