**Introduction**

Dynamic Host Configuration Protocol, or DHCP, is a very widely used network protocol. DHCP was first mentioned in RFC 1531, which was released in October 1993. This version had many flaws, which led to an updated RFC 2131 in March 1997. RFC 2131 is still the standard for IPv4 networks today. On a basic level, DHCP allows devices to request Internet Protocol parameters from a network server. One of the main parameters DHCP provides is automatically allocating IP addresses to these devices when they are requested.

**Background**

DHCP uses UDP to transport messages. The main reason for this addressing issues that come along with the protocols purpose. Most of the messages that are exchanged when using DHCP are broadcast throughout the local network, so there is no specific destination until the end of the transmission. With this, it’s unreasonable to attempt to set up a reliable TCP connection between the client and host. DHCP has the client and servers communicate on UDP port 67 for the server destination port and port 68 for the client destination port (Kozierok, 2005).

**Address Allocation**

DHCP has three methods of allocation for IP addresses. The most popular of these is dynamic allocation. With dynamic allocation, DHCP will assign the IP address to the client for a finite time period. This lease time is able to be set and changed by the network administrator. Once the lease time is over, or if the client releases the IP address, the IP address is reclaimed by the network and can be given to a new client. The other two methods do not allow the network to reuse an IP address. In automatic allocation, an IP address is permanently assigned to the client. With this, if a client leaves the network and returns, they will retain the same IP address. In the time that the client is off the network, the IP address is unable to be used. Manual allocation works similarly to this, except the IP address is manually assigned by the network administrator. DHCP has no responsibilities except to pass this preset network address to the client when it is requested. There are certain situations where manual or automatic allocation would be preferred, but in most situations, dynamic is the mechanism of choice.

The allocation of IP addresses is done with a scripted client-server interaction. First, the client will broadcast a ‘DHCPDISCOVER’ message to its local subnet with option parameters of network address and lease time preferences. The server will respond with a ‘DHCPOFFER’ message with an available network address. The server doesn’t necessarily have to reserve the address that was sent in this message, but some still choose to do so since it increases the protocol’s efficiency. When the client receives this message from the server, it will choose which network address offered that it will like to accept. The client will send out a ‘DHCPREQUEST’ message that has the server who’s offer has been accepted as well as the network address that the client would like to accept. When the server receives the ‘DHCPREQUEST’ message, it will respond with a ‘DHCPACK’ message that has additional parameters for the client’s network configuration. If the server was unable to handle the ‘DHCPREQUEST’ message from the client, whether because the network address is no longer available or for other reasons, it will reply with a ‘DHCPNAK’ message instead of the ‘DHCPACK’. The ‘DHCPNAK’ will tell the client to restart the process. If the client receives the ‘DHCPACK’, it is now fully set up with this allocated IP address. If the client wants to release the IP address as mentioned with dynamic allocation, it will send a ‘DHCPRELEASE’ message to the server. A client is able to request a previously owned IP address by dropping some of the initialization messages sent in the beginning (Droms, 1997).

**Security Issues**

Since DHCP uses UDP and Internet Protocol, it is expected that it would inherit some of the security flaws. An unauthorized DHCP server can be easily set up. This allows for malicious activity including sending false and/or excessive information to a client. It may also sound routing information that isn’t correct to a client. This is known as spoofing. On the other end, a client could pose as another client and intercept the data that was intended for another client. Doing this, a malicious client could become a resource hog by retrieving all of the resources available. This causes legitimate clients to have no resources available for use (Droms, 1997).

…….Questions he said he would like answered that I haven’t addressed yet

-Why does it exist?

-What RFC’s define it ---I mentioned 2 in the beginning but I didn’t go into any depth

- Any other points of interest?

- Definitely need a conclusion too