Slovenská technická univerzita v Bratislave

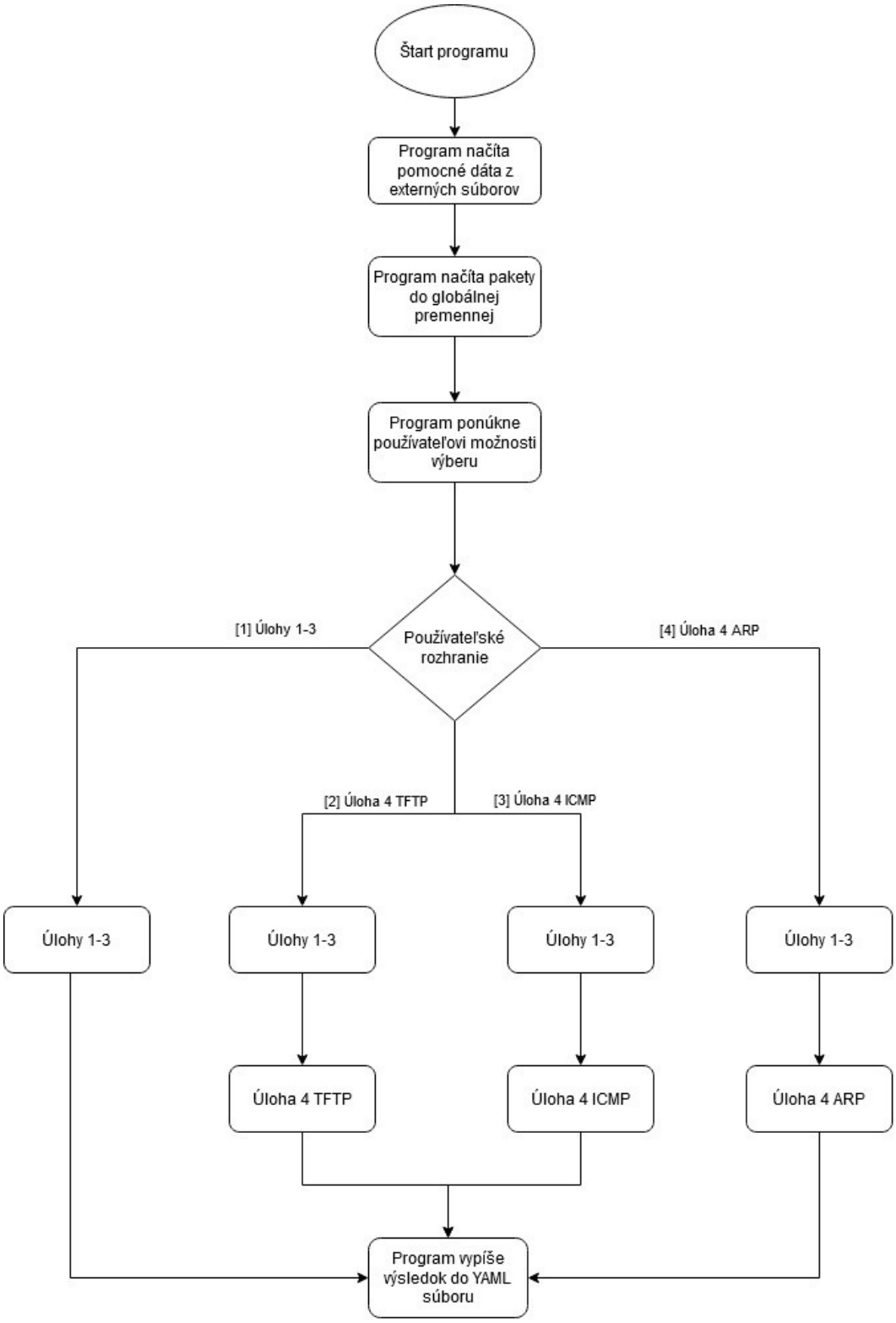
Fakulta informatiky a informačných technológií

Počítačové a komunikačné siete
Analyzátor sieťovej komunikácie

Samuel Kováč
2022/2023

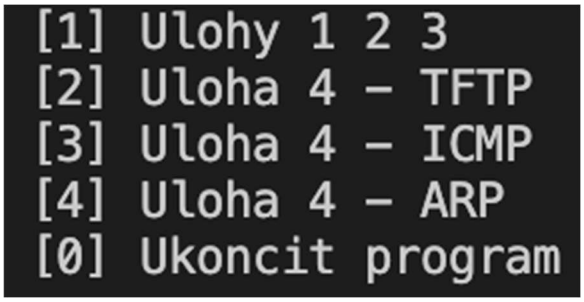
1. Stručné zadanie
   Navrhnite a implementujte programový analyzátor Ethernet siete, ktorý analyzuje komunikácie v sieti zaznamenané v .pcap súbore a poskytuje informácie o komunikáciách.

2. Blokový diagram programu



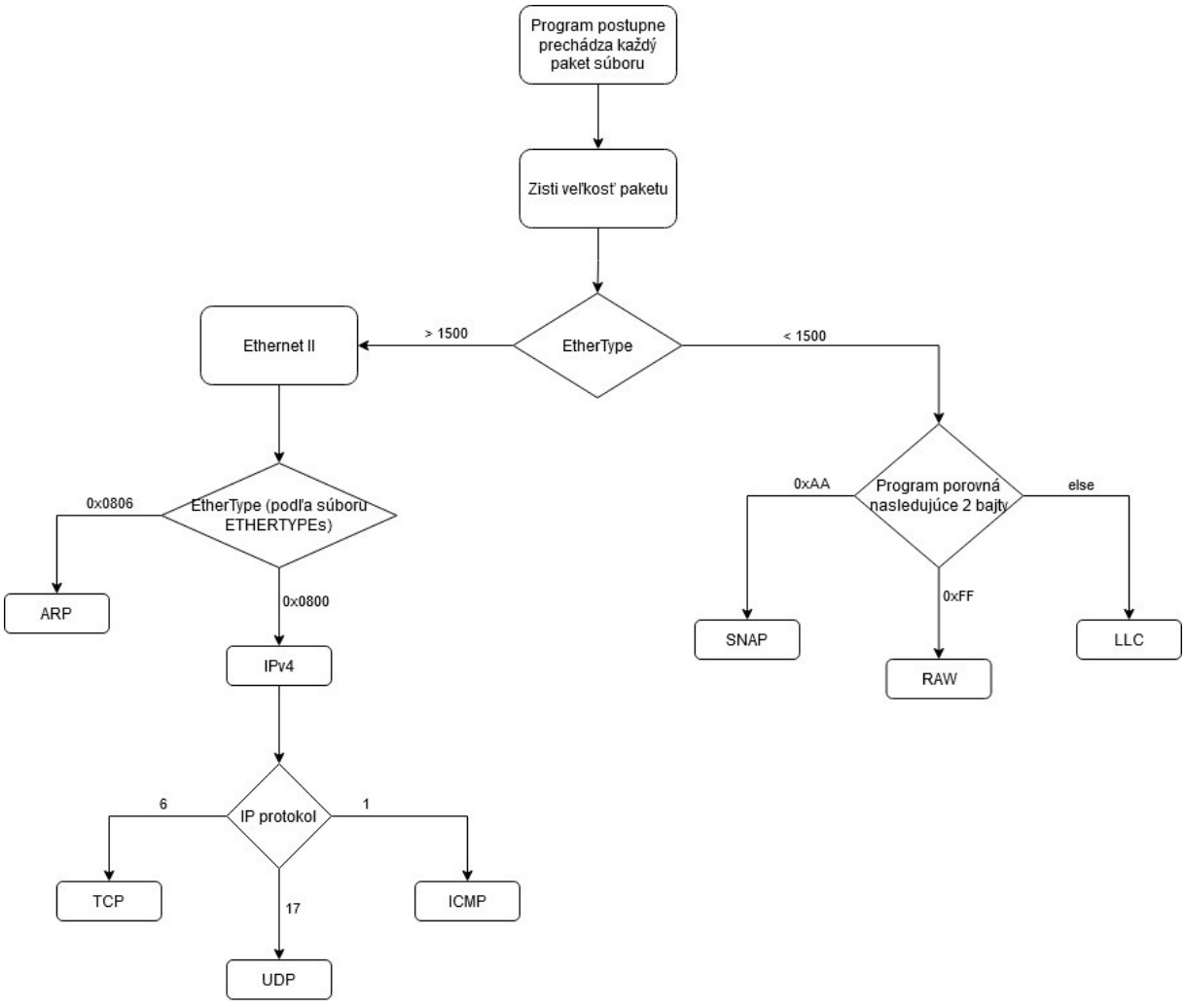
3. Popis programu

   Po spustení programu si program načíta pomocné zoznamy z externých súborov ETHERTYPEs.txt, IP_PROTOCOLs.txt, LSAPs.txt a TCP_PORTs.txt. Pcap súbor načítam do globálnej premennej *packetList* pomocou funkcie rpcap z knižnice pylibpcap.pcap. Používateľovi sa zobrazí grafické rozhranie s nasledujúcimi možnosťami :



   Pri výbere možnosti 1 sa spustí riešenie úloh 1-3 podľa zadania.
   Pri výbere možností 2-4 sa najprv spustia úlohy 1-3 a potom sa spustí funkcia podľa vybranej možnosti na základe zadania.
   Výberom možnosti 0 sa program ukončí.

4. Blokový diagram spracovania rámcov



5. Popis riešenia úloh 1-3

Program prechádza všetky pakety v súbore slučkou. Uloží si poradie paketu, následne zistí dĺžku rámca a začne analyzovať packet. Najprv porovná hodnotu EtherType s hodnotou 1500 v hexadecimálnom tvare (5DC). Ak je EtherType väčší, pozrieme sa na jeho hodnotu a porovnáme ju s ethertypes zo súboru ETHERTYPEs.txt. Ak sa jedná o IPv4 pokračujeme ďalej na porovnanie IP protokolu. V prípade že je EtherType menší ako 0x5DC (1500) pozrieme sa na nasledujúce 2 bajty a rozlíšime typ rámca na SNAP / RAW / LLC.

6. Riešenie úlohy 4

Podarilo sa mi implementovať len úlohy **f** – **j**. Postup riešenia týchto úloh je dosť podobný. Na začiatku program opäť prejde všetky pakety a vyfiltruje z nich len tie ktoré nás v danej úlohe zaujímajú. Podľa úlohy môže program zisťovať z paketu informácie ktoré neboli obsiahnuté v úlohách 1-3 ako napríklad *opcode* pri ARP alebo *icmp_type* pri ICMP. Následne tieto vyfiltrované pakety začne párovať a spájať do komunikácií. Pakety ktoré zostanú nespárované, vypíše ako *partial_communication*.

7. Implementačné prostredie

Program bol písaný v jazyku python, realizovaný už spomínanou knižnicou pylibpcap

8. Príklad štruktúry externého súboru

TCP_PORTs.txt


```
{
   "7": "ECHO",
   "19": "CHARGEN",
   "20": "FTP-DATA",
   "21": "FTP-CONTROL",
   "22": "SSH",
   "23": "TELNET",
   "25": "SMTP",
   "53": "DNS",
   "69": "TFTP",
   "79": "FINGER",
   "80": "HTTP",
   "110": "POP3",
   "111": "SUNRPC",
   "119": "NNTP",
   "139": "NETBIOS-SSN",
   "143": "IMAP",
   "179": "BGP",
   "389": "LDAP",
   "443": "HTTPS",
   "445": "MICROSOFT-DS",
   "1080": "SOCKS"
}
```