

Slovenská Technická Univerzita v Bratislave Fakulta informatiky a informačných technológií

Počítačové a komunikačné siete

Zadanie 1 – Analyzátor sieťovej komunikácie

Zadanie úlohy

Navrhните 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**.

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

IP adresy a počet odoslaných / prijatých 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.

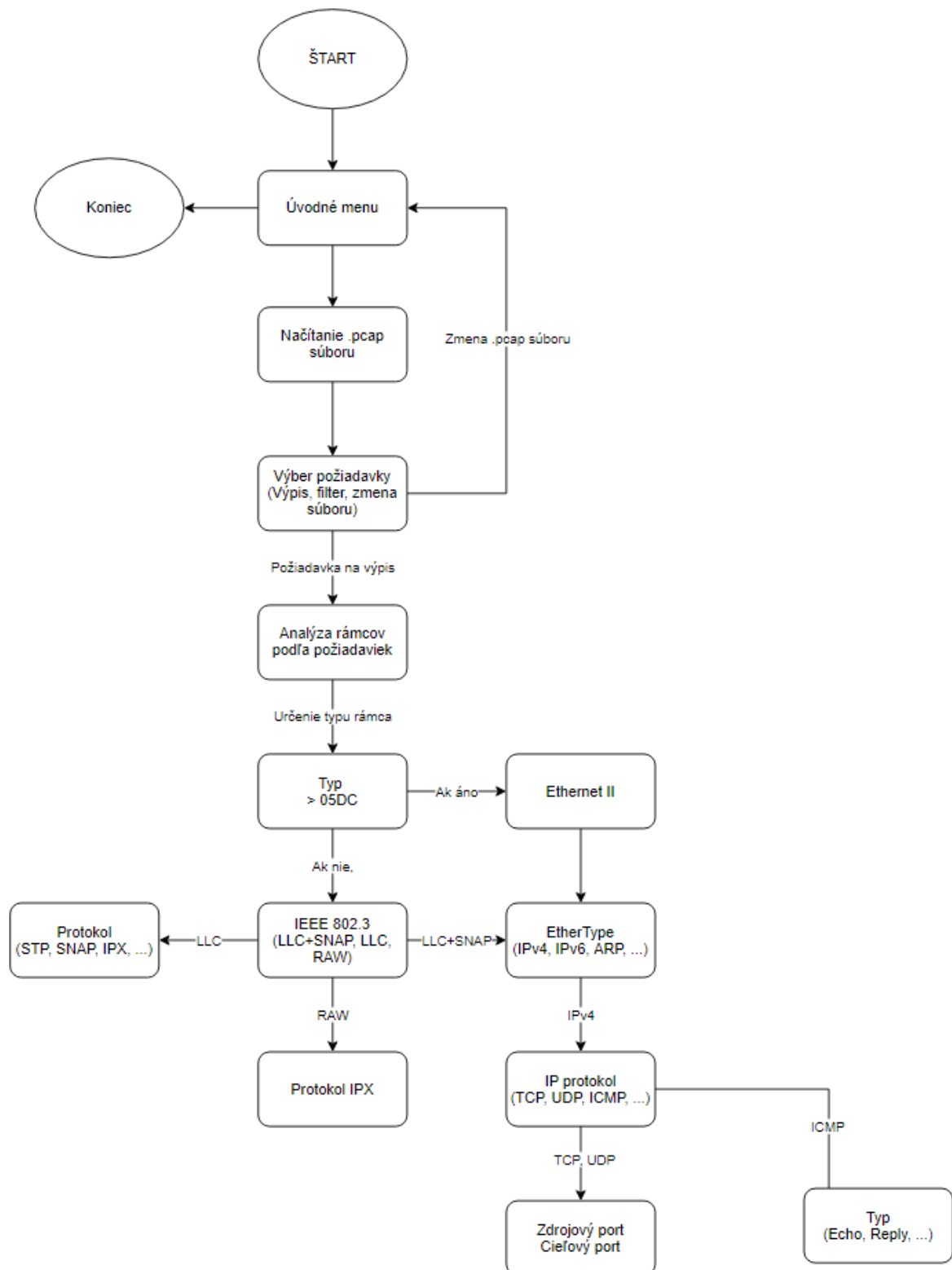
- i) **Všetky** ARP dvojice (request – reply), uveďte aj IP adresu, ku ktorej sa hľadá MAC (fyzická) adresa a pri ARP-Reply uveďte konkrétny pár - IP adresa a nájdená MAC adresa. V prípade, že bolo poslaných viacero rámcov ARP-Request na rovnakú IP adresu, vypíšte všetky. Ak sú v súbore rámce ARP-Request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

Blokový návrh – postup riešenia (fungovania)



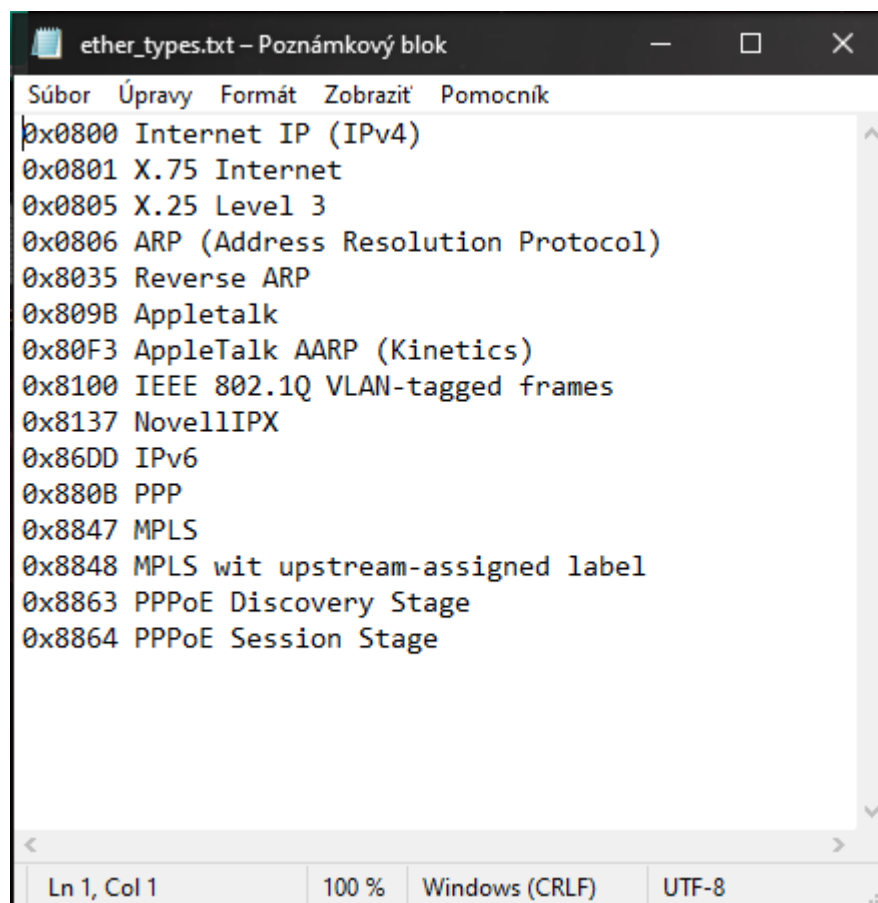
Mechanizmus analyzovania protokolov

Štruktúra externých súborov

PRAVIDLÁ

Nachádzajú sa v priečinku *rules*

- ether_types.txt
- icmp_types.txt
- ip_protocols.txt
- llc_saps.txt
- tcp_ports.txt
- udp_ports.txt



```
ether_types.txt - Poznámkový blok
Súbor  Úpravy  Formát  Zobrazit'  Pomocník
0x0800 Internet IP (IPv4)
0x0801 X.75 Internet
0x0805 X.25 Level 3
0x0806 ARP (Address Resolution Protocol)
0x8035 Reverse ARP
0x809B Appletalk
0x80F3 AppleTalk AARP (Kinetics)
0x8100 IEEE 802.1Q VLAN-tagged frames
0x8137 NovellIPX
0x86DD IPv6
0x880B PPP
0x8847 MPLS
0x8848 MPLS wit upstream-assigned label
0x8863 PPPoE Discovery Stage
0x8864 PPPoE Session Stage
Ln 1, Col 1    100 %    Windows (CRLF)    UTF-8
```

Formát pravidiel:

hex hodnota s veľkými písmenami, medzera, názov bez medzier

PCAP

Všetky pcap súbory na testovanie musia byť uložené v priečinku *examples*, aby ich program vedel vyhľadať.

Implementačné prostredie

Programovací jazyk: Python 3.8.6 64-bit

IDE: Visual Studio Code

Použité knihnice:

- binascii
- scapy
- os

```
1 from binascii import hexlify
2 import scapy.all as scapy
3 import os
```