VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Počítačové komunikace a sítě – 2. projekt

Sniffer paketů

24. apríla 2022

Juraj Dedič

Obsah

1 Zachytávanie paketov

Úlohou bolo vytvoriť program na zachytávanie, filtrovanie a zobrazovanie paketov v unixovom prostredí.

Pakety sú prijímané na učritom sieťovom rozhraní (sieťovej karte) z jedného zariadenia na druhé[1]. Pakety sa skladajú z Ethernet rámca, ktorý môže obsahovať IP paket, alebo inú technológiu na sieťovej vrstve. Ethernet rámec obsahuje informácie potrebné pre posielanie dát v rámci jednej siete. Hlavička tohto rámca obsahuje napríklad zdrojovú a cieľovú MAC adresu alebo veľkosť paketu v bajtoch[3]. Hlavička IP paketu obsahuje 32-bitové IPv4 adresy zariadení medzi ktorými prebieha komunikácia, pri IPv6 pakete majú adresy šírku 128-bitov. Na transportnej vrstve môžu najčastejšie byť protokoly TCP alebo UDP. Hlavičky oboch týchto protokolov obsahujú zdrojový a cieľový port.

2 Implementácia

Zvolený implmentačný jazyk je C++, s pomocou knižníc libpcap a libnet. Pri začiatku práce s libpcap bola dosť nápomocná stránka s dokumentáciou a návodmi k tejto knižnici [4].

2.1 Spracovanie argumentov

Spracovanie argumentov je riešené manuálne bez použitia knižnice na to určenej. To umožňuje väčšiu možnosť pri implementácií spracovania argumentov podľa špecifikácie. Kontrola argumentov prebieha vo for cykle. Po skončení tohto cyklu sa ešte skontroluje, či bol zadaný protokol, ktorý je treba filtrovať. Ak bol protokol vynechaný, budú sa zobrazovať všetky zadané protokoly.

Ak nebolo zadané sieťové zariadenie, na ktorom treba zachytávať pakety, budú zobrazené všetky dostupné zariadenia.

2.2 Zachytávanie a filtrovanie

Pred zachytávaním paketu je potrebné otvoriť zadané zariadenie. Pre tento účel je použitá funkcia `pcap_open_live`.

Zachytávanie je vykonávané v `while` cykle, s podmienkou, že počet prijatých paketov odpovedajúcich vstupným parametrom je menší ako číslo n. Paket je prijatý pomocou `packet_next` do bufferu.

Následne sa pomocou Ethretnetovej hlavičky určí typ paketu, z ktorých nás zaujímajú `IP`,`IPV6` a `ARP`. Iný typ znamená že paket nebude dalej spracovávaný. V prípade IP paketu, bude získaná IP hlavička obsahujúca typ prenášaného protokolu. Medzi týmito protokolmi hľadáme ICMP, TCP a UDP. v prípade TCP a UDP porovnáme port vo funkcii `has_port`, ktorá v prípade špecifikovaného portu zaistí, že budú zobrazované len pakety s takýmito portami (buď zdrojovým alebo cieľovým). Podobne je to aj s paketom IPV6, kde sú protokoly TCP, UDP alebo ICMPv6.

Ak je paket ARP, žiadne ďalšie filtrovanie nie je potreba.

2.3 Výpis informácií

Po zistení či paket zodpovedá vstupným parametrom nasleduje prípadný výpis potrebných informácií. U paketov ktoré nechceme vypisovať zostane premenná `protocol` typu `PacketProtocol` v stave `Unset`, v opačnom prípade bude obsahovať typ protokolu. Podľa tohto sa pri výpise rozhoduje.

Ako prvá je vypísaná informácia o časovej pečiatke paketu podľa normy RFC3339 [6]. Pečiatka je získaná z hlavičky paketu poskytovanej knižnicou pcap. Tu sa zdá najvhodnejšie konvertovať časovú pečiatku do UTC času, pre jednotný výpis zulu času.

Ďalšia časť výpisu, ktorá si vystačí s hlavičkou ethernet rámca je výpis zdrojovej a cieľovej MAC adresy aj veľkosti paketu v bajtoch. Výpis týchto informácií (čas aj ethernet rámec) je realizovaný vo funkcii `print_eth_header`, ktorá je volaná pred výpisom ostatných detailov.

[1] Môžu byť taktiež loopback rozhrania alebo virtuálne rozhrania, kde komunikácia prebieha na jednom fyzickom zariadení

Následne podľa hodnoty premennej `protocol` program rozhodne, ktorú funkciu pre výpis zavolá.

2.4 Výpis ICMP paketu

Keď že pod protokolom ICMP sa nachádza buď protokol IP alebo IPv6, vypíšeme najprv hlavičku príslušného protokolu. Následne je zavolaná funkcia `print_payload`, ktorá vypíše obsah paketu.

2.5 Výpis TCP paketu

Pre výpis TCP paketu potrebujeme taktiež najprv vypísať informácie o hlavičke IP prípadne IPv6 protokolu. Nasledovne treba vypočítať veľkosť hlavičky IP z `ihl` parametru. Pri IPv6 je táto hodnota vždy 40 bajtov [7]. Pre získanie informácii z TCP rámca je použitá `struct tcphdr` s potrebným offsetom. Tento vypočítame súčtom veľkosti hlavičiek IP a Ethernetu.

Následne po priradení do štruktúry, vieme získať zdrojový a cieľový port. Porty sú beznamienkové 16-bitové integery, no je potreba ich pred výpisom konvertovať do správneho endianu pomocou `ntohs`.

Ako posledné sa volá funkcia pre výpis paketu.

2.6 Výpis UDP paketu

Výpis UDP paketu obsahuje podobný postup ako pri TCP. Takže výpis hlavičiek IP a IPv6 je riešený rovnako, taktiež aj výpočet offsetu pre získanie UDP hlavičky.

Po priradení do `struct udphdr` s potrebným posunom (tj. súčet veľkostí Ethernet a IP/IPv6 hlavičiek), vieme získať porty. Čísla portov sú taktiež konvertované do správneho endianu a vypísané na štandardný výstup.

2.7 Výpis ARP paketu

Tento protokol je na linkovej vrstve takže takýto paket nemá IP hlavičku. Pri funkcii pre výpis ARP paketu sa zavolá funkcia pre výpis celého paketu.

2.8 Výpis informácii o Internet Protokole (IPv4)

Pri výpise hlavičky IP je použitá `struct iphdr` z ktorej získame zdrojovú a cieľovú IP adresu. Pre priradenie štruktúry správne musíme k ukazateľu na paket pripočítať aj veľkosť Ethernetovej hlavičky, ktorá je uložená v makre `ETH_HLEN`. Pri výbere informáciú z IP paketu bola užitočná [1].

2.9 Výpis informácii o Internet Protokole (IPv6)

Pre výpis u tohto protokolu je použitá `struct ip6_hdr`, získaná z paketového bufferu s pripočítaním veľkoti Ethernetovej hlavičky.

Pre výpis adries IPv6 podľa RFC5952 je implementovaná funkcia `print_ipv6`[5]. Výpis je realizovaný po dvojiciach, kde každá dvojica je od následujúcej oddelená dvojbodkou. Výnimka je pri výpise adries kde je za sebou viacero dvojíc takých že obe 8-bitové čísla sú rovné 0. Tu je potreba najdlhšiu postupnosť takýchto dvojíc vynechať.

Pri výpise IPv6 je preto potreba zistiť kde sa nachádza najdlhšia postupnosť nulových dvojíc pomocou funkcie `find_zeros`. Následne môžme vypísať adresu v požadovanom tvare, kde každá chcená dvojica je vypísaná vo funkcii `print_pair`. Týmto spôsobom sú vypísané zdrojové aj cieľové IPv6 adresy.

2.10 Výpis paketu

Výpis paketu je riešený vo funkcii `print_payload`, ktorá v cykle volá `print_payload_line` dokým nedojde na koniec paketu.

Každý riadok (okrem posledného) má 16 bajtov. Najprv nasleduje výpis offsetu, tj. pozícia v pakete. Postupnosť 16-tich bajtov s hexadecimálnym výpisom. Za nimi nasleduje výpis znakov. Overenie či je možné daný znak zobraziť je pomocou štandardnej funkcie `isprint`.

3 Testovanie a debugging

Pri testovaní a debuggingu bol pre simuáciu zariadenia použitý primárne Windows Subsystem for Linux s distribúciou Debian, ale taktiež aj Ubuntu 20.04 vo Virtualboxe. Výhoda použitia WSL bola, že paketov prechádzajúcich na tomto rozhraní bolo pomerne málo a tak bolo jednoduchšie zachytávať konkrétny paket.

Pre monitorovanie paketov bol použitý Wireshark na Windows systéme. Dosť bola nápomocná možnosť na zmenu výpisu času na požadovaný formát, či filtrovanie protokolov a portov.

Vytváranie ICMP paketov bolo riešené pomocou príkazu `ping` a ICMPv6 pomocou `ping6`. Porovnanie výpisu informácii o ICMPv6 pakete v ipk-snifferi a wiresharku môžeme vidieť nižšie.

```
g++ ipk-sniffer.cpp -o ipk-sniffer -lpcap
timestamp: 2022-04-24T20:54:23.254Z
src MAC: 00:15:5d:59:cd:7e
dst MAC: 33:33:ff:71:2c:cb
frame length: 86 bytes
protocol: ICMPv6
src IP:fe80::215:5dff:fe59:cd7e
dst IP:ff02::1001:ff71:2ccb

0x0000 33 33 ff 71 2c cb 00 15  5d 59 cd 7e 86 dd 60 00 33.q,... ]Y.~..`.
0x0010 00 00 00 20 3a ff fe 80  00 00 00 00 00 00 02 15 ... :... ........
0x0020 5d ff fe 59 cd 7e ff 02  00 00 00 00 00 00 00 00 ]..Y.~.. ........
0x0030 00 01 ff 71 2c cb 87 00  b4 64 00 00 00 00 fe 80 ...q,... .d......
0x0040 00 00 00 00 00 00 3c 90  a2 55 38 71 2c cb 01 01 ......<. .U8q,...
0x0050 00 15 5d 59 cd 7e                                ..]Y.~
```

Obr. 1: ICMPv6 paket v snifferi

```
Ethernet II, Src: Microsof_59:cd:7e (00:15:5d:59:cd:7e), Dst: IPv6mcast_ff:71:2c:cb (33:33:ff:71:2c:cb)
  > Destination: IPv6mcast_ff:71:2c:cb (33:33:ff:71:2c:cb)
  > Source: Microsof_59:cd:7e (00:15:5d:59:cd:7e)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fe80::215:5dff:fe59:cd7e, Dst: ff02::1:ff71:2ccb
    0110 .... = Version: 6
  > .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... 0000 0000 0000 0000 0000 = Flow Label: 0x00000
    Payload Length: 32
    Next Header: ICMPv6 (58)
    Hop Limit: 255
    Source Address: fe80::215:5dff:fe59:cd7e
    Destination Address: ff02::1:ff71:2ccb
    [Source SLAAC MAC: Microsof_59:cd:7e (00:15:5d:59:cd:7e)]
> Internet Control Message Protocol v6

0000  33 33 ff 71 2c cb 00 15  5d 59 cd 7e 86 dd 60 00   33·q,··· ]Y·~··`·
0010  00 00 00 20 3a ff fe 80  00 00 00 00 00 00 02 15   ···  :··· ········
0020  5d ff fe 59 cd 7e ff 02  00 00 00 00 00 00 00 00   ]··Y·~·· ········
0030  00 01 ff 71 2c cb 87 00  b4 64 00 00 00 00 fe 80   ···q,··· ·d······
0040  00 00 00 00 00 00 3c 90  a2 55 38 71 2c cb 01 01   ······<· ·U8q,···
0050  00 15 5d 59 cd 7e                                 ··]Y·~
```

Obr. 2: ICMPv6 paket vo Wiresharku

Pre vytváranie TCP a UDP paketov bol využívaný hlavne `packetsender`[2], ktorý umožňuje špecifikovať destináciu, port a jeden z protokolov. Niekoľko krát bol využitý aj nástroj `wget` s HTTP paketmi.

Literatura

[1] libpcap packet capture tutorial. Dostupné z: http://yuba.stanford.edu/~casado/pcap/section4.html

[2] Packet Sender. [online], [24. 4. 2022]. Dostupné z: https://packetsender.com/documentation

[3] Ethernet Frame Format. Říjen 2017. Dostupné z: https://www.geeksforgeeks.org/ethernet-frame-format/

[4] Carstens, T.: Programming with PCAP. Dostupné z: https://www.tcpdump.org/pcap.html

[5] Kawamura, S.; Kawashima, M.: A Recommendation for IPv6 Address Text Representation. RFC 5952, Srpen 2010, doi:10.17487/RFC5952. Dostupné z: https://www.rfc-editor.org/info/rfc5952

[6] Newman, C.; Klyne, G.: Date and Time on the Internet: Timestamps. RFC 3339, Červenec 2002, doi: 10.17487/RFC3339. Dostupné z: https://www.rfc-editor.org/info/rfc3339

[7] Wikipedia: IPv6 — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=IPv6&oldid=1080648470, 2022, [Online; accessed 24-April-2022].