VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

ANALYZÁTOR PAKETŮ

PACKET ANALYZER

PROJEKTOVÁ DOKUMENTACE

PROJECT DOCUMENTATION

AUTOR PRÁCE **PETR BUCHAL**

AUTHOR

BRNO 2017

Obsah

Úvod 1

1.1 Motivace 1

1.2 Cíl práce 1

Návrh Aplikace 2

2.1 L2 vrstva 2

2.2 L3 vrstva 2

2.3 L4 vrstva 3

2.4 Práce s daty 4

Popis implementace 5

3.1 Struktury 5

3.2 Implementace analyzátoru paketů 5

Návod na použití 6

4.1 Parametry programu 6

Závěr 7

5.1 Metriky kódu 7

Literatura 8

Kapitola 1

Úvod

V dnešní době prostupuje internet napříč téměř všemi sférami každodenního života a je tedy důležité, zabývat se tím, co sebou tato skutečnost nese. Běžného člověk většina věcí týkající se této záležitosti zajímat nemusí, stačí mu, aby to prostě fungovalo. Poté jsou zde ale lidé, jejichž prací je, zařídit právě, aby nebyla ohrožena efektivita, bezpečnost a obecně funkčnost internetu a potažmo všech sítí. Správa sítí je složitá disciplína, ale existuje spousta prostředků, které tuto činnost usnadňují. Každý informatik nicméně využívá ke své práci odlišné nástroje podle toho, které se na danou práci hodí nejlépe. Je tedy nápomocné, když tito lidé mají co největší počet kvalitních prostředků a mohou si vybrat opravdu ten, který je pro danou činnost nejlepší.

1.1 Motivace

Klíčový prvek sítí jako takových je skutečnost, že zařízení mezi sebou vzájemně komunikují. Posílají si zprávy různých typů a ty poté interpretují. Co ale když nastane nějaký problém v síti a je třeba zjistit jeho příčinu? Jednou z variant při pátrání po chybě je právě analýza paketů, tedy zpráv, které směřují přes síťovou kartu do zařízení a ze zařízení. Jenže bez nějakého podpůrného prostředku se jedná pouze o binární informace, které člověk nepřečte. Zde se nabízí možnost vytvoření programu, který bude binární formu transformovat do podoby, kterou nebude mít člověk problém přečíst.

1.2 Cíl práce

Vytvořený prostředek bude pracovat offline, tedy bude zpracovávat soubory se zachycenou síťovou komunikací. Vytvoření programu na analýzu paketů chci docílit s pomocí knihovny libpcap, která tento proces značně usnadňuje. Vytváří totiž nad daty jakousi abstraktní vrstvu s jejíž pomocí se s nimi lépe pracuje. Tato knihovna ovšem neusnadní problematiku sítí jako takových. Nachází se zde příliš mnoho protokolů, abych je zvládl ve školní práci analyzovat všechny, a tudíž dělám analýzu pouze několika základních protokolů. Z vrstvy síťového rozhraní analyzuji Ethernet a IEEE 802.1Q (včetně IEEE 802.1ad), z vrstvy síťové analyzuji IPv4, IPv6, ICMPv4 a ICMPv6 a z vrstvy transportní TCP a UDP. Uživatel programu si rovněž bude moci nechat agregovat, filtrovat nebo seřadit analyzovaná data podle různých parametrů.

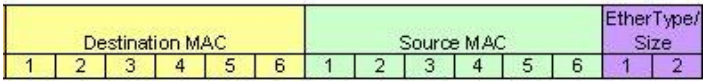
Kapitola 2

Návrh Aplikace

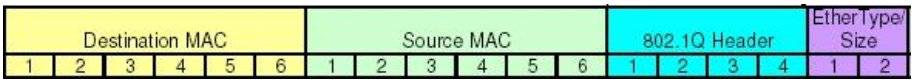
Program se zabývá analýzou několika síťových vrstev. Tato analýza probíhá postupně pro každou vrstvu zvlášť. Rozebereme si tedy, vrstvu po vrstvě, postup, jakým se analýza bude ubírat.

2.1 L2 vrstva

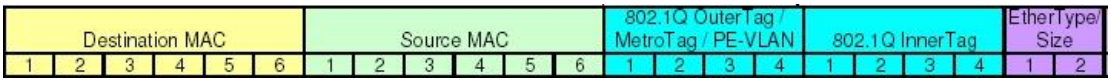
V L2 vrstvě se můžeme setkat se třemi typy hlaviček. Základní hlavička je typu Ethernet a obsahuje pole pro cílovou MAC adresu, zdrojovou MAC adresu a EtherType. Políčko EtherType nám pro naše účely říká, jaký protokol následuje, popřípadě zdali se jedná o jiný typ ethernetové hlavičky. Kód 0x8100 nám říká, že se bude jednat o hlavičku standartu IEEE 802.1Q a kód 0x88a8 indikuje hlavičku standartu IEEE 802.1ad (nejedná se tedy o ethernetovou hlavičku). V obou hlavičkách se vyskytuje EtherType na posledním místě (mezi ním a poli s MAC adresami jsou značky, které nás nezajímají). To znamená, že musíme identifikovat hlavičku, posunout se o daný počet bytů a získat hodnotu EtherType. Ten je totiž důležitý pro zjištění, zdali následující vrstva obsahuje IPv4 nebo IPv6 protokol.



Obrázek 1: Hlavička typu Ethernet Zdroj: https://upload.wikimedia.org/wikipedia/commons/f/f8/EthernetFrame.jpg



Obrázek 2: Hlavička typu IEEE 802.1Q Zdroj: https://upload.wikimedia.org/wikipedia/commons/2/23/TCPIP_802.1Q.jpg



Obrázek 3: Hlavička typu IEEE 802.1ad Zdroj: https://upload.wikimedia.org/wikipedia/commons/1/1b/TCPIP_802.1ad_DoubleTag.jpg

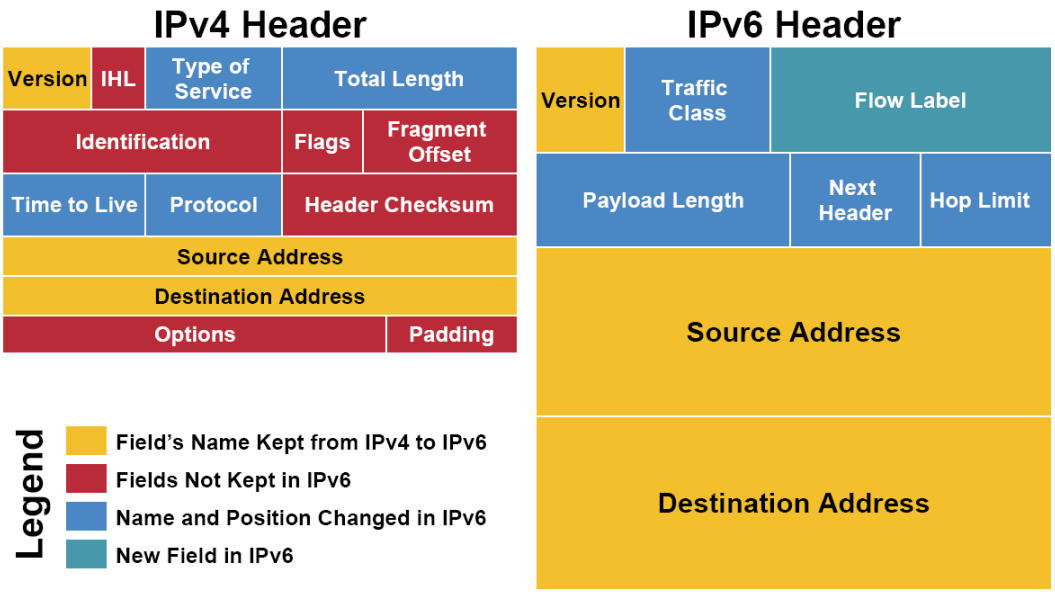
2.2 L3 vrstva

V L3 vrstvě se můžeme setkat se dvěma typy protokolů, konkrétně s IPv4 a IPv6. Ty jsou od sebe velmi odlišné a nesetkáme se tu jen s přidanými políčky jakožto změnou. Každý protokol je tedy nutné řešit zcela zvlášť.

Ještě než se začneme zabývat položkami hlaviček, je důležité zmínit se o samotných délkách hlaviček, protože ty jsou klíčové pro nalezení začátku hlavičky z vrstvy L4. V případě IPv4 je délka proměnná 40 – 60 bytů. V případě IPv6 je délka hlavičky fixních 40 bytů, ale tento protokol může obsahovat rozšiřující hlavičky a díky těm může být značně delší. Délka rozšiřujících hlaviček pro protokol IPv6 se určuje z jejich políčka délky. Konkrétně se k jeho hodnotě přičte 8 bytů a tohle číslo tvoří celkovou délku rozšiřující hlavičky. Rozšiřující hlavička osahuje pro nás ještě jednu důležitou informaci, a to kód následující hlavičky. Může se jednat buďto o kód hlavičky z následující L4 vrstvy nebo o kód další rozšiřující hlavičky.

Nyní se podíváme na pro nás důležité položky protokolu IPv4. Důležitým políčkem je IHL, z něj získáváme délku IPv4 hlavičky (po vynásobení čtyřmi). Položku identifikace bychom využili jako

část identifikace fragmentovaných paketů, ale fragmentaci jsem v tomto projektu nedělal, a tudíž ho více popisovat nebudu. Příznaky a offset fragmentu slouží rovněž pro práci s fragmentací. Další důležité políčko je TTL (time to live), značí kolik skoků může paket provést, než se zahodí. Tohle políčko je pro nás důležité, protože ho tiskneme. Poslední dvě důležitá políčka jsou pro nás zdrojová a cílová adresa IPv4, opět z důvodu tisku.
Z IPv6 hlavičky nás zajímají 4 políčka, konkrétně další hlavička, maximum skoků a zdrojová a cílová IPv6 adresa. Maximum skoků je IPv6 varianta TTL a stejně jako TTL ji tiskneme. O položce další hlavička jsme se již bavili výše. Jde o kód hlavičky, která následuje po současné.



Obrázek 4: Srovnání IPv4 a IPv6 Zdroj: https://i2.wp.com/www.ebrahma.com/wp-content/uploads/2013/12/ipv4-ipv6-header.gif

2.3 L4 vrstva

V L4 vrstvě řešíme čtyři různé protokoly. První dvojicí protokolů je TCP a UDP, které sebou většinou přenášejí data na aplikační vrstvě. Druhou kategorií jsou protokoly ICMPv4 (IPv4) a ICMPv6 (IPv6), které ve většině případů oznamují chyby v síti.
Jako první se podíváme na hlavičku protokolu TCP. Z té jsou pro nás důležité položky zdrojový a cílový port, číslo sekvence, potvrzený bajt a příznaky indikující různé situace. Ostatní věci jsou pro nás nedůležité stejně jako jakákoli data uložená v paketu. Analýza paketu zde končí.
V UDP hlavičce je pro nás důležitý jen zdrojový a cílový port, zbytek paketu opět zahazujeme a nezpracováváme.

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

Obrázek 5: Srovnání TCP a UDP hlavičky Zdroj: https://skminhaj.files.wordpress.com/2016/02/92926-tcp_udp_headers.jpg

ICMPv4 a ICMPv6 jsou protokoly, které nás informují o chybách, které nastaly při přenosu dat na síti. Pro nás důležité informace z těchto protokolů se nacházejí v obou hlavičkách hned na začátku, dokonce na stejných místech. Jedná se o typ a kód. Na základě těchto dvou polí se specifikuje, jaká chyba nastala. Nejdříve se chyba určí na základě typu a poté se specifikuje podle hodnoty kódu. Čísla specifikující jednotlivé chyby se pro ICMPv4 a ICMPv6 liší. V programu budeme obě dvě hodnoty tisknout a podle nich budeme vypisovat i definice chyb z RFC.

2.4 Práce s daty

Po proběhnutí analýzy je data buďto možné vytisknout anebo dále zpracovávat. První možností dalšího zpracování je filtrace dat, té uživatel docílí zadáním filtru vyhovujícího libpcap knihovně. Poté je data možné agregovat. Agregace je možná podle zdrojové a cílové MAC adresy nebo podle cílové a zdrojové IP adresy anebo podle cílového a zdrojového portu. Další možná manipulace s daty je řazení podle velikosti paketů nebo podle jejich počtu. Seřazení je vždy sestupné. A poslední možností je vypsání omezeného množství položek.

Kapitola 3

Popis implementace

Popis implementace programu rozdělím do dvou částí. První se bude zabývat strukturami a důvody jejich využití a druhá se bude zabývat implementací samotného analyzátoru.

3.1 Struktury

3.2 Implementace analyzátoru paketů

Kapitola 4

Návod na použití

Program je nutné spouštět alespoň s jedním analyzovatelným souborem (pokud nechceme vypsat nápovědu). Analyzovaný soubor musí být čitelný knihovnou libcap.

4.1 Parametry programu

-h — Vypíše nápovědu a ukončí program. Tento parametr nelze kombinovat s žádným jiným.

-a aggr-key — Zapnutí agregace podle klíče aggr-key, což může být *srcmac* značící zdrojovou MAC adresu, *dstmac* značící cílovou MAC adresu, *srcip* značící zdrojovou IP adresu, *dstip* značící cílovou IP adresu, *srcport* značící číslo zdrojového transportního portu nebo *dstport* značící číslo cílového transportního portu.

-s sort-key — Zapnutí řazení podle klíče sort-key, což může být packets (počet paketů) nebo bytes (počet bajtů). Řadit lze jak agregované, tak i neagregované položky. Řadí se vždy sestupně.

-l limit — Nezáporné celé číslo v desítkové soustavě udávající limit počtu vypsaných položek.

-f filter-expression — Program zpracuje pouze pakety, které vyhovují filtru danému řetězcem filter-expression.

file — Cesta k souboru ve formátu pcap (čitelný knihovnou libpcap). Možné je zadat jeden a více souborů.

Kapitola 5

Závěr

Program úspěšně provádí offline analýzu pcap souborů, ovšem u fragmentovaných paketů protokolu IPv4 vypíše pouze, že jsou fragmentované. Pro překlad slouží soubor Makefile, program je překládán překladačem g++. Prostředek byl otestován na referenčním serveru Merlin (CentOS/Linux).

5.1 Metriky kódu

Počet souborů: 1 soubor

Počet řádků zdrojového textu: 2588 řádků

Velikost spustitelného souboru: 145 160 bajtů

Literatura

[1] *Síťové aplikace a jejich architektura*. Brno: VUTIUM, 2014. ISBN 978-80-214-3766-1.