Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Počítačové a komunikačné siete

Analyzátor sieťovej komunikácie

Tibor Dulovec

Meno cvičiaceho: Ing. Miroslav Bahleda, PhD.
Čas cvičení: Štvrtok 18:00
Dátum vytvorenia: 19. 10. 2021

Obsah

Zadanie úlohy

Cieľom práce je navrhnutie a implementovanie programového analyzátora Ethernet siete, ktorý analyzuje komunikácie v sieti zaznamenané v načítanom .pcap súbore a poskytuje nasledujúce informácie o komunikáciách.

- Výpis všetkých rámcov v hexadecimálnom tvare postupne tak, ako boli zaznamenané v súbore.
    - Poradové číslo rámca v analyzovanom súbore.
    - Dĺžku rámca v bajtoch poskytnutú pcap API, ako aj dĺžku tohto rámca prenášaného po médiu
    - Typ rámca – Ethernet II, IEEE 802.3 (IEEE 802.3 s LLC, IEEE 802.3 s LLC a SNAP, IEEE 802.3 – Raw)
    - Zdrojovú a cieľovú fyzickú (MAC) adresu uzlov, medzi ktorými je rámec prenášaný.
    - Vo výpise jednotlivé bajty rámca usporiadajte po 16 alebo 32 v jednom riadku.
- Výpis vnorených protokolov pre rámce typu Ethernet a IEEE 802.3
- Analýza všetkých odosielajúcich uzlov a výpis najpoužívanejšieho
- Analýza komunikácií protokolov: HTTP, HTTPS, TELNET, SSH, FTP riadiace, FTP dátové, TFTP, ICMP, ARP

Implementačné prostredie

Program je vytvorený v programovacom jazyku Python vo verzií 3.9. Pre správne fungovanie sa využíva knižnica Scrapy, ktorá slúži iba pre správne načítanie súboru.

Externé súbory

db.txt

Súbor slúži pre preklad číselných hodnôt protokolov do textových. Program sa v ňom orientuje podľa indexov, kde ma hladať.

```
0800 IPv4
0806 ARP
86dd IPv6
88cc LLDP
9000 Loopback
42 STP
aa SNAP
ff RAW
e0 IPX
01 1 ICMP
06 6 TCP
11 17 UDP
0014 20 FTP-DATA
0015 21 FTP-CONTROL
0016 22 SSH
0017 23 TELNET
0050 80 HTTP
01BB 443 HTTPS
0035 53 DNS
0045 69 TFTP
```

Icmp.txt

Súbor slúži pre rozlišovanie a priraďovanie mien k typom a kódom ICMP rámcov

```
00 Echo-reply
03 Destination-Unreachable
04 Source-Quench
05 Redirect
08 Echo
09 Router-Advertisement
10 Router-Selection
11 Time-Exceeded
12 Parameter-Problem
13 Timestamp
14 Timestamp-Reply
15 Information-Request
16 Information-Reply
17 Address Mask-Request
18 Address Mask-Reply
30 Traceroute
# Destination Unreachable
00 Net-Unreachable
01 Host-Unreachable
02 Protocol-Unreachable
03 Port-Unreachable
04 Fragmentation-Needed-n-DF-set
05 Source-Host-Isolated
06 Destination-Network-Unknown
07 Destination-Host-Unknown
08 Source-Host-Isolated
```

Fungovanie programu

Používateľské rozhranie

```
[File 'trace-16.pcap' was loaded]

Actions list:
1 - Show all frames
2 - Show most used IP address
3 - Show all TFTP communications
4 - Show all ARP communications
5 - Show all ICMP communications
6 - Filter by TCP protocol
7 - Find communication by TCP protocol
q - Quit application
---------------------------
Type action >
```

Pri spustení programu, program informuje používateľa o tom, ktorý súbor sa načítal.

Ďalej ma používateľ možnosť vyberať akcie podľa jeho rozhodnutia. Menu akcií a voľba si z neho vyberať je zobrazená vždy po vykonaní niektorej z akcií.

Po vykonaní akcií sa výstup vypíše do konzole. Zároveň sa všetky rámce vypíšu aj do externého súboru „frames-output.txt“. Ten sa premazáva každým novým spustením programu.

1) Prvá možnosť zobrazí všetky rámce a informácie o nich
2) Druhá možnosť zobrazí všetky zdrojové IP adresy a vypíše tu najpoužívanejšiu. Zároveň vypíše aj počet použití.
3) Tretia možnosť vypíše všetky TFTP komunikácie
4) Štvrtá možnosť vypíše všetky ARP komunikácie
5) Piata možnosť vypíše všetky ICMP rámce
6) Šiesta možnosť vypíše všetky TCP protokoly podľa voľby používateľa

```
---------------------------
Type action > 6
Type protocol > ftp-control
===========================
```

7) Siedma možnosť vypíše všetky TCP komunikácie daného protokolu. Používateľské rozhranie sa správa podobne ako pri možnosti 6.
8) Po stalčení klávesy q sa aplikácia ukončí

Načítanie a spracovanie rámcov zo vzorky

Na začiatku programu sa zo vzorky načítajú všetky rámce. Tie sa spracujú vo funkcií *„calc_all_frames()“*

Všetky rámce sú objekty triedy **Frame,** ktoré majú v sebe všetky svoje vlastnosti. Vlastnosti a informácie, ktoré vyplývajú iba z daného rámca si vypočítava objekt sám. Ten iba dostane samotné dáta zo súboru, číslo, ktorý je v poradí a preloženú databázu protokolov z externého súboru db.txt.

Ďalšie vlastnosti a priradenie do komunikácií (ako napríklad TFTP), ktoré vyplývajú aj zo vzťahu k iným rámcom následne vypočítava samotná funkcia *„calc_all_frames()“*.

Jediné komunikácie, ktoré sa počítajú mimo tejto funkcie sú komunikácie podľa TCP protokolu.

Výpis všetkých rámcov

```
def print_frames(frames):
    for frame in frames:
        frame.print_frame()
        output.write(frame.export_to_string() + "\n")
```

Táto funkcia iba prechádza všetky rámce, ktoré dostane ako parameter a vypisuje ich podľa toho, ako to ma rámec definované v triede.

Zároveň táto funkcia robí zálohu do externého súboru.

Funkcia sa využíva aj v iných častiach kódu, stará sa vždy o výpis množiny rámcov.

Výpis najviac použitých zdrojových IP adries

Samotné adresy a počet ich použití sú zozbierané už v *„calc_all_frames()“* funkcií. Táto funkcia ich prehľadáva, vypisuje a zároveň vzájomným porovnávaním hľadá tu najviac používanú.

Potom vypíše tu najpoužívanejšiu

```
Type action > 2
=== List of all used IP address ===
10.20.5.2
10.10.2.1
Most used address 10.10.2.1 with 2183 pakets
================================================
```

Zobrazenie všetkých TFTP komunikácií

Vďaka tomu, že všetky komunikácie sú už vypísané. V kóde stačí jedným cyklom ich všetky prejsť a vypísať podľa potreby

```
def print_tftp():
    print("All TFTP communications")
    print("Count of all: " + str(len(communications_tftp)))
    for index, communication in enumerate(communications_tftp):
        print(f"Communication {str(index + 1)} - {len(communication[0])} frames")
        print(f"Source IP: {communication[0][0].sourceIpAddress}:{communication[0][0].sourcePort}     "
              f"Destination IP: {communication[0][0].destinationIpAddress}:{communication[0][0].destinationPort}")
        print("Frames:")
        print_frames_limits(communication[0])
```

V kóde sa využíva rozšírena funckia výpisu rámcov: *„print_frames_limits“*
Táto funkcia sa stará o to, aby v prípade veľkého počtu rámcov boli vypísané iba prvých a posledných 10.

Zobrazenie ARP komunikácií

Táto funkcia tak isto z predom zozbieraných rámcov vypíše všetky rámce. Zároveň informuje aj o tom, či ma komunikácia pár, zdrojovú MAC adresu, cieľovú MAC adresu, zdrojovú IP adresu a cieľovú IP adresu.

Zobrazenie ICMP rámcov

Funkcia iba prejde všetky rámce a odfiltruje tie, ktoré nie sú ICMP

Zobrazenie podľa TCP protokolu

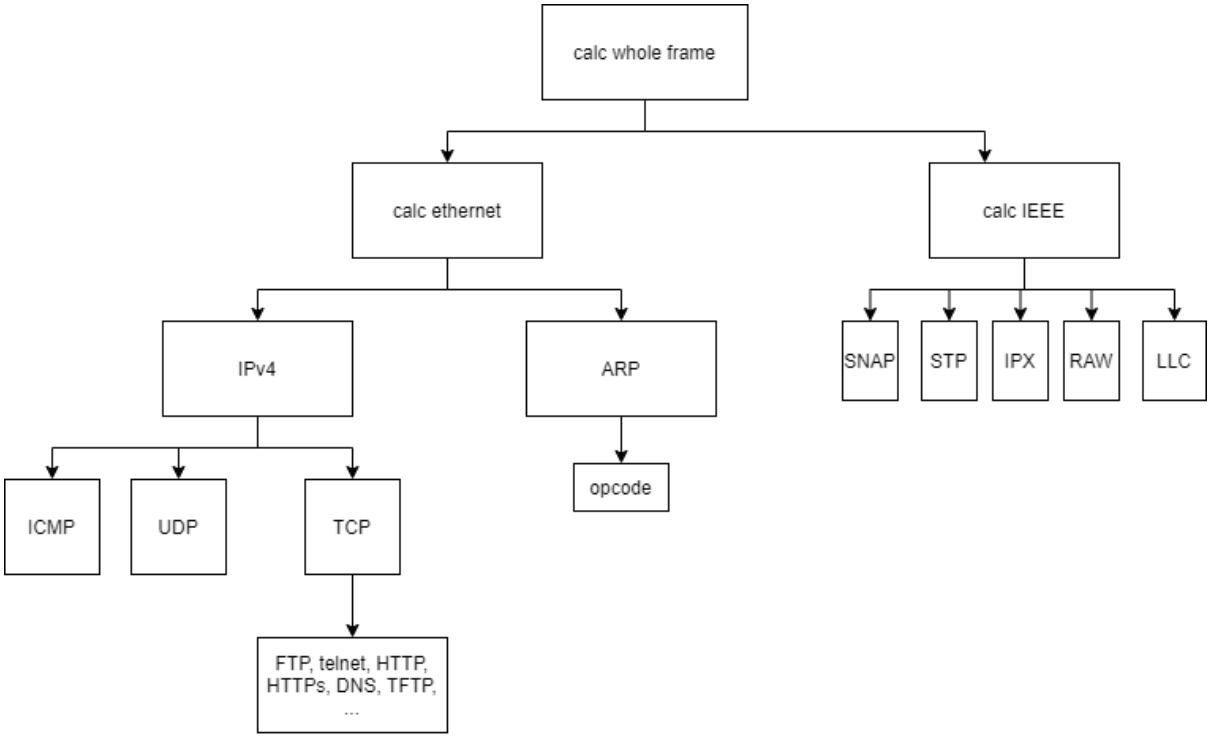
Pre túto funkciu funguje vyhľadávanie všetkých protokolov, ktoré sú v db.txt externom súbore.

Nájdenie komunikácie podľa vybraného TCP protokolu

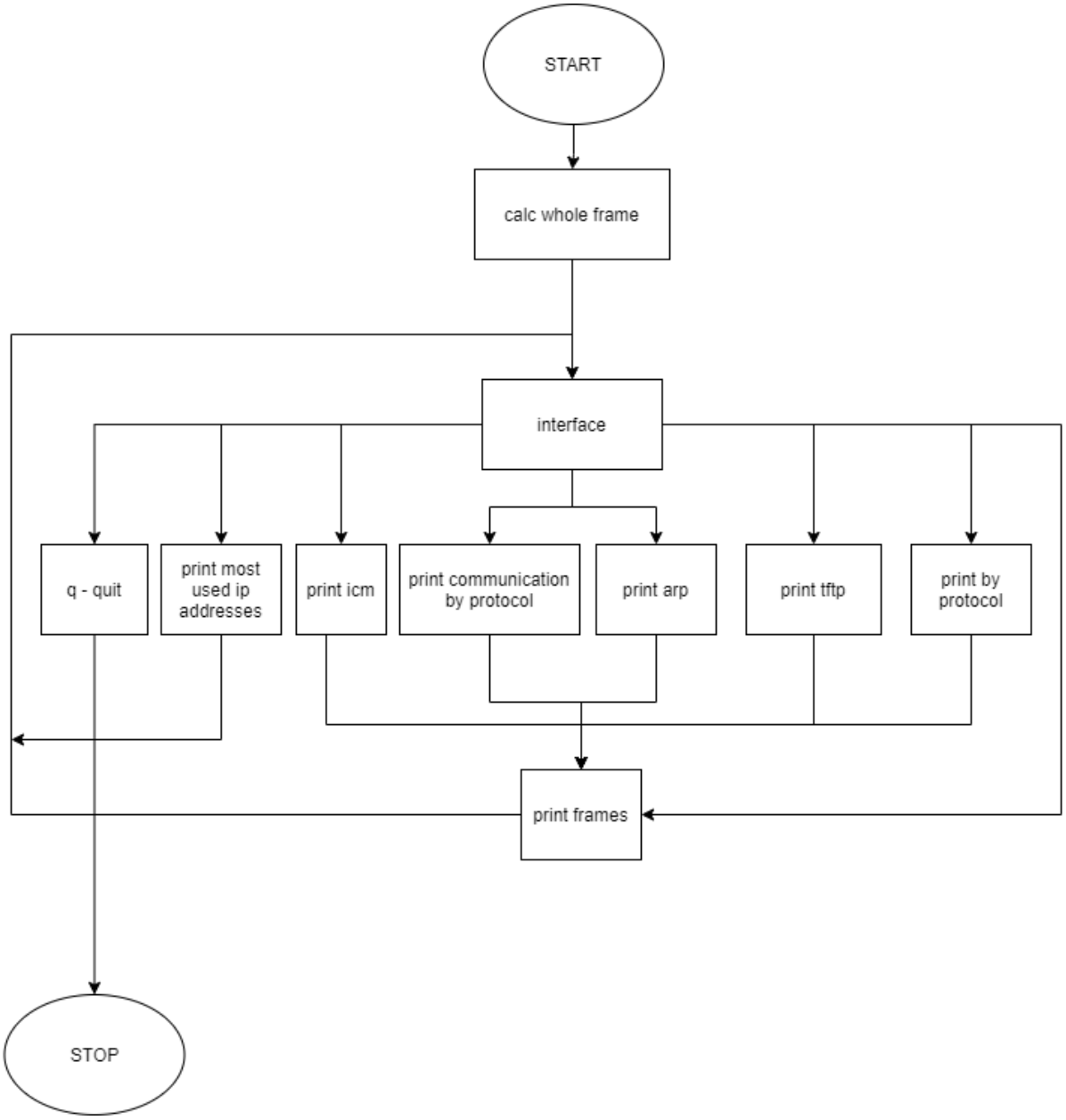
Táto funkcia prehľadáva všetky komunikácie a hľadá značky. Podľa nich sa rozhoduje, či vytvorí novú komunikáciu alebo ju priradí k už vytvorenej.

Každá nová vytvorená komunikácia je od začiatku vnímaná ako uzavretá. Po tom ako sa nájde koniec. (reset alebo FIN z oboch strán), sa označí ako uzatvorenú

Blokový návrh rozhodovania o určovaní rámcu



Blokokový návrh správania programu



Príklady výpisu rámca

```
Frame: 242
PCAP API packet length: 54 B
Real packet length: 64 B
Source MAC address: b4:b5:2f:74:cb:ae
Destination MAC address: 00:02:cf:ab:a2:4c
Ethernet II
 -IPv4
  -Source IP address: 192.168.1.33
  -Destination IP address: 173.252.110.27
 -TCP
  -HTTP
   -RST, ACK
   -Source port: 50018
   -Destination port: 80

00 02 cf ab a2 4c b4 b5  2f 74 cb ae 08 00 45 00
00 28 0f 61 40 00 80 06  00 00 c0 a8 01 21 ad fc
6e 1b c3 62 00 50 90 8d  ce 96 9f f6 d6 10 50 14
00 00 dd fb 00 00
```

```
Frame: 42
PCAP API packet length: 70 B
Real packet length: 74 B
Source MAC address: cc:08:09:d4:00:00
Destination MAC address: 02:00:4c:4f:4f:50
Ethernet II
 -IPv4
  -Source IP address: 12.0.0.1
  -Destination IP address: 12.0.0.5
 -ICMP
  -Destination-Unreachable
   -Port-Unreachable

02 00 4c 4f 4f 50 cc 08  09 d4 00 00 08 00 45 c0
00 38 00 18 00 00 ff 01  a2 e7 0c 00 00 01 0c 00
00 05 03 03 4b 98 00 00  00 00 45 00 02 20 38 09
00 00 7f 11 e9 be 0c 00  00 05 0c 00 00 01 05 e6
c2 fd 02 0c e6 74
```