

Solución Examen – 30 de julio de 2015

(ref: solredes20150730.odt)

Instrucciones

- Indique su nombre completo y número de cédula en cada hoja.
- Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.
- Escriba las hojas de un solo lado y utilice una caligrafía claramente legible.
- Comience cada pregunta teórica y cada ejercicio en una hoja nueva.
- Sólo se responderán dudas de letra. No se responderán dudas de ningún tipo los últimos 30 minutos del examen.
- El examen es individual y sin material. Apague su teléfono celular mientras esté en el salón del examen.
- Es obligatorio responder correctamente al menos 15 puntos en las preguntas teóricas y 20 de los problemas prácticos. Los puntos ganados en el curso se suman a los puntos de teórico.
- El puntaje mínimo de aprobación es de 60 puntos.
- Para todos los ejercicios, si es necesario, puede suponer que dispone de los tipos de datos básicos (p.ej. lista, cola, archivo, string, etc.) y sus funciones asociadas (ej: tail(lista), crear(archivo), concatenar(string, string)).
- Justifique todas sus respuestas.
- Duración: 3 horas. Culminadas las 3 horas el alumno no podrá modificar las hojas a entregar de ninguna forma.

Preguntas Teóricas

Pregunta 1 (9 puntos)

- Describe el funcionamiento del protocolo ARP.
- Un conmutador (switch) de capa 2 puro, ¿tiene tabla ARP?. Justifique su respuesta.
- Dada una red conmutada (compuesta por varios switches), considere dos PCs comunicándose entre ellos. ¿En algún momento de esa comunicación existen tramas con dirección MAC (origen o destino) que corresponda a cualquiera de los switches?

Solución

- Address Resolution Protocol* (ARP) resuelve el problema de comunicación dentro de la LAN, donde además de las direcciones IP se necesitan las direcciones MAC para que efectivamente las tramas se transfieran desde el origen al destino. Cada *host* tiene una tabla ARP, que contiene las correspondencias entre direcciones IP y direcciones MAC, y un campo TTL que indica la validez de dicha entrada en la tabla. Cuando un *host* necesita comunicarse con otro *host* de la LAN del cual conoce su dirección IP, si hay una entrada válida en la tabla ARP se comunica directamente, y en caso contrario utiliza el protocolo ARP para obtener la correspondencia. El mensaje *ARP Request* es encapsulado en una trama ethernet con MAC origen la del *host* origen y MAC destino FF-FF-FF-FF-FF-FF (*broadcast* ethernet), que contiene en su *payload* los campos ARP (MAC origen, IP origen, MAC destino - desconocida, se pone en todos ceros-, IP destino). Por ser un *broadcast* ethernet todos los *host* reciben la trama, y el Request sólo será respondido por el nodo que tiene la IP destino. La respuesta ARP tiene los mismos campos, solamente que ahora los datos de origen serán los del *host* que antes era destino, del que ahora se conoce la MAC.
- No, ya que al ser de capa 2 puro solo funciona a nivel de ethernet. ARP funciona entre equipos que manejan la capa IP y superiores.
- Los switches reenvían tramas de una interfaz a la otra, y mantienen siempre la MAC origen y destino. Se dice que los switches "no tienen MAC".

Pregunta 2 (6 puntos)

Suponga que se tiene una red de circuitos (en oposición a una red de conmutación de paquetes) en la que no hay pérdida ni errores de bits, y los mismos llegan al destino en orden. En este contexto, ¿es posible prescindir de mecanismos de confiabilidad y control de congestión como los que implementa TCP? Justifique.

Solución

Se supone que el circuito es creado solamente cuando es posible asegurar los recursos a lo largo de todo el camino de extremo a extremo, y por lo tanto, en las hipótesis planteadas, podemos prescindir de mecanismos de confiabilidad y control de congestión como los que implementa TCP.

Sin embargo, si consideramos los *hosts* en los extremos del circuito, aún podríamos tener control de acceso y la necesidad de repartir el ancho de banda disponible entre aplicaciones, y en ese caso la multiplexación y *fairness* de TCP serían necesarias.

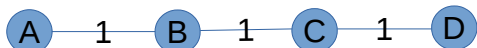
Redes de Computadoras, Introducción a las Redes de Computadoras y Comunicación de Datos
Finalmente, si la red de circuitos no asegura la recuperación "automática" ante fallos, se necesitaría alguna función de recuperación a nivel de transporte y/o aplicación.

Pregunta 3 (8 puntos)

Explique el fenómeno de conteo a infinito en el algoritmo de vector-distancia, y ejemplifique en una red de cuatro nodos.

Solución

El fenómeno del conteo a infinito es un problema del algoritmo de vector-distancia, que ante un aumento de costo (o la caída) de un enlace, demora en converger a la nueva situación mediante el intercambio iterativo de los vectores distancia entre nodos, generando posibles loops transitorios. El problema se origina en que cada nodo solo almacena la distancia calculada para cada destino, y los vecinos next-hop hacia dichos destinos. Consideremos la red de la figura:



En este caso tenemos que el vector distancia de A es (1,2,3) para alcanzar a B, C y D respectivamente, siempre pasando por B. Por su lado el de B es (1, 1, 2) para (A, C, D) pasando por A, C y C respectivamente.

Suponga que el costo del enlace BC cambia a 60. B recalcula su vector y lo envía a A, el cual avisa a B que tiene un camino a C de costo 2, y por lo tanto B recalcula su costo a C en 3 pasando por A (absurdo si uno mira el grafo). A recibe esta información, y actualiza su vector distancia, aumentando en 1 el costo, llegando con costo 4 a C a través de B. Esta interacción se dará hasta que el costo de las iteraciones supere a 60, en cuyo caso B decidirá que puede llegar a C a través del enlace BC, en lugar de enviar erróneamente el tráfico hacia A, generando un loop de tráfico.

Pregunta 4 (7 puntos)

Explique el concepto de "violación de capas", y mencione dos ejemplos en TCP/IP, explicando por qué se produce dicha violación.

Solución

El concepto refiere a que un protocolo de una capa haga uso de campos de información de otra capa del stack, y/o que la implementación de determinado protocolo de capa N haga uso de información de capa N-1 u otra capa. El caso más común es el de ARP, donde el protocolo incluye direcciones de capa 2 (MAC) y capa 3 (IP). Otro ejemplo son los protocolos de enrutamiento, los que intercambian datos de rutas (capa 3) utilizando para ello protocolos de capas superiores, por ejemplo sesiones TCP entre enrutadores BGP.

Pregunta 5 (10 puntos)

- Explique por qué se dice que TCP intenta ser un protocolo justo (fair).
- Suponga que usted es desarrollador de una aplicación que utiliza TCP, ¿qué podría hacer para intentar obtener un mayor ancho de banda para dicha aplicación?
- ¿Cómo podría obtener resultados similares si tiene acceso al sistema operativo?
- ¿Por qué UDP no es un protocolo fair?

Solución

- Si suponemos K conexiones TCP (lo suficientemente duraderas) que atraviesan un enlace cuello de botella cuyo ancho de banda R bps es menor a la suma de los anchos de banda de las K conexiones individuales, y además suponemos que no existe tráfico UDP en la red, entonces los algoritmos de control de congestión de TCP llevarán a que la tasa de transmisión de cada conexión individual tienda a R/K. A este fenómeno se le denomina *fairness*.
- Haría que mi aplicación abriera conexiones TCP concurrentes, para lograr un mayor ancho de banda en total para la aplicación, aplicando el concepto de *fairness* en forma egoísta.
- Configurando los parámetros de TCP para todo el sistema operativo, por ejemplo, para que no respeten el control de congestión. De esta forma, haría cambios en forma global al sistema y para todas las aplicaciones.
- Dado que UDP no tiene control de congestión, puede enviar segmentos a la velocidad que le permita el medio (o le demande la aplicación) y no se auto-limita acorde al estado de la red. De esta forma, UDP podría consumir un mayor ancho de banda frente a las conexiones TCP que coexistan en el mismo tiempo.

Problemas Prácticos

Problema 1 (30 puntos)

Implemente la función **void forwardPacket(Packet packet, Interface inInterface)** la cual dado un paquete y la interfaz de entrada del paquete realiza el reenvío (forwarding) del mismo.

Para esto deberá definir la estructura de la tabla de forwarding y los métodos de acceso a la misma que utilizará la función; dicha tabla se mantiene actualizada por el algoritmo de enrutamiento.

Además, la función deberá realizar correctamente el forwarding de paquetes de multicast. Para esto deberá utilizar la técnica Reverse Path Forwarding. Todos los routers pertenecen a todos los grupos de multicast activos.

Para la implementación se cuenta con las siguientes funciones auxiliares:

Address getSourceAddress (Packet packet) – Devuelve la dirección origen del paquete.

Address getDestinationAddress (Packet packet) – Devuelve la dirección destino del paquete.

void enqueuePacket (Packet packet, Interface outInterface) – Coloca un paquete en la cola de salida de una interfaz.

El tipo Address es un entero sin signo de 32 bits.

Solución

La tabla de routing tiene las siguientes entradas: Prefijo, Máscara, Gateway e Interfaz.

```
typedef struct ForwardingEntry{
    Address prefix;
    Address mask;
    Address gway;
    Interface if;
} RoutingEntry;

ForwardingEntry* forwardingTable;
int tableLength;

void forwardPacket(Packet packet, Interface inInterface)
{
    Address dest = getDestinationAddress (packet);
    if (packet.ttl > 1)
    {
        packet.ttl--;
        if (!isMulticast(dest))
        {
            Interface outIf = getOutInterface (dest);
            enqueuePacket (packet, outIf);
        }
        else
        {
            Address source = getSourceAddress (packet);
            Interface outIf = getOutInterface (source);
```

```
    if (outIf == inInterface)
    {
        for (all interfaces if <> inInterface)
            enqueuePacket (packet, if);
    }
    // si no, no se hace nada, drop packet.
}
}
else
{
    //enviar ICMP con ttl expire
}
}
```

```
Interface getOutInterface (Address destination)
{
    Address maxMask = 0x00000000;
    Interface matchingIf;
    for (int i=0; i<tableLength; i++) {
        if ((destination & forwardingTable[i].mask) ^ forwardingTable[i].prefix == 0) {
            if (forwardingTable[i].mask > maxMask) {
                matchingIf = forwardingTable[i].if;
                maxMask = forwardingTable[i].mask;
            }
        }
    }
    return matchingIf;
}
```

```
bool isMulticast (Address addr)
{
    Address multicast = 0xE0000000; // una dir de multicast siempre comienza con 1110
    Address mask = 0xF0000000;
    Address aux = addr & mask;
    if (aux ^ multicast == 0)
        return true;
    else
        return false;
}
```

Problema 2 (30 puntos)

Se considera una red académica con tres sitios A, B y C. En cada sitio se dispone de un conmutador (switch) de capa 2 con soporte de VLANs. Los sitios tienen una conectividad mallada entre los conmutadores, donde cada enlace es un trunk que puede transportar las VLANs definidas. Existen subredes diferenciadas para docentes, estudiantes, funcionarios y visitantes repartidas entre los tres sitios, y se estima que las necesidades de puestos de trabajo para cada subred no superarán los 100, 500, 40 y 254 respectivamente. Se han realizados mediciones, y se verifica que el tráfico entre subredes se realiza fundamentalmente dentro de cada sitio. La asignación de direcciones IP se realiza mediante un servidor DHCP para cada subred; para numerar toda la red se cuenta con el prefijo 10.10/16.

Se pide:

- a) Asigne prefijos a cada subred, minimizando el uso de direcciones IP y reservando la mayor cantidad posible de direcciones para uso futuro.
- b) Con la topología propuesta, ¿es posible comunicar puestos de trabajo de distintas subredes?
 - i) Si su respuesta es positiva, explique detalladamente como enviar un datagrama entre la red de docentes y la red de estudiantes.
 - ii) Si su respuesta es negativa:
 - 1) Proponga una solución para que las subredes se puedan comunicar entre sí, minimizando la cantidad de dispositivos a agregar, y especifique las características de dichos dispositivos.
 - 2) Proponga una solución para que las subredes se puedan comunicar entre si, buscando minimizar el tráfico entre sitios y especifique las características de los dispositivos a agregar.

Solución

a) Cada subred debe soportar los puestos de trabajo y además su servidor DHCP. Por lo tanto se deben asignar prefijos:

/25 (hasta 126 direcciones IP) para docentes

/23 (hasta 510 direcciones IP) para estudiantes

/26 (hasta 62 direcciones IP) para funcionarios

/23 (hasta 510 direcciones IP) para visitantes; no alcanza con /24 (hasta 254) porque se necesitan por lo menos 254 puestos más la interfaz del servidor DHCP.

Una posible asignación:

10.10.0.0/25 - docentes

10.10.0.128/26 - funcionarios

10.10.2.0/23 - docentes

10.10.4.0/23 - visitantes

b) No es posible porque las subredes no tienen un gateway de capa 3 para poder intercambiar paquetes. Los switches de capa 2 no lo permiten.

b.ii.1) Se debe agregar un "router in a stick" conectado a alguno de los switches, es decir, un router que tenga una interfaz en c/u de las subredes; se puede implementar con interfaces físicas o con un trunk y sub-interfaces. Hay que destinar una dirección IP en cada subred para c/u de las interfaces de este router.

La tabla de forwarding de este router sería:

Destino	Gateway	Máscara de subred	Interfaz
10.10.0.0	Directamente Conectada	255.255.255.128 (/25)	if1
10.10.128.0	Directamente Conectada	255.255.255.192 (/26)	if2
10.10.2.0	Directamente Conectada	255.255.254 (/23)	if3
10.10.4.0	Directamente Conectada	255.255.254 (/23)	if4

b.ii.2) Dado que se midió el tráfico y el mismo es fundamentalmente local a cada sitio, se debe instalar un "router in a stick" en c/u de los sitios. Hay que destinar una dirección IP en cada subred para c/u de las interfaces del router de cada sitio (en total 3 routers, 12 interfaces). Además el servidor DHCP debería manejar rangos por sitio para que la configuración del gateway sea coherente en cada sitio.