Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava

Počítačové a komunikačné systémy

Zadanie č.2

Analyzátor sieťovej komunikácie

Vereski Tijana

Cvičiaci: Ing. Marek Galinski
Študijný odbor: Internetové technologie
Ročník: 2. Ing
Akademický rok: 2019/2020
Termín cvičení: pondelok 17:00

OBSAH

1. Znenie zadania:

Navrhnite a implementujte programový analyzátor Ethernet siete, ktorý analyzuje komunikácie v sieti zaznamenané v .pcap súbore a poskytuje nasledujúce informácie o komunikáciách. Vypracované zadanie musí spĺňať nasledujúce body:

1) **Výpis všetkých rámcov v hexadecimálnom tvare** postupne tak, ako boli zaznamenané v súbore.

Pre každý rámec uveďte:
a) Poradové číslo rámca v analyzovanom súbore.

b) Dĺžku rámca v bajtoch poskytnutú pcap API, ako aj dĺžku tohto rámca prenášaného po médiu.

c) Typ rámca – Ethernet II, IEEE 802.3 (IEEE 802.3 s LLC, IEEE 802.3 s LLC a SNAP, IEEE 802.3 – Raw).

d) 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**. Pre prehľadnosť výpisu je vhodné použiť neproporcionálny (monospace) font.

2) Pre rámce typu **Ethernet II a IEEE 802.3 vypíšte vnorený protokol**. Študent musí vedieť vysvetliť, aké informácie sú uvedené v jednotlivých rámcoch Ethernet II, t.j. vnáranie protokolov ako aj ozrejmiť dĺžky týchto rámcov.

3) Analýzu cez vrstvy vykonajte pre rámce Ethernet II a protokoly rodiny TCP/IPv4:

Na konci výpisu z bodu 1) uveďte pre IPv4 pakety:
a) Zoznam IP adries všetkých vysielajúcich uzlov,

b) IP adresu uzla, ktorý sumárne odoslal (bez ohľadu na príjemcu) najväčší počet paketov a koľko paketov odoslal (berte do úvahy iba IPv4 pakety).

IP adresy a počet poslaných paketov sa musia zhodovať s IP adresami vo výpise Wireshark -> Statistics -> IPv4 Statistics -> Source and Destination Addresses.
4) V danom súbore analyzujte komunikácie pre zadané protokoly:

a) HTTP

b) HTTPS

c) TELNET

d) SSH

e) FTP riadiace

f) FTP dátové

g) TFTP, **uveďte všetky rámce komunikácie**, nielen prvý rámec na UDP port 69

h) ICMP, uveďte aj typ ICMP správy (pole Type v hlavičke ICMP), napr. Echo request, Echo reply, Time exceeded, a pod.

i) **Všetky** ARP dvojice (request – reply), uveďte aj IP adresu, ku ktorej sa hľadá MAC (fyzická) adresa a pri ARP-Reply uveďte konkrétny pár - IP adresa a nájdená MAC adresa. V prípade, že bolo poslaných viacero rámcov ARP-Request na rovnakú IP adresu, vypíšte všetky. Ak sú v súbore rámce ARP-Request bez korešpondujúceho ARP-Reply (alebo naopak ARP-Reply bez ARP-Request), vypíšte ich samostatne.

Vo všetkých výpisoch treba uviesť aj IP adresy a pri transportných protokoloch TCP a UDP aj porty komunikujúcich uzlov.
V prípadoch komunikácií so spojením vypíšte iba jednu kompletnú komunikáciu - obsahuje otvorenie (SYN) a ukončenie (FIN na oboch stranách alebo ukončenie FIN a RST alebo ukončenie iba s RST) spojenia a aj prvú nekompletnú komunikáciu, ktorá obsahuje iba otvorenie spojenia. Pri výpisoch vyznačte, ktorá komunikácia je kompletná.
Ak počet rámcov komunikácie niektorého z protokolov z bodu 4 je väčší ako 20, vypíšte iba 10 prvých a 10 posledných rámcov tejto komunikácie. **(Pozor: toto sa nevzťahuje na bod 1, program musí byť schopný vypísať všetky rámce zo súboru podľa bodu 1.)** Pri všetkých výpisoch musí byť poradové číslo rámca zhodné s číslom rámca v analyzovanom súbore.

5) Program musí byť organizovaný tak, aby čísla protokolov v rámci Ethernet II (pole Ethertype), IEEE 802.3 (polia DSAP a SSAP), v IP pakete (pole Protocol), ako aj čísla portov v transportných protokoloch boli programom **načítané z jedného alebo viacerých externých textových súborov.** Pre známe protokoly a porty (minimálne protokoly v bodoch 1) a 4) budú uvedené aj ich názvy. Program bude schopný uviesť k rámcu názov vnoreného protokolu po doplnení názvu k číslu protokolu, resp. portu do externého súboru. Za externý súbor sa nepovažuje súbor knižnice, ktorá je vložená do programu.

6) V procese analýzy rámcov pri identifikovaní jednotlivých polí rámca ako aj polí hlavičiek vnorených protokolov nie je povolené použiť funkcie poskytované použitým programovacím jazykom alebo knižnicou. **Celý rámec je potrebné spracovať postupne po bajtoch.**

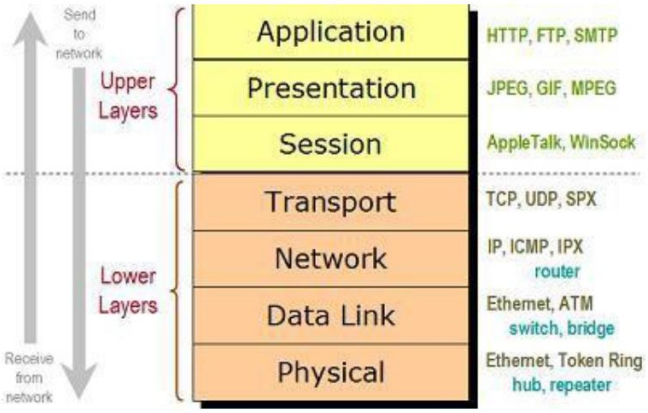
7) Program musí byť organizovaný tak, aby bolo možné jednoducho rozširovať jeho funkčnosť výpisu rámcov pri doimplementovaní jednoduchej funkčnosti na cvičení.

8) Študent musí byť schopný preložiť a spustiť program v miestnosti, v ktorej má cvičenia.

V danom týždni, podľa harmonogramu cvičení, musí študent priamo na cvičení doimplementovať do funkčného programu (podľa vyššie uvedených požiadaviek) ďalšiu prídavnú funkčnosť.

2. Analýza

Program by mal slúžiť ako prostriedok pre analýzu zachytených .pcap súborov. Jeho úloha je analyzovať zachytené rámce, ktoré boli zaznamenané pomocou monitorovacieho prostriedku (konkrétne pre naše potreby to bol program WireShark).
Analyzátor pracuje len po štvrtú vrstvu modelu OSI. Je však schopný rozoznať mnohé typy, najmä pre Ethernet 2 a rodinu protokolov TCP.



Obr. 1. – model OSI

3. Implementácia / Návrh riešenia

3.1 Špecifikácia požiadaviek

Program je na základe požiadaviek zo zadania realizovaný v programovacom jazyku Python. Realizovaný bol vo vývojárskom prostredí JetBrains PyCharm 2019.2.3, s využitím knižnice pcap.h. Táto knižnica bola nevyhnutnou súčasťou zadanej úlohy, pretože bez nej by prostredie nebolo schopné interpretovať zadané .pcap súbory.
Program bol napísaný na vlastnom notebooku, no napriek tomu je spustiteľný aj na škoslkých počítačoch, tak ako to upravuje znenie zadania. Jeho spustenie by však vyžadovalo manuálnu konfiguráciu knižnice pcap.

3.2 Hrubý návrh architektúry

Všetky časti programu boli naprogramované v jazyku Python. Zaujímavé bolo nájsť riešenie problému čítania zo zdrojového rámca, poskytnutého pre prostredie .pcap súborom. Na základe tejto vedomosti bolo vypracovanie ďalších častí prijateľne časovo aj vedomostne náročné. Celý rámec sa dal chápať ako mohutné pole údajov, s ktorým mohol programátor narábať vďaka zaužívaným konvexiám v oblasti sietí.

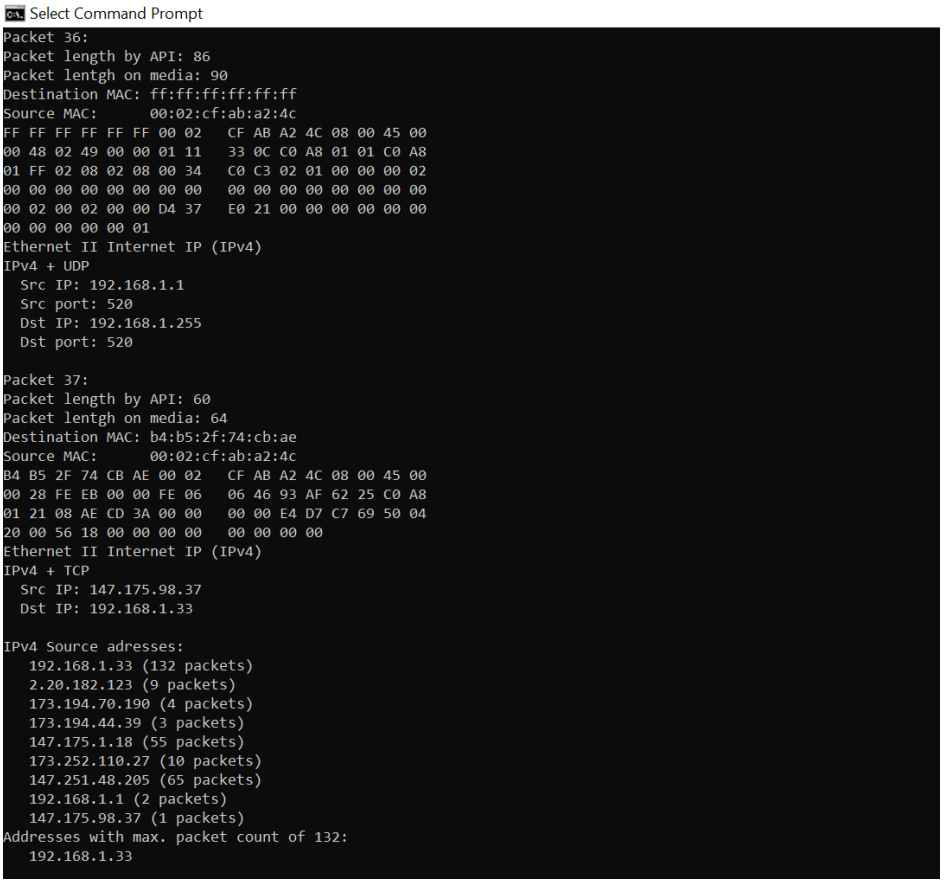
Vnorené protokoly a ich identifikácia prebiehali na základe porovnávania s textovými súbormi, ktoré boli vytvorené špeciálne za týmto účelom. Tým pádom je možné meniť výpis programu bez akéhokoľvek zásahu do samotného kódu.

3.3 Ukážkový vstup

Program obsahuje možnosť zadať manuálne vstup.
Napríklad do konzoly napísať: c:\Users\User\PycharmProjects\pks-z2>python pks-z2.py port.txt eth-3.pcap

3.4 Ukážkový výstup

Program zanalyzuje po spustení všetky rámce, ktoré .pcap súbor obsahuje. Na základe zadania som analyzovala všetky potrebné časti. Príklad výstupu môžeme vidieť na obrázku 2.

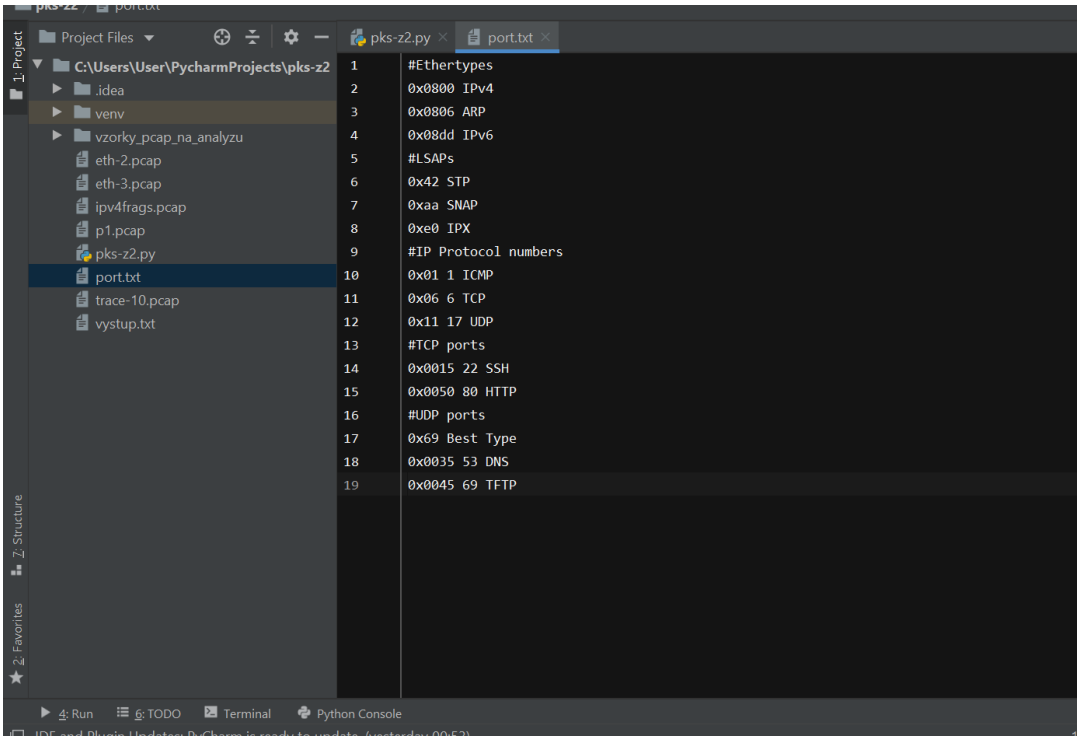


Obr 2. Ukážka výstupu

3.5 Pomocné .txt súbory

Na fungovanie programu je využitý jeden pomocný súbor, v ktorom sú uvedené známe hodnoty pre porty či protokoly. Jeho zmena by sa priamo prejavila zmenou údajov v ďalšej analýze, pretože pri porovnaní hodnôt by bola zistená zhoda a vypísaný nový názov.

Príklad súboru môžeme vidiet na obrázku 3.



Obr 3. Pomocný súbor

4. Záver

Program je vyústenie niekoľkých týždňov snahy o pochopenie fungovania takéhoto problému. Spĺňa všetky minimálne požiadavky pre korektné odovzdanie zadania a navyše spĺňa takmer všetky veci nad rámec minimálnych požiadaviek.