

Fakulta informačních technologií  
Vysoké učení technické v Brně

Počítačové komunikace a sítě – 2. projekt  
Varianta ZETA: Sniffer paketů

# Obsah

|          |                             |          |
|----------|-----------------------------|----------|
| <b>1</b> | <b>Úvod</b>                 | <b>2</b> |
| <b>2</b> | <b>Implementace</b>         | <b>2</b> |
| <b>3</b> | <b>Testování</b>            | <b>2</b> |
| <b>4</b> | <b>Kompilace a spuštění</b> | <b>3</b> |
| <b>5</b> | <b>Příloha</b>              | <b>4</b> |

# 1 Úvod

Sniffer paketů je síťový analyzátor, který umožňuje odchycení a filtrování paketů na určitém síťovém rozhraní. V počítačových sítích se používá k analýze komunikace s různými protokoly, v tomto případě TCP, UDP, ICMP a IGMP z rodiny TCP/IP[4].

## 2 Implementace

Analyzátor je napsaný v jazyce C s podpůrnými knihovnami jako `pcap.h` a další. Když se sniffer spustí bez parametrů, vypíše se seznam dostupných rozhraní pomocí funkce `findalldevs` z již zmiňované knihovny `pcap.h`[3], která vrací požadovaný seznam rozhraní. Po vypsání tohoto seznamu se program ukončí. Naopak zadání rozhraní při spuštění má za následek spojení s rozhraním, čímž otevřeme dané rozhraní pro naslouchání paketů. Díky funkci `pcap_open_live`[1] můžeme toto spojení navázat. Po vytvoření spojení program odchytí paket, `pcap_next_ex`, a uloží ho do bufferu.

Po získání ip hlavičky paketu může program zjistit protokol, na kterém paket pracuje. Pokud byl zadán parametr `-n`, program odchytává počet paketů specifikován hodnotou tohoto parametru, zda-li tento parametr zadán nebyl, zpracovává se pouze jeden paket. Další parametry, které ovlivňují filtraci paketů jsou `-t` nebo `--tcp`, které mají za následek filtrování TCP paketů, `-u` nebo `--udp`, které zase filtrují UDP pakety, `--icmp`, který filtruje ICMP pakety a `--igmp`, filtrující IGMP pakety. V případě filtrování pouze ICMP/IGMP paketů jsou zde použitelné parametry `--icmp-only` a `--igmp-only`. Bez použití žádného z těchto filtrujících parametrů, program filtruje pouze UDP a TCP pakety, dle zadání. Podobně také s použitím parametru `-p` sniffer filtruje příchozí a odchozí pakety na základě portu, na který, popřípadě ze kterého, paket přichází nebo odchází.

Po rozlišení protokolu, následuje uložení hlavičky paketu do příslušné struktury, `udphdr` pro UDP, `tcphdr` pro TCP, a vypsání informací o paketu jako čas odchycení, zdrojová a cílová ip adresa nebo doménové jméno a port. Pro získání doménového jména je použita funkce `getaddrinfo` z knihovny `netdb.h`. Není-li pro danou ip adresu nalezeno doménové jméno, je, ve výpise, použita ip adresa. Po vypsání těchto dat z paketu následuje vypsání dat celého paketu, včetně hlaviček, v přehledném formátu.

## 3 Testování

Pro testování snifferu byl, mimo jiné, použit nástroj `curl`[2], se kterým se jednoduše simuluje provoz na síti. Dále jsem použil program `Wireshark`[5], se kterým byl porovnáván výstup snifferu. Po spuštění paket snifferu pro naslouchání na určitém rozhraní, byl spuštěn i program `Wireshark` na téže rozhraní. Po přijetí paketu a jeho výpisu jsem pozastavil `Wireshark` a porovnal výstupy obou programů, pro zajištění správnosti našeho snifferu paketů.<sup>1</sup>

---

<sup>1</sup>Viz. obrázky v příloze

## 4 Kompilace a spuštění

Pro kompilaci programu je k dispozici Makefile, který ho pomocí překladače gcc přeloží a vytvoří spustitelný soubor. Ten je terminálová aplikace, spustitelná pomocí:

```
./ipk-sniffer -i rozhrani [-p port] [--tcp|-t] [--udp|-u] [--icmp] [--igmp]
[--icmp-only] [--igmp-only] [-n num] [--help]
```

kde `-i` je rozhraní, na kterém se odchyťávají pakety, `-p` je port, na kterém se odchyťávají pakety, `-t/--tcp` je filtrace TCP paketů, `-u/--udp` je filtrace UDP paketů, `--icmp` je filtrace ICMP paketů, `--icmp-only` je filtrace pouze ICMP paketů, `--igmp` je filtrace IGMP paketů, `--igmp-only` je filtrace pouze IGMP paketů, `--help` je výpis nápovědy,

## 5 Příloha

```
~/Documents/4. semestr/IPK/ipk-project-2(master*) » sudo ./ipk-sniffer -i wlo1 -p 80 -n 10
10:00:28.54865 pop-os : 37096 > one.one.one.one : 80

0x0000: 0c 80 63 e0 6a 1e 20 16 b9 84 dc c1 08 00 45 00 ..c.j... ..E.
0x0010: 00 3c 71 7c 40 00 40 06 06 2c c0 a8 00 6a 01 01 .<q|@. @. .j..
0x0020: 01 01 90 e8 00 50 3c 0c d0 ef 00 00 00 00 a0 02 ....P<. ....
0x0030: fa f0 c3 42 00 00 02 04 05 b4 04 02 08 0a 27 ed ...B.... ..'.
0x0040: d6 2a 00 00 00 01 03 03 ..*.....

10:00:28.56343 one.one.one.one : 80 > pop-os : 37096

0x0000: 20 16 b9 84 dc c1 0c 80 63 e0 6a 1e 08 00 45 00 ..... c.j...E.
0x0010: 00 34 00 00 40 00 38 06 7f b0 01 01 01 01 c0 a8 .4..@.8. ....
0x0020: 00 6a 00 50 90 e8 8e 6d 22 45 3c 0c d0 f0 80 12 .j.P...m "E<....
0x0030: ff ff 5d 02 00 00 02 04 05 b4 01 01 04 02 01 03 ..].....
0x0040: 03 .

10:00:28.57315 pop-os : 37096 > one.one.one.one : 80

0x0000: 0c 80 63 e0 6a 1e 20 16 b9 84 dc c1 08 00 45 00 ..c.j... ..E.
0x0010: 00 28 71 7d 40 00 40 06 06 3f c0 a8 00 6a 01 01 .(q|@. @. .?..j..
0x0020: 01 01 90 e8 00 50 3c 0c d0 f0 8e 6d 22 46 50 10 ....P<. ...m"FP.
0x0030: 01 f6 c3 2e 00 .....

10:00:28.57795 pop-os : 37096 > one.one.one.one : 80

0x0000: 0c 80 63 e0 6a 1e 20 16 b9 84 dc c1 08 00 45 00 ..c.j... ..E.
0x0010: 00 6f 71 7e 40 00 40 06 05 f7 c0 a8 00 6a 01 01 .oq~@. @. ....j..
0x0020: 01 01 90 e8 00 50 3c 0c d0 f0 8e 6d 22 46 50 18 ....P<. ...m"FP.
0x0030: 01 f6 c3 75 00 00 47 45 54 20 2f 20 48 54 54 50 ...u..GE T./..HTTP
0x0040: 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 2e 31 2e /1.1..Ho st:1.1.
0x0050: 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 1.1..Use r-Agent:
0x0060: 20 63 75 72 6c 2f 37 2e 36 35 2e 33 0d 0a 41 63 .curl/7. 65.3..Ac
0x0070: 63 65 70 74 3a 20 2a 2f 2a 0d 0a 0d cept:.*/*...
```

Obrázek 1: Výstup ipk-sniffer.c

The image shows a Wireshark packet capture window and a terminal window. The Wireshark window displays a list of captured packets, with the selected packet (Frame 3) showing details of an HTTP GET request to 1.1.1.1. The terminal window shows the output of the curl command, displaying the HTML response from Cloudflare.

Wireshark Packet List:

| No. | Time        | Source        | Destination   | Protocol | Length | Info  |
|-----|-------------|---------------|---------------|----------|--------|---|
| 3   | 0.212689966 | 192.168.0.106 | 1.1.1.1       | TCP      | 74     | 37096 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=669898282 TSecr=0 |
| 4   | 0.235976558 | 1.1.1.1       | 192.168.0.106 | TCP      | 66     | 80 → 37096 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1 WS=1024      |
| 5   | 0.236026831 | 192.168.0.106 | 1.1.1.1       | TCP      | 54     | 37096 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0  |
| 6   | 0.236219546 | 192.168.0.106 | 1.1.1.1       | HTTP     | 125    | GET / HTTP/1.1  |
| 7   | 0.252517977 | 1.1.1.1       | 192.168.0.106 | TCP      | 54     | 80 → 37096 [ACK] Seq=1 Ack=72 Win=65536 Len=0                                       |
| 8   | 0.265419201 | 1.1.1.1       | 192.168.0.106 | TCP      | 657    | 80 → 37096 [PSH, ACK] Seq=1 Ack=72 Win=65536 Len=603 [TCP segment of a reassembl... |
| 9   | 0.265439311 | 192.168.0.106 | 1.1.1.1       | TCP      | 54     | 37096 → 80 [ACK] Seq=72 Ack=604 Win=63744 Len=0                                     |
| 10  | 0.265644147 | 1.1.1.1       | 192.168.0.106 | HTTP     | 59     | HTTP/1.1 301 Moved Permanently (text/html)  |
| 11  | 0.265648882 | 192.168.0.106 | 1.1.1.1       | TCP      | 54     | 37096 → 80 [ACK] Seq=72 Ack=609 Win=64128 Len=0                                     |
| 12  | 0.265728333 | 192.168.0.106 | 1.1.1.1       | TCP      | 54     | 37096 → 80 [FIN, ACK] Seq=72 Ack=609 Win=64128 Len=0                                |
| 13  | 0.303163711 | 1.1.1.1       | 192.168.0.106 | TCP      | 54     | 80 → 37096 [FIN, ACK] Seq=609 Ack=73 Win=65536 Len=0                                |
| 14  | 0.303186251 | 192.168.0.106 | 1.1.1.1       | TCP      | 54     | 37096 → 80 [ACK] Seq=73 Ack=610 Win=64128 Len=0                                     |

Terminal Output:

```
~/Documents/4. semestr/IPK/ipk-project-2(master*) » curl 1.1.1.1
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>cloudflare-lb</center>
</body>
</html>
~/Documents/4. semestr/IPK/ipk-project-2(master*) »
```

Obrázek 2: Výstup Wireshark a curl

## Literatura

- [1] Carstens, T.: Programming with pcap. [online], sekce Opening device for sniffing.  
URL <https://www.tcpdump.org/pcap.html>
- [2] Curl: Documentation. [online].  
URL <https://curl.haxx.se/docs/>
- [3] Wikipedia: pcap. [online].  
URL <https://cs.wikipedia.org/wiki/Pcap>
- [4] Wikipedia: TCP/IP. [online].  
URL <https://cs.wikipedia.org/wiki/TCP/IP>
- [5] Wireshark, F.: Documentation. [online].  
URL <https://www.wireshark.org/docs/>