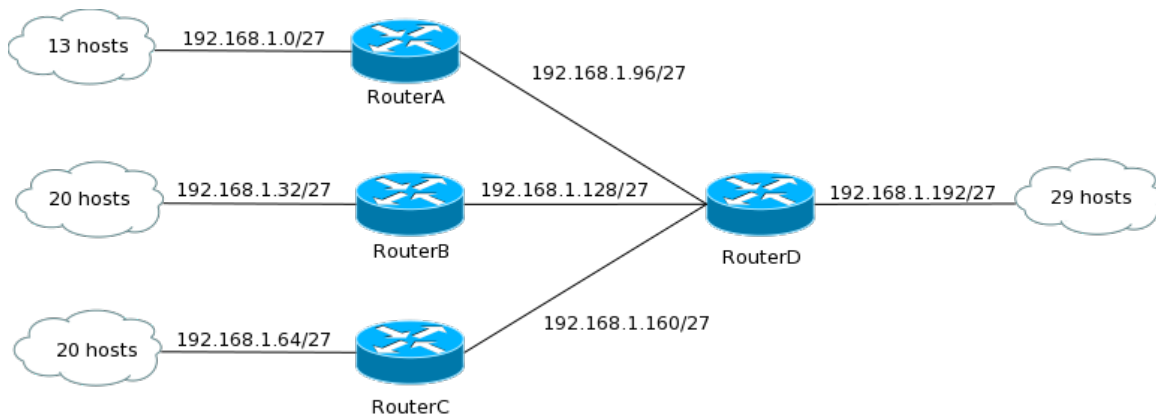


Parte 1 (40%)

Variable Length Subnet Masks (VLSM)

La red de la Figura 1 usa máscaras fijas (como las vistas en clase) o esquema de classfull Network. Los estudiantes debe de hacer una investigación básica sobre VLSM para resolver el problema a continuación:

Figura 1 *Classful Network*



En VLSM, se usa más dar la información /27. Este /27 significa que es una máscara 255.255.255.224 . Es decir, tenemos 3 bits para subredes, y por consiguiente, se podrán tener 30 Hosts. En mascarar fijas, toda la red debe de tener la misma máscara.

Se deben de contestar las siguientes preguntas:

1. (15%)
 - a. Explique los terminos Classfull y Classless?
 - b. Porque se usa VLSM?
 - c. Cuales son las ventajas?
 - d. Que requisitos deben de tener los routers en la red.
 - e. Cual es el concepto de bloques de direcciones?
 - f. Cual sería un procedimiento recomendado para trabajar con VLSM, es decir, como deberíamos resolver el problema de asignación de direcciones IP cuando se utiliza VLSM?
2. (25%) Resuelva la Topología de la figura 1
 - a. Cuanto desperdicio hay con mascarar fijas (así como está resuelto el problema de la figura 1.
 - b. De el procedimiento para resolver esta topología. Al menos deben de mencionar cuantas direcciones IP se requieren por red de Hosts y enlaces entre routers (en la conexión entre routers no existes hosts a conectar, solo los 2 puertos dibujados)
 - c. Dé el esquema de direccionamiento usando VLSM de la figura 1. Vuelva a dibujar la topología con el esquema de direccionamiento.
 - d. Cual es la eficiencia en el uso de direcciones IP usando VLSM, de el ahorro de direcciones IP.

Parte 2 (60%)

Trabajo de Investigación y Programación de Sockets con TCP y UDP

Trabajo Básico

Los procesos de aplicación envían mensajes a través de sockets. Existen dos protocolos en el nivel de transporte, TCP y UDP.

Una aplicación típica de red, consiste en dos programas, un programa cliente y un programa servidor. Cuando estos dos programas se ejecutan, se crean un proceso cliente y un proceso servidor. Estos procesos se comunican entre sí leyendo de y escribiendo en sockets.

Típicamente, todo esto se hace en dos computadoras (hosts), pero también se pueden hacer en una sola computadora o host. Investigar y explicar como se hace en una sola computadora.

Estos programas deben de hacerse en Python versión 2.7.

Se debe investigar los conceptos de cómo hacer estos 4 programas, en TCP servidor TCP, cliente TCP, servidor UDP y cliente UDP. Buscar en Internet los conceptos de cómo escribir estos programas. El trabajo debe de explicar estos conceptos.

En TCP (orientado a conexión y con confiabilidad), el servidor tiene que estar listo para responder pedidos del cliente y debe de tener un socket de bienvenida. El cliente crea un socket y especifica el destino. Debe de crear el 3 way handshake.

En UDP (sin conexión y sin confiabilidad), el servidor no tiene que estar listo ya que no hay un handshake. El servidor crea un socket y no hay un socket de bienvenida. El cliente crea un socket e inicia.

Existen ejemplos de esto en la literatura. El estudiante debe de explicar cada instrucción de los programas, explicar las instrucciones utilizadas, y hacer un diagrama de bloques de los programas.

Lo ideal es hacerlo para ambiente MAC, pero sino puede hacerse en otras plataformas.

Para probar los programas, se deben de transmitir una serie de caracteres desde el cliente y estos deben de ser recibidos en el servidor y cambiados a mayúsculas y volverlos a enviar en mayúscula al cliente desde el servidor.

Debe de tener una forma para terminar el envío de caracteres. Deben de especificar cuantas conexiones cliente/servidor son posibles. Cual es el tamaño máximo del datagrama..