16 de Setiembre 2019
Universidad de Costa Rica
Maestría en Computación
Tarea Programada 2
José Quesada

Muestreo de tráfico en la red

Se realizó la lectura del artículo Caracterización de tráfico encriptado y por VPN usando *features* relacionadas con tiempo por Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun y Ali A. Ghorbani. Basado en este artículo se buscó replicar un dataset con características similares a las mencionadas en el miso. La idea principal es construir un sistema que nos permita extraer datos similares a los del artículo para que puedan ser clasificadas con un clasificador como el que se menciona o el que se construyó en la Tarea Programada 1. Para este proceso se utilizó SDN y los beneficios que ofrece OpenFlow para poder extraer estadísticas de los flujos con los cuáles podemos sacar las métricas necesarias para construir tuplas similares a las que se busca obtener.

Las herramientas utilizadas para la construcción de este sistema fueron pox como controlador de nuestra red de SDN, además se usó mininet para la creación de la red y las pruebas de la misma. Se utilizó el l2 learning switch que se encuentra como ejemplo en el repositorio de pox de github, con la leve modificación que se utilizó idle timeout en 0 y hard timeout en 0, esto con la principal razón de que los flows no se venzan. Para extraer las estadísticas se utilizó el método de _handle_FlowStatsReceived que ofrece pox en su core de funciones al suscribirse a los eventos del controlador. La idea de utilizar este método es que nos permite recibir estadísticas de los flujos cada cierto tiempo con el fin de que utilicemos esta información para armar los flujos (tuplas) que queremos.

A partir de los Flows que se obtienen de los FlowsStats de pox, buscamos construir una clase Flow en la que llevamos un contador de la cantidad de paquetes y bytes recibidos en un momento determinado, para cada flow, que se encuentra caracterizado por ip destino, ip fuente, puerto destino y puerto fuente. Además, buscamos asociar en otra clase llamada FullFlow, los dos Flows (clase definida anteriormente), el de fuente a destino y el de destino a fuente, como se observa en la Figura 1, esto con el fin de tener en una sola clase la información relevante a ambas direcciones de la transmisión de datos.

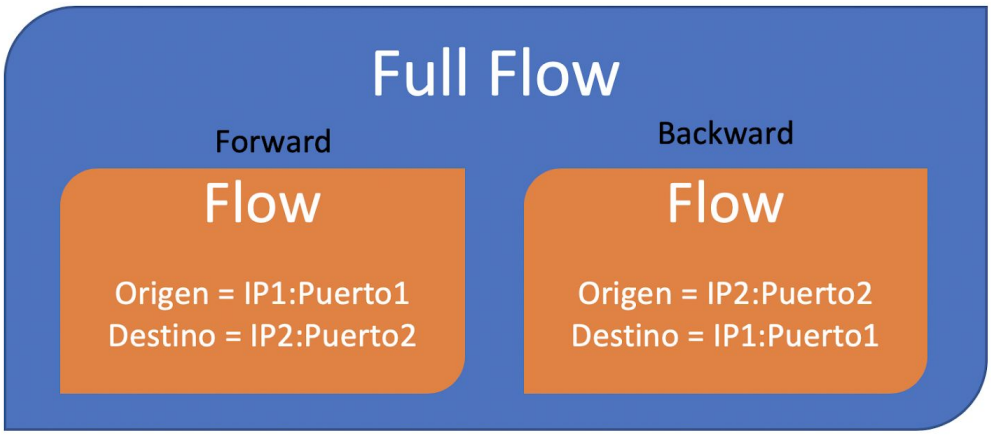

Figura 1. Descripción de FullFlow.

A partir de toda la información obtenida dentro de FullFlow que contiene la información de ambos Flows, uno por cada dirección, es que se pudieron construir las tuplas con las características descritas en el artículo estudiado. Para calcular los fb_psec o bytes del flow por segundo, se sumaron todos los bytes obtenidos en ambas direcciones y se dividió entre la duración del flow. Para los fp_psec o paquetes del flow por segundo se sumaron todos los paquetes obtenidos en ambas direcciones y se dividió entre la duración del flow. Para las estadísticas de fiat o tiempo de arribo entre paquetes en dirección fuente-destino, se utilizaron todas aquellas mediciones de una dirección del Flow en las que la cantidad de paquetes era mayor a 1 y basado en eso se divide la cantidad de paquetes entre la duración desde la última lectura para calcular el fiat de este momento. Con estas mediciones de fiat luego se calculan las medias, máximos, mínimos y desviación estándar. Para biat o tiempo de arribo entre paquetes en dirección destino-fuente, se realiza lo mismo que fiat, pero se utilizan las mediciones del Flow en dirección opuesta. Para flowiat o tiempo de arribo entre paquetes del flow, se unen las mediciones obtenidas entre fiat y biat y luego se realiza el mismo proceso. Para calcular el tiempo que el flow estuvo activo se suma la duración de todas aquellas mediciones que tuvieran paquetes mayor a 1 y para calcular el tiempo que el flow estuvo *idle* o inactivo, se suma la duración de todas aquellas mediciones que tuvieran paquetes igual a 0. Al obtener todas estas medidas se cumple con la obtención de las mismas columnas que fueron descritas en el artículo.

Además, se fue más allá que sólo la obtención de estadísticas para los Flows, se implementó la posibilidad de bloquear un flujo de datos cuando excede cierta cantidad de bytes. Para esta parte se utilizó un bloqueo de tráfico utilizando pox, en el cual se cambian las reglas haciendo un flowmod con acciones vacías. Para esto se utiliza la

cantidad de bytes y si esta es igual al número definido por MAX_BYTES se bloquea el tráfico haciendo un cambio sobre las reglas de flow mod utilizando una modificación para quitarle la acción de packet out.