Cuestionario

Redes IP

Redes clase A: dedicadas a grandes redes, dirección global de 32 bits, los primeros 8 bits del ip corresponden a la red y el resto al host. Comienzan con 0, y van desde 0 a 127 (00000000 hasta 01111111). 2^7=128 redes. Su máscara natural es 255.0.0.0

Redes clase B: dedicadas a redes medianas, comienzan con 10 en binario, van desde 128 hasta 191 (10000000 hasta 10111111). 2^14=16384 redes. Su mascara natural es 255.255.0.0.

Redes clase C: dedicadas pequeñas y privadas. Comienzan con 110 en binario y van desde 192 hasta 223 (11000000 hasta 11011111). 2^21=2097152 redes. Su mascara natural es 255.255.255.0.

b) Las direcciones de Internet privadas son:



c) ***Broadcast***, es una forma de transmisión de [información](https://es.wikipedia.org/wiki/Informaci%C3%B3n) donde un nodo [emisor](https://es.wikipedia.org/wiki/Emisor) envía información a una multitud de nodos [receptores](https://es.wikipedia.org/wiki/Comunicaci%C3%B3n) de manera simultánea, sin necesidad de reproducir la misma transmisión [nodo](https://es.wikipedia.org/wiki/Nodo_(inform%C3%A1tica)) por nodo.

Para el ejemplo, la direccion de red es 154.25.0.0/255.255.0.0. La direccion de broadcast para esta red es 154.25.255.255, que corresponde a la máxima dirección posible de cada subred.

d) El mecanismo para generar las subredes es a través de la modificación de los octetos de la máscara dependiendo la clase de la red

e)Ejercicio: Se desea dividir la red clase B 117.39.0.0 en 4 sub-redes que tengan la misma cantidad de hosts cada una

1) Al ser una red clase B la mascara natural es 255.255.0.0

2)la máscara para generar las subredes es 255.255.0.0-255.255.64.0-255.255.128.0-255.255.192.0

(tercer octeto con el cual separamos las sub redes)

00xxxxxx-00000000 (0)

01xxxxxx-01000000 (64)

10xxxxxx-10000000 (128)

11xxxxxx-11000000 (192)

3) 117.39.0.0 - 117.39.63.255

117.39.64.0 - 117.39.127.255

117.39.128.0 - 117.39.191.255

117.39.192.0 - 117.39.254.255

4) Se pueden conectar (16383 direcciones posibles - 2 direcciones prohibidas) host por cada subred

5) Se pueden conectar (65536 direcciones posibles - 8 direcciones prohibidas) host a toda la red.

6) 1-0.999908=0.000091555=0.0091%

f) 2) En el caso de que queremos dividir en 8 la primer sub red tendriamos:

(tercer octeto con el cual separamos las sub redes)

00000xxx-00000000 (0)

00001xxx 00001000 (8)

00010xxx-00010000 (16)

00011xxx-00011000 (24)

00100xxx-00100000 (32)

00101xxx-00101000 (40)

00110xxx-00110000 (48)

00111xxx-00111000 (56)

Funciones de bibliotecas de C para manipulación de direcciones IP

1. https://www.systutorials.com/docs/linux/man/3-inet\_aton/

***Funciones de bibliotecas de C para manipulación de direcciones IP***

***a) Explore las diferentes funciones disponibles para la manipulación de direcciones IP (busque inet\_aton() entre otras).***

· inet\_addr(): convierte una dirección IP dada en la notación de cifras y puntos en un unsignedlong . La asignación se puede hacer:

o ina.sin\_addr.s\_addr = inet\_addr("10.12.110.57");

· inet\_aton() ("aton" es " asciitonetwork"): devuelve un valor distinto de cero si tiene éxito, y cero cuando falla. Y la dirección se devuelve en inp .

· inet\_ntoa() ("ntoa" significa "networktoascii ") según se muestra a continuación:

o printf("%s", inet\_ntoa(ina.sin\_addr)); Eso imprimirá la dirección IP

Toma un struct in\_addr como argumento, y no un long, y devuelve un puntero a char. Éste apunta a una zona estática de memoria dentro de inet\_ntoa(), así que cada vez que se llame a inet\_ntoa() se perderá la última dirección IP que pediste. Por ejemplo:

char \*a1, \*a2;

a1 = inet\_ntoa(ina1.sin\_addr); // esta es 192.168.4.14

a2 = inet\_ntoa(ina2.sin\_addr); // esta es 10.12.110.57

printf("address 1: %s\n",a1);

printf("address 2: %s\n",a2);

imprimirá: address 1: 10.12.110.57 address 2: 10.12.110.57

***b) Explique cómo es el orden de transmisión cuando se transmiten palabras multibyte en la red: ¿Qué se transmite antes, el byte más significativo o el menos significativo? Ejemplifique para un unsignedlong y para un unsigned short.***

Existen dos formas distintas de ordenación de bytes (a veces se les llama "octetos"): primero el byte más significativo, o primero el byte menos significativo. A la primera forma se la llama "Ordenación de bytes de la red" [Network Byte Order]. Algunos ordenadores almacenan internamente los números según la Ordenación de bytes de la red, mientras que otros no.

***c) Busque cuales son las funciones para convertir el orden de los bytes entre el utilizado por la red y el utilizado por la computadora.***

htons() -- "Host to Network Short " (short de máquina a short de la red)

htonl() -- "Host to Network Long" (long de la máquina a long de la red)

ntohs() -- "Network to Host Short " (short de la red a short de máquina)

***d) En una computadora basada en 80x86 se desea transmitir por la red los siguientes números: 32769 y 65538. Sabiendo que se representarán como enteros sin signo que ocupen la menor cantidad posible de almacenamiento correspondiente a un tipo de datos en C, determine cuál sería el tipo de cada uno, como se almacenarían en memoria y como se transmitirían por la red.***

***Funciones de bibliotecas de C para manipulación de direcciones IP***

***a) Explore las diferentes funciones disponibles para la manipulación de direcciones IP (busque inet\_aton() entre otras).***

· inet\_addr(): convierte una dirección IP dada en la notación de cifras y puntos en un unsignedlong . La asignación se puede hacer:

o ina.sin\_addr.s\_addr = inet\_addr("10.12.110.57");

· inet\_aton() ("aton" es " asciitonetwork"): devuelve un valor distinto de cero si tiene éxito, y cero cuando falla. Y la dirección se devuelve en inp .

· inet\_ntoa() ("ntoa" significa "networktoascii ") según se muestra a continuación:

o printf("%s", inet\_ntoa(ina.sin\_addr)); Eso imprimirá la dirección IP

Toma un struct in\_addr como argumento, y no un long, y devuelve un puntero a char. Éste apunta a una zona estática de memoria dentro de inet\_ntoa(), así que cada vez que se llame a inet\_ntoa() se perderá la última dirección IP que pediste. Por ejemplo:

char \*a1, \*a2;

a1 = inet\_ntoa(ina1.sin\_addr); // esta es 192.168.4.14

a2 = inet\_ntoa(ina2.sin\_addr); // esta es 10.12.110.57

printf("address 1: %s\n",a1);

printf("address 2: %s\n",a2);

imprimirá: address 1: 10.12.110.57 address 2: 10.12.110.57