

Progetto Gestione di Rete

Analisi volumetrica del traffico e visualizzazione
tramite bubblegraph

Andrea Bonanno



Dipartimento di Informatica
Università degli Studi di Pisa
8 Settembre 2019

1 Introduzione

Il progetto consiste nella realizzazione di un esempio di cattura del traffico di rete, di parsing dei pacchetti e dell'estrazione di alcune metriche rilevanti (volume complessivo dei payload inviati e ricevuti da ogni coppia di hosts e il numero di pacchetti scambiati), e di una presentazione delle stesse tramite grafici a bolla. Il progetto fa uso della libreria *lpcap* per la cattura del traffico di rete e della libreria *chart.js* per il plot del grafico a bolla.

2 Implementazione

Una piccola collezione di file sorgente in linguaggio C realizza la cattura del traffico, il parsing e la gestione in memoria delle metriche raccolte e infine la generazione di un report *report.js* contenente un riepilogo in formato *JSON* dei dati da visualizzare. Nello specifico *sniffer.c* implementa tutte le chiamate alla libreria *lpcap* necessarie ad avviare la cattura, che può avvenire in modalità *live* o *offline*, in base al fatto che l'utente fornisca un'interfaccia di rete oppure un file di cattura *.pcap* al momento dell'esecuzione. Il sorgente *parser.c* realizza le funzioni di parsing dei pacchetti, leggendo i campi opportuni e calcolando gli offset degli stessi in base al protocollo dello stesso (*TCP*, *UDP*, *ICMP*). Il sorgente *hash.c* implementa una hash table per la memorizzazione dei dati e le funzionalità di serializzazione in formato *JSON* a cattura ultimata. Infine la lettura di *report.js* e il plot del grafico all'interno di una pagina html *graph.html* sono realizzate in linguaggio JavaScript in *charter.js*.

3 Rappresentazione dei dati

Ogni elemento del grafico a bolla è capace di rappresentare quattro dimensioni di dato, codificate rispettivamente nelle coordinate sull'asse delle ascisse e ordinate, raggio della bolla e colore della stessa. Si è scelto di utilizzare la posizione sugli assi cartesiani della bolla per identificare gli indirizzi IP degli hosts a cui appartiene il traffico, il raggio per indicare il totale payload dello stesso e il colore per disambiguare il protocollo a cui il traffico appartiene. Nello specifico l'asse delle ascisse etichettato come *Host(TX)* rappresenta

l'host mittente e l'asse delle ordinate etichettato come *Host(RX)* rappresenta l'host destinatario. Durante l'implementazione si è scelto di rappresentare il raggio dei cerchi come radice quadrata del numero di bytes, in quanto l'area è una quantità proporzionale in maniera quadratica al raggio di un cerchio. Si effettua inoltre una normalizzazione dei raggi delle bolle rispetto al payload massimo relativo alla cattura da rappresentare, in modo che nessuna bolla domini completamente il grafico. Si è infine fatto uso delle funzionalità di personalizzazione della libreria *chart.js* per modificare i valori rappresentati dagli assi e per la creazione di custom tooltips per riportare informazioni più dettagliate, ad esempio il numero di pacchetti rappresentati da una bolla e il loro esatto ammontare in byte o KiB. La sovrapposizione di più bolle sullo stesso punto del grafico (per esempio nel caso di una coppia di hosts che ha generato traffico usando più protocolli) è stata mediata dalla trasparenza delle bolle e dalla possibilità di filtrare gli elementi visualizzati sul grafico in base al protocollo di appartenenza (click sugli elementi della legenda).

4 Utilizzo

I prerequisiti per la compilazione sono il package *libpcap-dev*, il tool *automake* e lo GNU C compiler (*gcc*). L'esecuzione del *makefile* tramite il comando *make* genera l'eseguibile *sniffer*, la cui esecuzione supporta le seguenti opzioni:

- *-d devicename* Device mode. L'utente fornisce un nome di interfaccia di rete valido per la cattura del traffico. Nel caso questa opzione non venga utilizzata, *lpcap* seleziona autonomamente la prima interfaccia valida.
- *-o filename* Offline mode. L'utente fornisce un file *.pcap* per la cattura del traffico in modalità offline. Ignorata se si utilizza la Device mode.
- *-n number* L'utente specifica il numero esatto di pacchetti da catturare (default 100).
- *-v* Verbose mode. L'eseguibile stampa su *stdout* in tempo reale un breve report per ogni pacchetto catturato.

Il *makefile* è dotato di un phony target *clean* per la rimozione di *report.js*, dell'eseguibile *sniffer* e di tutti gli object file. Se la cattura

del traffico ha avuto successo, il grafico risultante è contenuto in *graph.html*.

5 Esempio

