

Redes de Computadoras
Solución – 17 de diciembre de 2018
(ref: serc20181217.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)

- a) Identifique los componentes del retardo en las redes de paquetes, y señale cuáles son los más significativos.
- b) Cuando existe congestión en la red, ¿cual de estos componentes se ve mas afectado?.

Solución Pregunta 1

- a)**
**Retardo de procesamiento en el nodo.*
Es el tiempo que tarda el nodo en procesar el mensaje. Por ejemplo chequeo de paridad (CRC), determinar enlace de salida (enrutamiento).
**Retardo de cola*
Tiempo de espera en colas del enlace de salida para transmisión.
**Retardo de transmisión.*
Es el tiempo que tarda un paquete en ser transmitido por la interfaz del nodo. Sea R =ancho de banda del enlace (bps) y L =longitud del paquete (bits), el tiempo de envío $=L/R$.
**Retardo de propagación.*
Es el tiempo que tarda un paquete en viajar por el medio desde que sale de la interfaz de origen hasta que llega a la interfaz destino. Sea d = longitud del enlace físico y s = velocidad de propagación en el medio, el retardo de propagación $= d/s$.

En diversos escenarios los retardos más significativos pueden variar; en una LAN, con cables cortos y velocidades de transmisión superiores a 100Mbps, los retardos de transmisión y propagación serán típicamente despreciables. En enlaces intercontinentales, por ejemplo satelitales, el retardo de propagación puede ser muy importante, en el orden de cientos de milisegundos, y si la velocidad de transmisión (ancho de banda) es baja, del orden de 64 Kbps, el retardo de transmisión puede ser influyente. Normalmente el retardo de procesamiento es despreciable, y el retardo de cola es variable y depende del tráfico de la red.

- b)**
El retardo de encolamiento depende del nivel de congestión del router ya que si se reciben mas datagramas por segundo de los que se logra reenviar, la cola de espera crecerá y más tiempo deberá esperar un datagrama en ella hasta que sea reenviado.

Pregunta 2 (8 puntos)

Explique que función cumple y como funciona el protocolo Address Resolution Protocol (ARP).

Solución Pregunta 2

El protocolo ARP Address Resolution Protocol, permite determinar la dirección MAC del nodo que tiene configurada cierta dirección IP.
Cada nodo en la red (ya sea un host o un router) presenta una tabla ARP. Ésta tabla contiene registros con los siguientes campos, IP address, MAC address y TTL (Time to live), para los nodos de su LAN.
< IP address; MAC address; TTL >

Redes de Computadoras

El TTL indica después de que tiempo es eliminado el registro.

Ejemplo de funcionamiento:

- Cuando el equipo A desea enviar un datagrama al equipo B, y la dirección MAC de B no se encuentra en la tabla ARP, debe realizar los siguientes pasos
- Envía un ARP query conteniendo la dirección de B, y su dirección IP y MAC. Lo envía broadcast con la dirección FF:FF:FF:FF:FF:FF, en el encabezado de ethernet. Este mensaje es recibido por todos los nodos de la LAN.
- Cuando B recibe el paquete ARP query, lo contesta con un ARP reply al equipo A con su dirección MAC. Este mensaje es enviado en forma unicast
- El equipo A graba el par IP.MAC en la tabla ARP hasta que pase el TTL

Pregunta 3 (8 puntos)

- Mencione y describa el protocolo de acceso múltiple que utiliza Ethernet.
- ¿Qué algoritmo se ejecuta ante una colisión? Explique su funcionamiento.

Solución Pregunta 3

a) Ethernet utiliza el protocolo de acceso múltiple CSMA/CD (CSMA con detección de colisiones). La familia de protocolos CSMA/CD siguen dos reglas fundamentales:

Sondeo de portadora: cada nodo escucha el canal antes de transmitir. Si actualmente se está transmitiendo una trama de otro nodo por el canal, el nodo esperará un intervalo de tiempo aleatorio y luego volverá a sondear para ver si existe portadora en el canal. Si comprueba que el canal está inactivo, el nodo comenzará a transmitir su trama. En caso contrario, el nodo esperará otro intervalo aleatorio de tiempo y volverá a repetir este proceso.

Detección de colisiones: un nodo que esté transmitiendo escuchará qué es lo que hay en el canal mientras dure la transmisión. Si detecta que otro nodo está transmitiendo una trama que interfiere la suya, dejará de transmitir y empleará algún tipo de protocolo para determinar cuándo debe volver a intentar transmitir de nuevo.

b) Se utiliza el algoritmo de espera exponencial (exponential backoff).

Después de abortar la transmisión de una trama debido a una colisión el nodo transmisor entra en la fase de espera exponencial (exponential backoff). Específicamente, a la hora de transmitir una trama que ya ha experimentado n colisiones, el nodo selecciona un valor aleatorio para K del conjunto $\{0, 1, 2, \dots, 2^m - 1\}$, donde $m = \min(n, 10)$. El nodo espera entonces $K * 512$ periodos de bit (K veces el tiempo necesario para transmitir 512 bits) antes de intentar una nueva transmisión.

Pregunta 4 (8 puntos)

- Describa las características de las arquitecturas de Capa de Aplicación Cliente-Servidor y Peer-to-Peer.
- Mencione un ejemplo de una arquitectura híbrida indicando las características de cada capa que posee.

Solución Pregunta 4

a)

Cliente-Servidor

En estas arquitecturas, típicamente las aplicaciones tienen dos componentes fundamentales: el cliente y el servidor.

Los requerimientos de los clientes (hosts), a los que no se les requiere ser "always-on", son servidos por servidores, estos sí "always-on"

Cliente: típicamente, quien inicia el contacto con el servidor, requiriendo servicios ofrecidos por él.

Servidor: quien provee el servicio demandado por el cliente.

En general, no hay comunicación directa entre clientes.

Ejemplo1 DNS (Servicio de resolución de nombres)

Ejemplo2 Web

Peer-to-Peer

En este tipo de arquitecturas hay un uso mínimo (o nulo) de servidores dedicados.

Todos los peers ofician de clientes y también de servidores, lo que le otorga a este tipo de arquitectura la propiedad de ser altamente escalable.

Los peers "aparecen" y "desaparecen" de la red, por lo que, desde el rol de servidores, no cumplen con la propiedad de "always-on", en comparación con el caso de la arquitectura Cliente-Servidor.

Redes de Computadoras

Estas arquitecturas aprovechan la capacidad de procesamiento y almacenamiento de los peers, así como el ancho de banda de sus conexiones a Internet, liberando de dichos requerimientos a los “servidores tradicionales”. Además de diferenciarse de la arquitectura Cliente-Servidor en cuanto al paradigma de funcionamiento, también se diferencia en cómo escalan, ya que en este caso la capacidad de crecimiento está autocontenida en los peers que participan de la misma. En el caso de la arquitectura Cliente-Servidor, la demanda de crecimiento se concentra del lado del Servidor.

Ejemplo: Bit Torrent (transferencias de archivos).

b)

La arquitectura híbrida presenta un modelo con una arquitectura donde un nodo, puede realizar una consulta a un servidor para saber donde están los otros nodos en la red. Una vez hecha la consulta, el nodo podrá establecer la conexión directa con otro nodo para compartir su información. La aplicación P2P debe informar a este servidor, de su conexión y desconexión para mantener la integridad del servicio.

Pregunta 5 (8 puntos)

- a) Explique el proceso de multiplexación y demultiplexación realizado en la capa de transporte.
- b) Para el caso de TCP y UDP indique que información utiliza para realizar la tarea.

Solución Pregunta 4

a)

Proceso de multiplexación – Refiere en la capa 4 a la unión de datos de múltiples aplicaciones (sockets) que envían, encapsulando cada dato con su respectivo cabezal en un segmento, y enviándolos todos juntos a la capa de red.

Proceso de demultiplexing – Refiere a la entrega de cada segmento en el receptor a la correcta aplicación (socket) que debe recibirlos.

b)

En el caso de UDP se utiliza únicamente el puerto destino para decidir a que socket se deben enviar los datos del datagrama.

En el caso de TCP se utiliza la cuaterna <ip origen, puerto origen, ip destino, puerto destino> que identifica a cada conexión.

Es decir que datos con igual ip destino y puerto destino pueden ser entregados a sockets distintos si la ip origen o el puerto origen son distintos.

Problema 2 (30 puntos)

Se desea implementar un programa que permita descargar una página web desde un servidor Web que soporta el protocolo HTTP1.0

El programa será invocado como `wget URL PORT`, por ejemplo:

```
wget example.com/index.html 80
```

Al ejecutarse el programa escribirá a disco el archivo html y todas las imágenes (que usen el tag ``) que esta página contenga. Todas las imágenes deberán descargarse en paralelo. No hay limitaciones arbitrarias sobre el tamaño de los objetos que se descargan.

Se pide:

- a) Implemente en un lenguaje de alto nivel el programa descrito. Dispone de la API de sockets del curso, estructuras de datos estándar, funciones para parsear strings, etc. También dispone de una función `list=find_images(filename)` que recibe el nombre de un archivo en formato html y devuelve una lista de urls de imágenes que aparecen en tags ``.
- b) Explique las modificaciones necesarias para que su programa soporte el protocolo HTTP1.1.

Solución Problema 2

```

a)
-- función auxiliar parsear urls
function parseurl(url)
    -- separamos el hostname, puerto (opcional), y nombre de archivo
    -- ejemplo: "http://(host)[:port?](path)(filename)"
    return host, port, path, filename
end

-- descarga un objeto usando http y lo escribe en un archivo
function download_object(url)
    host, port, path, filename = parseurl(url)
    port = port or arg[2] or 80 --si url no tenia puerto, usamos argumento
    m = socket.tcp()
    s = m.connect(host, port)
    file = io.open(filename, "w") --archivo para escritura

    -- enviamos el get en modo bloqueante
    s.settimeout(-1)
    get = "GET "..path..filename.." HTTP/1.0\r\nHost: "..host..":"..port.." \r\n"
    s.send(get)

    -- leemos el objeto en modo no bloqueante (puede ser muy grande)
    s.settimeout(0)
    headerskipped = false -- bandera para saltar el cabezal http
    repeat
        data, err = s.receive()
        if headerskipped == false then
            -- al principio esperamos saltar el cabezal
            pos = string.find(data, "\r\n\r\n") --final del cabezal http
            if pos then --si encontramos
                data = string.sub(data, pos+2) --después del cabezal es objeto
                file.write(data)
                headerskipped = true
            end
        else
            file.write(data)
        end
    until err == "closed"
    file.close()
end

-- descargamos la página html
download_object( arg[1] )

-- iniciamos las descargas de imágenes en paralelo
_, _, _, htmlfilename = parseurl(url)
images = find_images_in file( htmlfilename )
for image in images do
    thread.new( download_object, image )
end

```

b)
Debido a que en HTTP1.1, las conexiones TCP se mantienen abiertas, se debe cambiar la condición por la que se termina de recibir el mensaje HTTP. Por lo tanto, mientras se lee HTTP deberá parsearse el cabezal buscando el campo "Content-length". Si este se encuentra, deberá tenerse en cuenta durante la lectura del objeto como indicador de que se descargó completamente (además del control de cierre de conexión). Además, se debe modificar el pedido GET para que indique la versión 1.1.