

## Clase PaqueteDHCP

Esta clase es responsable de llenar la trama del protocolo con todos los debidos campos para ser insertado en un objeto de tipo DatagramPacket o procesar los campos recibidos en un DatagramPacket y armar el paquete dhcp y ser procesado por el servidor.

El paquete dhcp se compone de los siguientes campos definidos por el RFC 2131:

**Op:** Define el tipo de mensaje 1=bootrequest(petición del cliente), 2= bootreply (respuesta del servidor dhcp)

**Htype:** tipo de hardware usado por el cliente. En este caso se llena el campo con el mismo dato que llega en el discover para responderle al cliente.

**Hlen :** Longitud de la direccion hardware. Tambien conocida como la dirección MAC

**Hops:** se pone en 0. Campo usado cuando se tiene intermediaries entre el servidor dhcp y el cliente

**Xid:** Identificador único de la transacción realizada. Es id es elegido aleatoriamente por el cliente y usado hasta que la comunicación entre el servidor y el cliente culmine.

**Secs:** tiempo de transcurrido desde la petición de una dirección o renovación de una ya asignada.

**Flags:** Este campo está en 0 pues el servidor soporta broadcast.

**CIaddr:** Dirección ip del cliente. Campo únicamente usado si al cliente ya le fue asignada una ip con anterioridad.

**Yiaddr:** La dirección ip que le ofrecerá el servidor al dhcp.

**Giaddr:** Dirección de pasarela del intermediario (si hay alguno).

**Chaddr:** Dirección MAC del cliente

**Options:** Opciones adicionales para la asignación ip.

## Funcionamiento de la clase

Un paquete dhcp es un arreglo de octetos o bytes y la aplicación lo tratará como tal en el transcurso de la aplicación. Esto facilita el proceso pues la posición en el arreglo de cada campo del protocolo está definido en rfc 2131.

El proceso básico para leer un mensaje de un cliente a través de un un objeto PaqueteDHCP es el siguiente:

1. Se llama al constructor PaqueteDHCP

```
public PaqueteDHCP(byte []data){  
    this.data=data;  
}
```

El parámetro data es un arreglo de bytes cuyo tamaño es definido de acuerdo al mensaje proveniente del cliente. Puede ser de la siguiente manera:

```
PaqueteDHCP pack = new PaqueteDHCP(messageIn.getData());
```

2. Se acceden a los campos que se quieran procesar:

```
public byte[] getCHADDR(){  
    return deAaBdeDATA(28,34);  
}
```

En este caso se quiere acceder a la dirección MAC del cliente. Esto pasa cuando el servidor recibe un DISCOVERY. Para lograr esto necesitamos una sola porción del mensaje (del byte 28 al 34-1). Esto nos retornara un arreglo de bytes usando el método deAaBdeDATA(int inicio,int fin):

```
private byte[] deAaBdeDATA(int A, int B){  
    byte[] b = new byte[B-A];  
    int j=0;  
    for (int i=A;i<B;i++)  
    {b[j] = data[i];  
    j++;  
    }  
    return b;  
}
```

Para crear un mensaje de tipo DHCP se siguen los siguientes pasos:

1. Se crea un objeto de tipo PaqueteDHCP usando el constructor por defecto

```
public PaqueteDHCP() {}
```

2. Se van estableciendo los valores de los datos y se van a agregando al arreglo de bits data en el lugar indicado por el RFC. Por ejemplo

```
public void setCHADDR(byte[] s){  
    data = this.concatDataDeAaB(28, 34, s);  
}
```

Los métodos set de la clase PaqueteDHCP insertan in arreglo de bits en los intervalos determinados por parámetro del método concatDataDeAaB(int inicio,int, fin,byte []s). El intervalo es definido por cada setter de acuerdo con el RFC 2131.

```
private byte[] concatDataDeAaB(int A, int B, byte[] b){  
    byte[] d = new byte[B];  
    int j=0;  
    for (int i=0;i<B;i++)  
    {if(i<A){  
        if (i >= data.length) {d[i]=(byte) 0x00;} else {d[i]=data[i];}  
        else{d[i] = b[j];j++;}  
    }  
    return d;  
}
```