

Počítačové komunikace a sítě 2019/2020

Sniffer paketů

Daša Nosková (xnosko05)

Obsah

1	Úvod	2
1.1	Paket	2
1.1.1	Fyzická a linková úroveň	2
1.1.2	Sieťová úroveň	2
1.1.3	Transportná úroveň	2
2	Implementácia	3
3	Testovanie	4

Kapitola 1

Úvod

Cieľom projektu je vytvoriť sieťový analyzátor, ktorý na danom porte zachytáva a filtruje pakety. Program zachytáva každý paket vyhovujúci filtrom na danom sieťovom rozhraní.

1.1 Paket

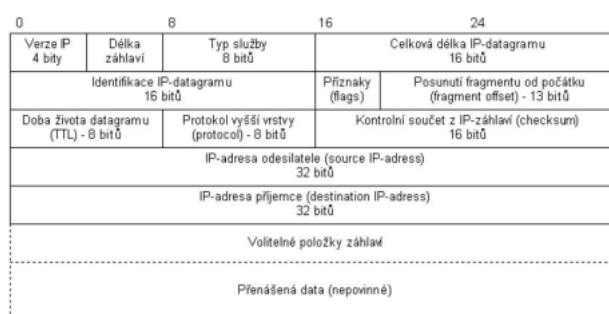
Pakety prenášajú dáta rôznej dĺžky z jedného počítača na druhý. Dáta sú prenášané cez viac vrstiev, pričom pri každej vrstve sa pripoja k paketu nové informácie. Internet využíva na prenos paketov protokol TCP/IP, ktorý sa skladá zo 4 vrstiev: linková a fyzická vrstva, IP, TCP/UDP, aplikačná vrstva.[1]

1.1.1 Fyzická a linková úroveň

Ethernet obsahuje 6 bajtovú linkovú adresu príjemcu a adresu odosielateľa, pole špecifikujúce protokol sieťovej vrstvy a špecifikáciu protokolu. [1]

1.1.2 Sieťová úroveň

Internet protokol, IP, reprezentuje sieťovú vrstvu. Prenáša IP-datagramy, ktoré sa skladajú zo záhlavia a prenášaných dát. Záhlavie má spravidla 20 B a obsahuje informácie napríklad o IP adresách príjemcu a odosielateľa.[1]



Obr. 1.1: IP datagram
[1]

1.1.3 Transportná úroveň

Transportná vrstva sa skladá z protokolov TCP a UDP zaistujúcich spojenie medzi aplikáciami na 2 počítačoch. TCP potvrdzuje prijímané dáta, UDP nie.[1] Každý protokol má priradené číslo, tj. pre TCP = 6 a pre UDP = 17. Adresou je port.

Kapitola 2

Implementácia

Program je zložený zo súborov `ipk-sniffer.c`, `ipk-sniffer.h`, `filter.c`, `filter.h`, `error.h`.

Program začína parsovaním argumentov pomocou knižnice `argp.h`. Implicitne sú nastavené hodnoty parametrov ako na obrázku 2.

```
// default values of arguments
arguments.interface = NULL;
arguments.port = -1;
arguments.tcp = false;
arguments.udp = false;
arguments.num = 1;
```

Po dokončení spracovania argumentov sa zavolá funkcia `sniff` s priloženými argumentami a prípadnými filrami. Vo funkcii `sniff()` sa otvorí rozhranie na ktorom sa budú zachytávať pakety, nastaví sa prípadný filter a začne sa zbierať určený počet paketov.

Vo funkcii `process_paket()` sa rozdelí paket na hlavičky jednotlivých vrstiev z ktorých sa extrahujú informácie, ktoré sú následne ako aj celý obsah paketu vypísané.

Funkcia `check_protocol()` skontroluje či ide o protokol TCP alebo UDP a daný typ predá ako parameter funkcii `get_port()`, ktorá vracia čísla portov príjmateľa a odosielateľa. V prípade ak protokol nie je TCP ani UDP, program vypíše upozornenie, a pokračuje ďalej.

Funkcia `check_src_dst_addr()` vracia IP adresu príjmateľa a odosielateľa. V programe nie je implementovaný prevod IP adresy na doménové meno.

Celý obsah paketu sa vypisuje pomocou funkcie `print_packet()`, ktorá pri každom zavolaní vypíše 16 B z daného paketu a 16 znakov v ASCII. Konverzia hexadecimálneho čísla na ascii hodnotu zabezpečuje funkcia `convert_ascii()`.

Kapitola 3

Testovanie

Testovanie bolo vykonané porovnaním výpisom paketov s paketmi z programu Wireshark.

```
0x0100: EC 21 80 63 C7 55 E6 3A 8C 05 DA F9 C2 E7 C9 C8 .l.c.U.....
0x0110: 45 4F 2A 64 ED 4C CA 10 39 B4 C7 E4 68 07 38 AF E0*d.L...9...k.8.
0x0120: 0F FC 35 DE 21 65 F2 AB 59 D8 72 DC B4 0D F4 50 .5.le.Y.r...P
0x0130: CB B9 1A 8D A0 AB 48 B5 E2 FF A4 76 16 F9 D6 48 .....H...v...P
0x0140: 98 6F 70 06 F3 EC CE E4 E6 68 87 C2 C0 9C 49 75 .op....k....Iu
0x0150: 28 62 C3 06 58 C5 04 35 77 F9 A9 82 CA 63 82 75 (b..X..Sw....c.u
0x0160: 08 2D BD 24 50 FC 3C DB A5 BE 28 30 61 23 2E A6 ..$].<... (0a#...
0x0170: 84 48 9B 78 07 0E 9B E5 40 37 71 C2 1C EA D5 3D .K.{...M7q....=
0x0180: F8 FE 7A 25 28 97 74 22 A7 02 51 C8 37 ..z*+.t"...Q.7

Packet number [9], length of this packet is: 289
11:32:26.986672 35.186.224.53 : 443 > 192.168.1.148 : 40764
0x0000: F8 63 3F 15 E8 B4 F0 2F A7 97 0A 58 08 00 45 00 .c?..../.X..E.
0x0010: 01 13 74 08 00 00 79 06 06 AE 23 BA E0 35 C0 A8 .t...y...e#.5...
0x0020: 01 94 01 BB 9F 3C BC 4B FF 74 1A 69 99 DA 80 18 .....<L.S.i....
0x0030: 04 1A 55 B3 00 00 01 01 08 0A C2 24 5F 3E 70 06 ..0d....$?p...
0x0040: DE 98 17 03 03 00 DA FE CB 75 08 07 8F 2A 05 B8 de 98 17 03 03 00 DA FE CB 75 08 07 8F 2A 05 B8
0x0050: 3D 3F 20 32 00 9E 91 CD B3 4E E1 68 37 59 87 CB =?..2'....N.h7y...
0x0060: 0E 69 D8 04 C9 AE 2C D0 0E 7A 20 B9 78 50 5C 43 .t.....z...{P[C
0x0070: B4 D5 57 16 68 1F A3 70 4B B4 EA F2 AA 36 4D EE .W.h...pK...6M.
0x0080: AC CE 92 5C 4C FA 2B 5D EE 98 F1 76 2B 68 11 47 ...[L+]...v+k.G
0x0090: 98 C3 86 F7 D3 38 17 3F 70 3B D4 49 3F E4 6F B4 .....;?};.I?..o.
0x0100: D5 A4 C0 B3 8D F1 B0 28 1C 15 4D E8 AF 06 8D 67 .....(.M...g
0x0110: A9 AC 37 22 8C C6 2A B3 E8 9A 53 25 39 31 79 F1 .7"...*..$K91y.
0x0120: 02 79 74 0C 1E B4 3D 3A 58 67 22 86 A4 1B E8 7B .yt...=:Xg"....{
0x0130: 16 F5 B4 F7 CA 79 B8 0B F0 FF 02 E5 B4 0C 26 .....y.....j&
0x0140: 54 2B EB 9B 77 39 22 D1 10 6D 52 7A A2 32 0F CD T+..w9"...RZ.2..
0x0150: C2 37 5E B1 AD 03 57 93 63 A8 D6 C8 0C 3E 3F 0A .7^...M.c....>?.
0x0160: 08 B3 F5 C8 B4 52 C7 D1 2F 6E 82 E9 75 77 1F 35 .....R../n..uw.5
0x0170: B4 A7 26 EF 18 2F 9B 81 48 12 C3 F0 1F 3C A1 01 .8../..H....<..
0x0180: 93

Packet number [10], length of this packet is: 105
11:32:26.986678 35.186.224.53 : 443 > 192.168.1.148 : 40764
0x0000: F8 63 3F 15 E8 B4 F0 2F A7 97 0A 58 08 00 45 00 .c?..../.X..E.
0x0010: 00 5B 74 0C 00 00 79 06 07 65 23 BA E0 35 C0 A8 .[t...y...e#.5...
0x0020: 01 94 01 BB 9F 3C BC 4C 00 53 1A 69 99 DA 80 18 .....<L.S.i....
0x0030: 04 1A 4F 04 00 00 01 01 08 0A C2 24 5F 3F 70 06 ..0d....$?p...
0x0040: DE 98 17 03 03 00 22 10 8D A1 E5 C9 E4 98 D4 60 .....".
0x0050: 82 66 35 62 BF 39 61 CD 44 03 55 E6 A6 79 AE 70 .f5b.9a..D.U...y.p
0x0060: B0 50 E2 A6 93 49 66 A8 1E ..P...If..

dasa@dasa: ~/Dokumenty/school/TPK/proj2$
```

```
*wireshark
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp
No. Time Source Destination Protocol Length Info
8 4.185061256 53.224.186.35.bc.go... dasa.local TCP 66 443 -> 40764 [ACK] Seq=1 A
9 4.185456651 53.224.186.35.bc.go... dasa.local TLsv1.2 105 Application Data
10 4.204688396 53.224.186.35.bc.go... dasa.local TLsv1.2 152 Application Data
11 4.204857688 dasa.local 53.224.186.35.bc.go... TCP 66 40764 -> 443 [ACK] Seq=328
12 4.206388013 53.224.186.35.bc.go... dasa.local TLsv1.2 301 Application Data
13 4.206410121 53.224.186.35.bc.go... dasa.local TLsv1.2 289 Application Data
14 4.206415743 53.224.186.35.bc.go... dasa.local TLsv1.2 105 Application Data
15 4.206494913 dasa.local 53.224.186.35.bc.go... TCP 66 40764 -> 443 [ACK] Seq=328
16 4.206992007 dasa.local 53.224.186.35.bc.go... TLsv1.2 105 Application Data
17 4.222124688 53.224.186.35.bc.go... dasa.local TCP 66 443 -> 40764 [ACK] Seq=623...

Frame 14: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0
Ethernet II, Src: HuaweiTe_97:0a:58 (f0:2f:a7:97:0a:58), Dst: IntelCor_15:e8:84 (f8:63:3f:15:e8:84)
Internet Protocol Version 4, Src: 53.224.186.35.bc.googleusercontent.com (35.186.224.53), Dst: dasa.local
Transmission Control Protocol, Src Port: 443, Dst Port: 40764, Seq: 584, Ack: 328, Len: 39
Transport Layer Security
```

```
0000 f8 63 3f 15 e8 84 f0 2f a7 97 0a 58 08 00 45 00 .c?..../.X..E.
0010 00 5b 74 0c 00 00 79 06 07 65 23 ba e0 35 c0 a8 .[t...y...e#.5...
0020 01 94 01 bb 9f 3c bc 4c 00 53 1a 69 99 da 80 18 .....<L.S.i....
0030 04 1a 4f 04 00 00 01 01 08 0a c2 24 5f 3f 70 06 ..0d....$?p...
0040 de 98 17 03 03 00 22 10 8d a1 e5 c9 e4 98 d4 60 .....".
0050 82 66 35 62 bf 39 61 cd 44 03 55 e6 a6 79 ae 70 .f5b.9a..D.U...y.p
0060 b0 50 e2 a6 93 49 66 a8 1e ..P...If..
```

Obr. 3.1: Porovnanie paketov

Literatúra

- [1] Klement, M.: *Technologie počítačových sítí – úvod do problematiky počítačových sítí*. Olomouc: Univerzita Palackého v Olomouci, druhé vydání, 2019, ISBN 978-80-244-5580-8.