Caso 0

El objetivo de esta práctica es modelizar el comportamiento de un firewall real, mediante la teoría de colas pudiendo analizarlo mediante las ecuaciones correspondientes e implementarlo mediante una herramienta de simulación. Para la parte de ecuaciones se utiliza Mathematica, mientras que la simulación se realiza mediante OMNeT++. Para ello nos hemos basado en el paper “Performance Modeling and Analysis of Network Firewalls” (2012) de Khaled Salah, Khalid Elbadawi y Raouf Boutaba.

Los firewalls son equipos de red en los que se configuran una serie de reglas. Cada una de estas reglas incluye unas condiciones y una acción. Cuando un paquete llega al firewall, se compara secuencialmente con las reglas y cuando cumple todas las condiciones de una regla, se ejecuta la acción de dicha regla y se detiene el procesamiento. Por tanto, una consideración importante será la posición de la regla que haga el “match”, ya que cuanto más abajo este, mayor será el tiempo medio de estancia en el sistema de los paquetes.

Para modelizarlo mediante la teoría de colas, el firewall se considera como una cola formada por un número K de servidores en serie, que representan las K reglas del firewall, seguido de un último servidor que representa el procesado final del paquete, en el que se ejecuta la acción. En nuestro caso vamos a modelizar un único flujo de tráfico, entendiendo como tal una secuencia de paquetes que hacen “match” en la misma regla, que será la regla M-esima. Este experimento es interesante, ya que nos permite conocer el rendimiento del firewall en caso de ataque DoS o DDoS. En ese caso el/los atacantes tratarían de lanzar una gran cantidad de tráfico apuntando a la regla más abajo posible.

A continuación, se muestran brevemente los cálculos programados y la modelización:

# Mathematica:

Con esta herramienta se ha implementado un notebook que realiza varios cálculos. En primer lugar, partiendo de las ecuaciones de balance para cada estado, se ha desarrollado un algoritmo que calcula las N\*K +1 probabilidades de estado de un firewall de flujo único en función de lambda, mu y r siendo:

Lambda (tasa de llegada de los paquetes al firewall)

Mu (tasa de servicio del paquete al enlace)

r (tasa de comprobación de reglas)

N(numero de servicios aplicados a cada paquete) y una cola de paquetes

K(tamaño de la cola de entrada al firewall)

Seguidamente se obtienen los siguiente datos estadísticos de caracterización matemática del comportamiento del firewall, incluidos los ya mencionados:

p0(probabilidad de no haber paquetes en el firewall)

P[K,N](probabilidad de estado para cada regla y paquetes en sistema)

X(tiempo medio de servicio)

gamma(throughput para el flujo lambda 1)

Ploss(probabilidad de pérdida de paquetes en el firewall)

KA(numero medio de paquetes en el sistema)

W(tiempo medio en el sistema)

Wq(tiempo medio de espera)

Uutil(utilización de la CPU del firewall)

NOTA: Hay datos como el thoughput y la pérdida de paquetes que pueden ser calculados de dos maneras. Se han implementado ambas maneras. El código es bastante explicativo de lo que va sucediendo en cada momento.

# OMNeT++:

La red diseñada es la siguiente:

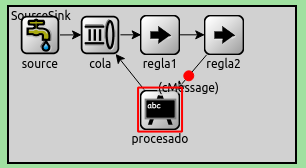


Figura 1: Red de OMNeT++

En ella tenemos un nodo “Source”, que genera mensajes con una tasa de llegadas lambda, representando los paquetes de entrada el firewall. Estos paquetes son encolados en un nodo llamado “StopWait”, que es una cola finita en el que cada vez que llega un paquete se añade a la cola y si esta está llena, lo descarta. Cuando recibe el aviso de que se ha terminado de procesar el paquete anterior, saca el siguiente paquete en la cola.

Los paquetes salientes van a una serie de nodos llamados “ReglaN”, que representan cada una de las reglas existentes en el firewall, que se ejecutan secuencialmente. El comportamiento de las reglas es sencillo, cuando reciben un paquete lo procesan y lo pasan a la siguiente regla. En nuestra simulación, este procesado se modela como un tiempo de espera aleatorio, en base a una tasa de servicio lambda. Para la simulación del caso 0, se ha supuesto que la regla que hace el “match” es la regla M=2, con el objetivo de simplificarlo. En caso de que la regla matcheada fuera otra, podríamos añadir más nodos “Regla” o modificar el comportamiento de “Regla” de forma que pongamos un único nodo que espere M-veces un tiempo aleatorio de distribución exponencial y media 1/lambda.

Finalmente, el paquete es enviado al nodo “Procesado”, que representa el procesamiento del paquete en función de la acción asociada a la regla que ha hecho el “match”. Este procesado se ha considerado determinista, para diferenciarlo de las reglas, aunque como el tamaño de los paquetes en un entorno real sería variable, podría considerarse aleatorio. Cuando el nodo “Procesado” termine de procesar el paquete, enviará un mensaje a la cola “StopWait”, que a modo de ACK, permita la entrada al sistema del siguiente paquete.