Progetto di
Complementi di Gestione di Reti

studente:
Giovanni Avveduto
matr. 270019

Il programma AvvedutoSniffer si occupa di analizzare il traffico di rete, creando un grafico con le statistiche dei protocolli analizzati e throughput; con le opzioni -p e -f stampa sullo stdout le informazioni sui singoli pacchetti, e sui flussi.
Il flusso viene considerato scaduto se per 5 secondi non arrivano pacchetti che appartengono ad esso, o se è cominciato da più di 30 secondi (è possibile modificare questi tempi dalle variabili "flow_timeout" e "flow_lifetime").

```
usage:  AvvedutoSniffer [opts [optargs] ]
        This sniffer needs root privileges
        Options:
        -d deviceName: specify the device name to sniff
        -a: set device in promiscuous mode
        -r rrdname: RRD filename
        -i imagename: RRDgraph image filename
        -c: delete RRD file if exists yet
        -n: threads number
        -v: verbose, print RRD commands
        -p: print captured packet to stdout
        -f: print flow info to stdout
        -u updatetime: RRDupdates every x seconds (default 1)
        -t rrdgraphtime: RRDgraph every x seconds (default 5)
        -w windowsize: RRDgraph window's size in minutes (default 15)
```

Architettura del programma

Il programma è strutturato in modalità multi threading per sfruttare tutti i core disponibili. Un thread "pcap_loop" si occupa di catturare i pacchetti (dal primo device disponibile, o dal device settato con l'opzione "-d dev_name") usando la libreria libpcap, analizza gli header dei pacchetti (fino agli header TCP, UDP, e ICMP nel caso di pacchetti IP e IPv6; riconosce anche pacchetti ARP), se richiesto stampa le informazioni del pacchetto; se l'analisi dei flussi è attivata inserisce una struttura dati contenenti le informazioni raccolte sul pacchetto in una coda. Se il programma è fatto partire con la modalità multi threading "-n th_num" vengono create 'th_num-1' code e 'th_num-1' thread "flow_analyzer"; in questo caso ciascun pacchetto viene inserito in una delle code a seconda di una funzione hash determinata sulla quintupla "ip_src, port_src, ip_dst, port_dst, protocollo" che caratterizza ciascun flusso; in questo modo tutti i pacchetti appartenenti ad un flusso vengono inseriti sempre nella stessa coda. Ogni coda è strutturata in modo da non esserci il

bisogno di sincronizzazione tra il singolo scrittore e il singolo lettore. In una tabella Hash vengono memorizzate tutte le informazioni relative ai flussi. Ogni thread "flow analyzer" prende dalla propria coda le informazioni sui pacchetti, e aggiorna le informazioni sui flussi contenute nella tabella hash. Ogni thread "flow_analyzer" accede alla tabella hash a seconda dell'hash in modo tale che in ogni locazione della tabella lavora un solo thread, rendendo non necessario l'utilizzo di semafori. Il thread "main" si occupa di aggiornare il database RRD e di disegnare i grafici ad intervalli di tempo impostabili come opzioni. L'ultimo thread "flow_printer" si occupa di eliminare i flussi scaduti dalla hash table e stamparli sullo stdout. Tutte le strutture dati e i thread necessari all'analisi dei flussi non vengono create se l'analisi dei flussi non è ablitata.