

Redes de computadoras, 5 de enero de 2025

SIMULACIÓN DE PROTOCOLOS CON NS-2

Martínez Buenrostro Jorge Rafael

correo, molap96@gmail.com

Universidad Autónoma Metropolitana
Unidad Iztapalapa, México

Cuestionario

Describa qué hace cada script ejecutado en la práctica, ilustrando las partes en donde se configura cada una de las capas del Modelo TCP/IP.

Script 1

Ya que este script no incluye nodos, enlaces, o tráfico de paquetes, vamos a describir cada parte del script.

Crea una instancia del simulador NS-2 y la asigna a la variable ns. Este objeto es el núcleo de la simulación, permitiendo configurar nodos, tráfico, y otros elementos.

```
set ns [new Simulator]
```

Crea un archivo **out.nam** en modo escritura. Este archivo registrará todos los eventos de simulación, como el envío/recepción de paquetes y otros eventos relevantes

```
set nf [open out.nam w]
```

Le indica al simulador que registre todos los eventos de simulación en el archivo **out.nam** previamente creado.

```
$ns namtrace-all $nf
```

Define una función llamada **finish** que:

global ns nf : Declara las variables **ns** y **nf** como globales para usarlas dentro de la función.

\$ns flush-trace : Asegura que toda la información de traza se escriba en el archivo antes de cerrarlo.

close \$nf : Cierra el archivo de traza **out.nam**

exec nam out.nam & : Ejecuta NAM y abre el archivo **out.nam** para visualizar la simulación.

exit 0 : Finaliza la ejecución del script de manera exitosa.

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    exec nam out.nam &  
    exit 0  
}
```

Programa la ejecución del procedimiento **finish** después de 5 segundos de tiempo simulado. Esto significa, que independientemente de la configuración adicional, la simulación terminará a los 5 segundos.

```
$ns at 5.0 "finish"
```

Inicia la ejecución de la simulación. A partir de este punto, todos los eventos configurados en el simulador se procesan secuencialmente.

```
$ns run
```

Script 2

Como vimos en el punto anterior las siguientes líneas del script son parte de la configuración general y la creación del procedimiento de finalización.

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}
```

Crea dos nodos (*n0* y *n1*) que actuarán como origen y destino en la simulación. Esta parte del script corresponde a la **Capa de Enlace de Datos**, ya que los nodos representan interfaces de red.

```
set n0 [$ns node]
set n1 [$ns node]
```

Crea un enlace dúplex entre los nodos **n0** y **n1** con:

Ancho de banda : 1 Mbps

Retardo : 10 ms

Cola : DropTail (cola FIFO)

La relación con el modelo TCP/IP es con la **Capa de Enlace de Datos** ya que representa el enlace físico y la cola de paquetes. Así como la **Capa de Red** ya que el retardo y el ancho de banda afectan la entrega de paquetes.

```
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

Crea un agente UDP (*udp0*) y lo asocia al nodo **n0**. Se relaciona con el modelo TCP/IP ya que representa la **Capa de Transporte**, ya que el protocolo UDP es un protocolo en esta capa.

```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
```

Crea una fuente de tráfico CBR (*cbr0*) con:

Tamaño de paquete : 1000 bytes

Intervalo entre paquetes : 0.005 segundos

Se conecta al agente UDP *udp0*

Representa la **Capa de Aplicación**, ya que genera tráfico constante para enviar a través de la red.

```
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 1000
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
```

Crea un agente Null (*null0*) que actúa como receptor de tráfico en el nodo **n1**. Representa la **Capa de Transporte**, ya que recibe y descarta paquetes sin procesarlos.

```
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
```

Conecta el agente UDP (*udp0*) en el nodo **n0** con el agente Null (*null0*) en el nodo **n1**. Relaciona las **Capas de Transporte y Red**, al definir una ruta lógica entre el origen y el destino.

```
$ns connect $udp0 $null0
```

Programa el inicio del tráfico CBR a los 0.5 segundos y su detención a los 4.5 segundos de tiempo de simulación. Impacta la **Capa de Aplicación**, ya que controla cuándo se envían los datos.

```
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

1. Capa de Enlace de Datos

- Creación de nodos.
- Configuración del enlace dúplex.

2. Capa de Red

- Configuración del enlace afecta el enrutamiento y entrega de paquetes.

3. Capa de Transporte

- Configuración del agente UDP y receptor Null.

4. Capa de Aplicación

- Generación de tráfico constante (*CBR*).
- Programación del inicio y finalización del tráfico.

Script 3

¿Cómo podría usarse NS-2 para explicar la relación entre el retardo de propagación y el de transmisión en un enlace? ¿Qué parte de un script TCL debería modificarse para ejemplificar esto?

Para el segundo ejemplo de la práctica, modifique el código para que el tamaño del paquete sea igual a 1,500 bytes. ¿Qué diferencias observa en la animación con NAM?

Investigue la disponibilidad de simuladores de red y realice una tabla con al menos cinco simuladores que compare sus características básicas.