

Redes de Computadoras
Solución – 21 de diciembre de 2013
(ref: solredes1312.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 (8 puntos)

El mecanismo SACK (*Selective Acknowledgement*) se utiliza de la siguiente manera en una sesión TCP: mediante el campo de Opciones del encabezado TCP el receptor le anuncia al emisor los bloques no contiguos de segmentos que llegaron exitosamente, por lo que éste debe retransmitir los segmentos perdidos. Por ejemplo, si se recibe un segmento en el que su campo Opciones es el siguiente:

100	200	220	240	245	250
-----	-----	-----	-----	-----	-----

significa que fueron correctamente recibidos los segmentos

100..199, 220..239 y 245..249, habiéndose perdido los segmentos 200..219 y 240..244. Considerando que en el campo Opciones se puede incluir la información respecto a pocos bloques (a lo sumo 10), analice la conveniencia de utilizar la estrategia SACK en los siguientes contextos:

- a) muchas pérdidas de segmentos en ráfagas.
- b) muchas pérdidas de segmentos no secuenciales.

Solución:

La manera en la que está implementada la señalización de segmentos perdidos utilizada por SACK condiciona su eficiencia en determinados contextos

- a) En este caso, si las ráfagas son pocas, aun cuando cada ráfaga implique un número significativo de segmentos, es posible llevar dicha información en un solo segmento hacia el emisor, por lo que en este caso sí conviene el uso de la estrategia SACK.
- b) En este otro caso, dada la gran granularidad que se observa en los segmentos perdidos, SACK no es conveniente utilizarlo ya que no permitirá anunciar en un solo segmento que viaje hacia el emisor cuáles fueron los perdidos y por lo tanto la estrategia resulta de muy baja eficiencia.

Pregunta 2 (6 puntos)

Considere el entramado de conmutación de un router:

- a) Describa brevemente las alternativas para realizar la conmutación de paquetes.
- b) Describa el fenómeno de bloqueo HOL (*Head Of Line* – Cabeza de Línea).

Solución:

- a) Ref. Sección 4.3.2 libro del curso.
- b) Ref. Sección 4.3.4 libro del curso.

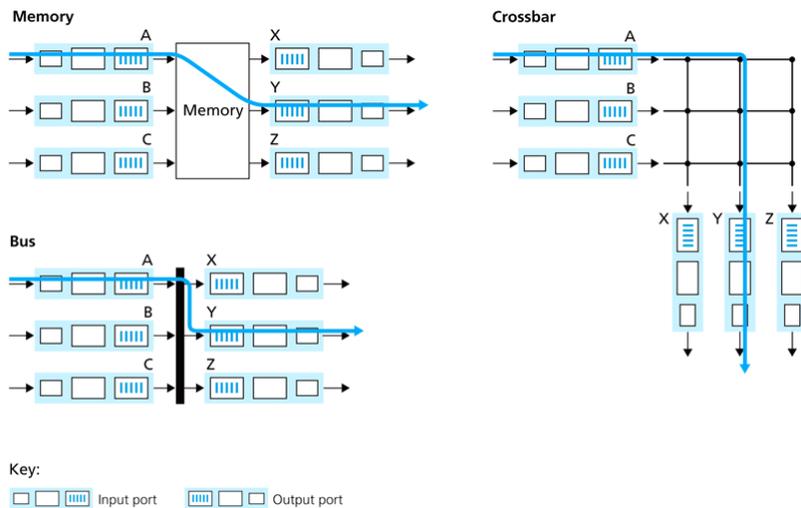


Figure 4.8 ♦ Three switching techniques

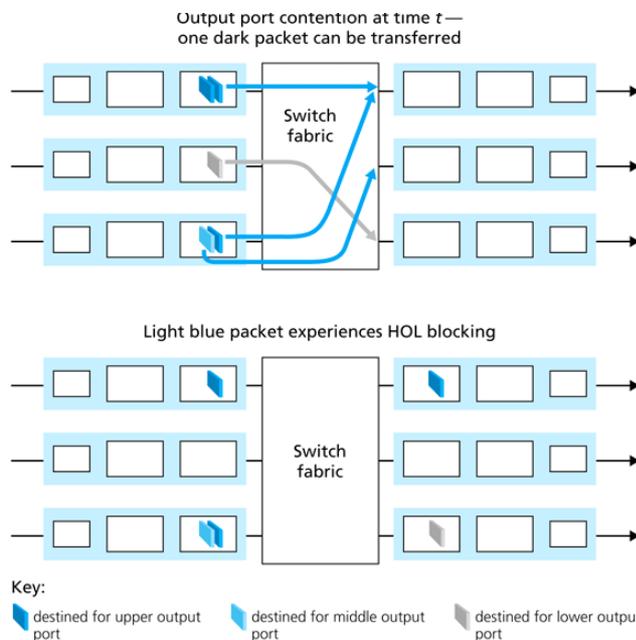


Figure 4.11 ♦ HOL blocking at an input queued switch

Pregunta 3 (10 puntos)

- a) ¿Qué es una *Virtual LAN* (VLAN)? Describa su utilidad, y muestre un ejemplo.
- b) Suponga que se cuenta con 2 conmutadores (*switches*) con varias VLAN definidas, interconectados mediante un enlace troncal (*trunk*). Cuáles son las diferencias entre las tramas que se transmiten en el enlace troncal y las que se transmiten en los enlaces con los sistemas finales (*hosts*).
- c) ¿Es posible cursar tráfico entre VLANs diferentes a nivel de capa de enlace o requiere la intervención de las capas superiores?

Solución:

- a) Es una tecnología que permite la creación de redes virtuales lógicas dentro de la misma red física en un entorno de conectividad de capa 2 ethernet. Las VLANs definen dominios de broadcast independientes para cada red virtual, que normalmente se corresponde con una subred IP. En un conmutador es posible definir VLANs de acuerdo a diferentes criterios, como por ejemplo, asignar un conjunto de puertos a cada VLAN. Los *hosts* conectados a una determinada VLAN tienen conectividad de capa de enlace aislada del resto de las VLANs definidas.

Redes de Computadoras

Como ejemplo consideremos una empresa con varios departamentos, los que se encuentran distribuidos en diferentes sitios geográficos. La solución permite definir VLANs por departamento, y que en cada sitio se puedan asignar *hosts* a las diferentes VLANs. Por lo tanto, los diferentes departamentos tienen presencia en todos los sitios, sin necesidad de conmutadores especializados para cada subred.

- b) Los enlaces troncales transmiten tramas que incluyen información adicional de identificación de VLANs (etiquetas o *tags* de VLANs), según el protocolo 802.1q, mientras que los *hosts* utilizan tramas ethernet "normales". Es decir, los *host* no tienen conocimiento del uso de VLANs.
- c) Dado que las VLANs definen dominios de broadcast (subredes) de capa 2 aislados, para cursar tráfico entre diferentes VLANs es necesario utilizar los servicios de capa de red, es decir, se necesita un enrutador que interconecte las VLANs a nivel IP.

Pregunta 4 (8 puntos)

Considere el protocolo DHCP:

- a) Describa los mensajes que se intercambian en el protocolo y entre que entidades. ¿Qué protocolo de transporte utiliza?
- b) ¿Cuál es su utilidad? ¿Lo utilizaría en una red de servidores que implementan el balanceo de carga de un sitio web?

Solución:

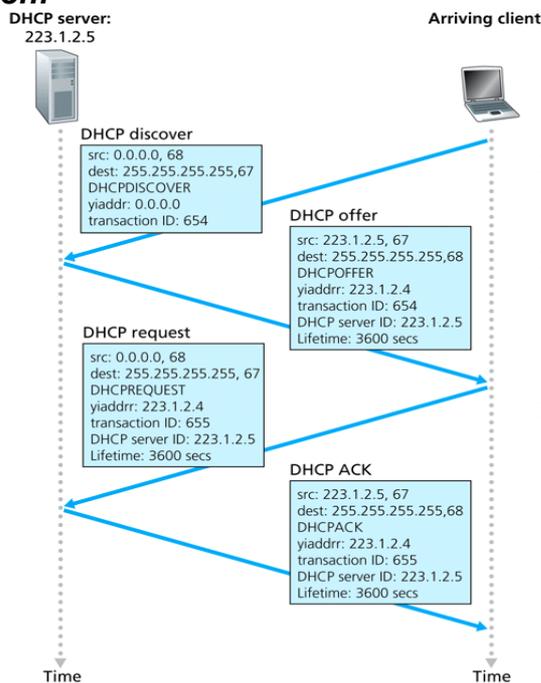


Figure 4.21 ♦ DHCP client-server interaction

a) En la figura se muestran los mensajes intercambiados entre el servidor DHCP y un cliente. Los mensajes se transportan en UDP.

b) El protocolo DHCP permite la autoconfiguración de clientes que no están conectados permanentemente a una red, en particular la asignación de direcciones IP, *default gateway*, DNS.

En el caso de una red de servidores, por definición son sistemas conectados permanentemente y deben ser alcanzables por su nombre (definido en el DNS); por lo tanto, no es conveniente asignar direcciones IP temporales a los servidores.

Pregunta 5 (8 puntos)

Se considera el servicio DNS:

- a) Describa una forma de utilizarlo para distribuir la carga de un sitio web replicado en varios servidores.
- b) De un ejemplo de consulta recursiva, y un ejemplo de consulta iterativa. Describa cual es la utilidad de cada tipo de consultas.

Solución:

- a) Los sitios están replicados en varios servidores, cada uno en un sistema terminal distinto con una dirección IP diferente. Estos servidores tienen un conjunto de direcciones IP asociado con un único nombre de host canónico, almacenado en la base de datos DNS. Cuando el servidor recibe una consulta de un cliente, responde con el conjunto de direcciones pero va rotando las direcciones en la lista de forma que el primero de la lista sea siempre uno distinto. Los clientes usualmente usan la primer dirección de la lista, por lo que sucesivas peticiones DNS son dirigidas a los distintos servidores. (De la sección 2.5.1 del Libro del Curso)
- b) Las consultas recursivas minimizan el número de mensajes usados para resolver una consulta, y la implementación del nodo que inicia la consulta es más simple. Las consultas iterativas minimizan la carga sobre los servidores DNS. En la práctica, lo más común es que el primer nivel (al DNS local) sea recursivo, y el resto iterativo. (De la sección 2.5.2 del Libro del Curso)

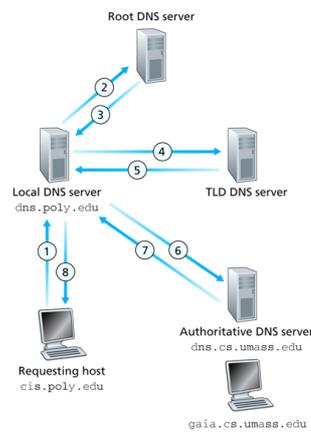


Figure 2.21 ♦ Interaction of the various DNS servers

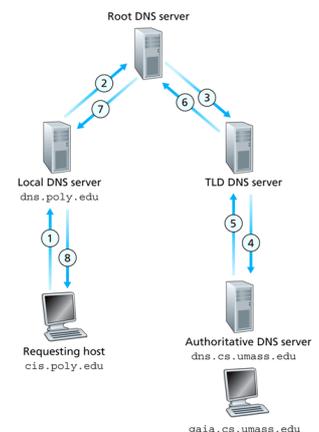


Figure 2.22 ♦ Recursive queries in DNS

Problemas Prácticos

Problema 1 (30 puntos)

Se desea implementar un sistema P2P para compartir archivos. El sistema consiste de aplicaciones cliente que realizarán la transferencia de archivos y una aplicación de tracker para registrar la ubicación de los archivos compartidos. El funcionamiento general del sistema para la transferencia de un archivo consiste en:

- El cliente que comparte el archivo (uploader) coloca el archivo en una lista local de archivos compartidos.
- El uploader escucha por pedidos en una dirección (IP y Puerto) dada.
- El uploader registra su dirección y sus archivos compartidos en el tracker.
- El cliente que desea descargar el archivo (downloader) consulta al tracker por el archivo y obtiene la dirección del uploader de ese archivo.
- El downloader realiza el pedido del archivo al uploader.
- El uploader inicia la transferencia del archivo al downloader.

El tracker escucha pedidos en ipTracker, puertoTracker. La comunicación entre cliente y tracker utiliza los siguientes comandos:

- SEARCH <nombre archivo>
Para la búsqueda de un archivo.
Como respuesta se obtendrá 'fail' o el resultado de la búsqueda con un peer disponible en el siguiente formato:
FILE <ip:puerto>
- SHARE <nombre archivo> <ip:puerto>
Para el registro de un archivo.
Como respuesta se obtendrá 'ok' en caso de éxito o 'fail' en otro caso.

Se pide

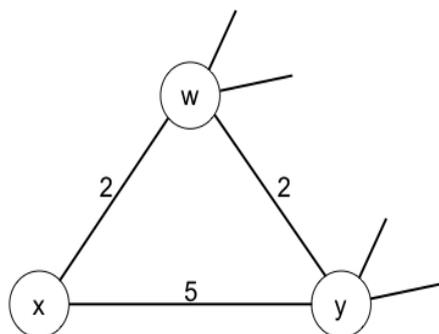
Implementar los siguientes servicios de la aplicación cliente que debe funcionar como uploader y downloader:

- *File download (String nomArchivo)* que es utilizado por los usuarios para comenzar la descarga de un archivo. Dado un nombre de archivo obtiene un peer con ese archivo y lo descarga.
- *int share (String nomArchivo, File archivo)* es utilizado por los usuarios para marcar un archivo como compartido en la red P2P, devuelve 0 en caso de éxito y -1 en otro caso.
- *void upload ()* es el servicio encargado de la atención de pedidos de otros clientes.

Nota:

Tiene disponibles todas las primitivas de socket y entrada/salida estandar.

Problema 2 (30 puntos)



Se considera el fragmento de red que muestra la figura. El nodo **x** está directamente conectado a dos vecinos **w**, **y**. Los nodos **w**, **y** tienen costos mínimos 5 y 6 respectivamente a un nodo **u** que no se muestra en la figura. Todos los costos en la red tienen valores enteros positivos.

- Represente el vector de distancias de **x** para los destinos **w**, **y**, **u**.
- Expresé un cambio del costo $c(x, w)$ o $c(x, y)$ de forma tal que **x** informe a sus vecinos de un nuevo camino de costo mínimo hacia **u** como resultado de ejecutar el algoritmo de vector-distancia.
- Expresé un cambio del costo $c(x, w)$ o $c(x, y)$ de forma tal que **x** NO informe a sus vecinos de un nuevo camino de costo mínimo hacia **u** como resultado de ejecutar el algoritmo de vector-distancia.

Solución Problema 1

```

File download (String nomArchivo) {
    Socket sktTracker = new Socket()
    sktTracker.connect(ipTracker, puertoTracker)
    sktTracker.send(construir_mensaje("SEARCH", nomArchivo))
    respuesta = sktTracker.read()
    sktTracker.close()
    if (respuesta != "fail") {
        ipPeer = obtenerIP(respuesta)
        portPeer = obtenerPuerto(respuesta)
        Socket sktPeer = new Socket()
        sktPeer.connect(ipPeer, portPeer)
        sktPeer.send(construir_mensaje("GET", nomArchivo))
        File archivo = sktPeer.read()
        sktPeer.close()
        return archivo
    } else {
        return null
    }
}

int share(String nomArchivo, File archivo) {
    Socket skt = new Socket()
    skt.connect(ipTracker, puertoTracker)
    skt.send(construir_mensaje("SHARE", nomArchivo, myIP:myPort))
    respuesta = skt.read()
    skt.close()
    if (respuesta != "ok") {
        return -1
    } else {
        listaCompartidos.add(archivo)
        return 0
    }
}

void upload () {
    Socket master = new Socket()
    master.bind(myIP, myPort)
    while true {
        Socket client = master.accept()
        runThread(uploadAux, client) //ejecuta en un nuevo thread la
funcion uploadAux
    }
}

void uploadAux (Socket skt){
    respuesta = skt.read()
    if (isGet(respuesta)){
        nombreArchivo = parseNombre(respuesta)
        File archivo = open(nomArchivo)
        skt.send(archivo)
        skt.close()
    }
}
}

```

Solución Problema 2

a)

- Dx(w) = 2
- Dx(y) = 4
- Dx(u) = 7

La ejecución del algoritmo de vector de distancias para llegar a esos valores es (el tiempo avanza de arriba a abajo):

en w	u	w	x	y
Dw	5	0	2	2
Dx	inf	inf	inf	inf
Dy	inf	inf	inf	inf

en x	u	w	x	y
Dw	inf	inf	inf	inf
Dx	inf	2	0	5
Dy	inf	inf	inf	inf

en y	u	w	x	y
Dw	inf	inf	inf	inf
Dx	inf	inf	inf	inf
Dy	6	2	5	0

en w	u	w	x	y
Dw	5	0	2	2
Dx	inf	2	0	5
Dy	6	2	5	0

en x	u	w	x	y
Dw	5	0	2	2
Dx	7	2	0	4
Dy	6	2	5	0

en y	u	w	x	y
Dw	5	0	2	2
Dx	inf	2	0	5
Dy	6	2	4	0

en w	u	w	x	y
Dw	5	0	2	2
Dx	7	2	0	4
Dy	6	2	4	0

en x	u	w	x	y
Dw	5	0	2	2
Dx	7	2	0	4
Dy	6	2	4	0

en y	u	w	x	y
Dw	5	0	2	2
Dx	7	2	0	4
Dy	6	2	4	0

b)

Subir c(x,w) a 7 o más. Lo que ocurre es lo siguiente:

en w	u	w	x	y
Dw	5	0	6	2
Dx	7	2	0	4
Dy	6	2	4	0

en x	u	w	x	y
Dw	5	0	2	2
Dx	11	7	0	5
Dy	6	2	4	0

en y	u	w	x	y
Dw	5	0	2	2
Dx	7	2	0	4
Dy	6	2	4	0

en w	u	w	x	y
Dw	5	0	6	2
Dx	11	7	0	5
Dy	6	2	4	0

en x	u	w	x	y
Dw	5	0	6	2
Dx	11	7	0	5
Dy	6	2	4	0

en y	u	w	x	y
Dw	5	0	6	2
Dx	11	7	0	5
Dy	6	2	5	0

en w	u	w	x	y
Dw	5	0	7	2
Dx	11	7	0	5
Dy	6	2	5	0

en x	u	w	x	y
Dw	5	0	6	2
Dx	11	7	0	5
Dy	6	2	5	0

en y	u	w	x	y
Dw	5	0	6	2
Dx	11	7	0	5
Dy	6	2	5	0

en w	u	w	x	y
Dw	5	0	7	2
Dx	11	7	0	5
Dy	6	2	5	0

en x	u	w	x	y
Dw	5	0	7	2
Dx	11	7	0	5
Dy	6	2	5	0

en y	u	w	x	y
Dw	5	0	7	2
Dx	11	7	0	5
Dy	6	2	5	0

c)

Si c(x,y) baja a 4, el costo del camino más corto de x a y no cambia por lo que el costo a cualquier otro nodo tampoco cambia. Su vector de distancias no cambia por lo que no informa nada a sus vecinos.