Compilando Conocimiento

Redes Computacionales

Ciencias de la Computación

Oscar Andrés Rosas Hernandez

Febrero 2018

Índice general

Parte I

Aparatos Físicos

Capítulo 1

Hub

Capítulo 2

Switch

Capítulo 3

Routers

Capítulo 4

Access Points: Puntos de Acceso

Parte II

Network Layer: Capa de Red

Capítulo 5

Protocolo IP

5.1. Introducción

Debido a la cantidad de cables necesarios para conectar cada red con cada otra red del mundo no todas las redes tienen una conexión directa, es decir, no existe un cable entre tu red lócal y los servidores de Facebook por ejemplo.

Por eso existe el Protocolo IP que nos permite comunicarnos entre redes.

En resumén lo que permite es que tu red local solo este conectada a unas pocas redes y a varios routers, estos tienen algo llamado una tabla de direcciones, que les permite navegar entre redes hasta encontrar su destino.

El enrutamiento es parecido a la recursión, en el sentido en que no soluciona tu problema sino que solo te lleva un paso más cerca.

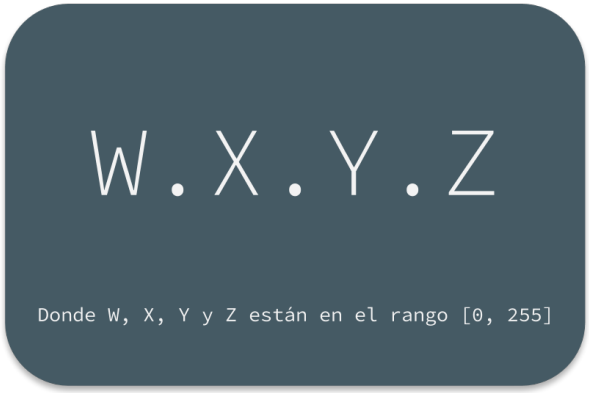
5.2. Direcciones IP

Es un identificador único (o casi, ya verás después porque). Necesitamos un identificador único porque es lo que nos permite enviar información y que la información que esperamos de regreso sepa a donde llegar.

5.2.1. Dirección IPv4

Como fue originalmente desarrollado este esquema podría alocar un identificador de **32 bits** a cada dispositivo que se quisiera conectar a internet. Esto nos daría algo así como 4 mill millones de posibles direcciones IP.

La convención es que estos serían representados como 4 conjuntos de 8 bits representados en decimal (una forma un poquito más amigable al público general), es decir:



Por ejemplo una IP v4 válida podría ser 140.247.220.12.

Problemas con IPv4

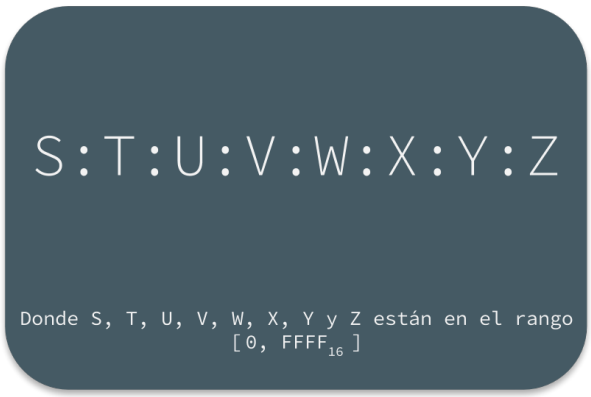
Ahora, recuerda que te dige que IP v4 acepta unos 4 mil millones de direcciones válidas, ahora el problema es que ahora mismo hay vivos mas de 7 mil millones de personas (A principios del siglo XXI) cada una con seguramente más de un dispositivo que quieran conectar a internet.

Por lo tanto tenemos que encontrar una forma de solucionar esto.

5.2.2. Dirección IPv6

Como vimos antes, ahora que parece que la cantidad de direcciones IPv4 se nos esta quedando corta, poco a poco estamos pasando de IPv4 a IPv6 que contará con nada menos y nada mas que **128 bits** para una dirección, es decir nos permitirá tener unas más o menos: $340,282,366,920,938,463,463,374,607,431,768,211,456$ posibles direcciones IP. Un chingo.

La convención es que estos serían representados como 8 conjuntos de 65535 bits representados en hexadecimal (porque de otra manera sale un númerote), es decir:



Por ejemplo una IPv6 podría ser $2001:0DB8:0000:0042:0000:8A2E:0370:7334$

Haciendo un poco más faciles las Direcciones IPv6

Ahora, todo esta mucho mejor que con IPv4, pero tenemos un pequeño problema, sus direcciones son moustrosamente enormes, por lo que tuvimos que hacer algunas simplificaciones para los humanos:

- Ignora los ceros dentro de cada grupo de 4 dígitos hexadecimales:

  Ejemplo:

  De $2001:0DB8:0000:0042:0000:8A2E:0370:7334$
  a $2001:0DB8:0:42:0:8A2E:370:7334$

- Si tienes un montón de ceros pon :: y da por sentado que quien lee esta dirección tiene cerebro y puede entender que ahí van ceros:

  Ejemplo:

  De $2001:0DB8:0000:0042:0000:0000:0000:0000$
  a $2001:0DB8:0000:0042::$

5.3. CheckSum

```java
public class Checksum {

  /**
   * Calculate the Internet Checksum of a buffer (RFC 1071 - http://www.faqs.org/rfcs/rfc1071.html)
   * Algorithm is
   * 1) apply a 16-bit 1's complement sum over all octets (adjacent 8-bit pairs [A,B], final odd length is [A,0])
   * 2) apply 1's complement to this final sum
   *
   * Notes:
   * 1's complement is bitwise NOT of positive value.
   * Ensure that any carry bits are added back to avoid off-by-one errors
   *
   *
   * @param buf The message
   * @return The checksum
   */
  public static long calculateChecksum(byte[] buf) {
    int length = buf.length;
    int i = 0;

    long sum = 0;
    long data;

    // Handle all pairs
    while (length > 1) {
      // Corrected to include @Andy's edits and various comments on Stack Overflow
      data = (((buf[i] << 8) & 0xFF00) | ((buf[i + 1]) & 0xFF));
      sum += data;
      // 1's complement carry bit correction in 16-bits (detecting sign extension)
      if ((sum & 0xFFFF0000) > 0) {
        sum = sum & 0xFFFF;
        sum += 1;
      }

      i += 2;
      length -= 2;
    }

    // Handle remaining byte in odd length buffers
    if (length > 0) {
      // Corrected to include @Andy's edits and various comments on Stack Overflow
      sum += (buf[i] << 8 & 0xFF00);
      // 1's complement carry bit correction in 16-bits (detecting sign extension)
      if ((sum & 0xFFFF0000) > 0) {
        sum = sum & 0xFFFF;
        sum += 1;
      }
    }

    // Final 1's complement value correction to 16-bits
    sum = ~sum;
    sum = sum & 0xFFFF;
    return sum;

  }

    public static void main(String[] args){
        byte[] buf = {
                        (byte) 0x45,
                        (byte) 0x00,
                        (byte) 0x00,
                        (byte) 0x3c,
                        (byte) 0x0a,
                        (byte) 0x1c,
                        (byte) 0x40,
                        (byte) 0x00,
                        (byte) 0xff,
                        (byte) 0x06,
                        (byte) 0x00,
                        (byte) 0x00,
                        (byte) 0xa8,
                        (byte) 0xb0,
                        (byte) 0x03,
                        (byte) 0x19,
                        (byte) 0xa8,
                        (byte) 0xb0,
                        (byte) 0x03,
                        (byte) 0x6c
                    };

        long resultado = Checksum.calculateChecksum(buf);
        System.out.printf("Valos del check: %02X\n",resultado);
  }//main
```

```
84
85 }
```

Parte III

Application Layer: Capa de Aplicaciones

Capítulo 6

DHCP

Espera, espera ... ¿Cómo obtengo mi dirección IP?

6.1. Introducción

Este es un protocolo que nos permite dar una dirección IP a cada dispositivo. Esta dado por *Dynamic Host Configuration Protocol*

Después de todo, si no existiera este protocolo, ¿Cómo obtengoo una dirección IP? No puedo tomar la que quiera, porque sino puede que haya más de una persona que piense como yo y eliga mi misma dirección. Y si sabes lo que es que alguien en tu calle tenga el mismo número que tu, sabes que ahí tienes un problema.

Capítulo 7

DNS

Espera, espera ... ¿y cuál es la IP de Cats.com?

7.1. Introducción

Este es una aplicación que nos permite hace una traducción de direcciones IP a nombres un poco más amigables para los seres humanos. Esta dado por *Domain Name System*. Es básicamente como las páginas amarillas del internet