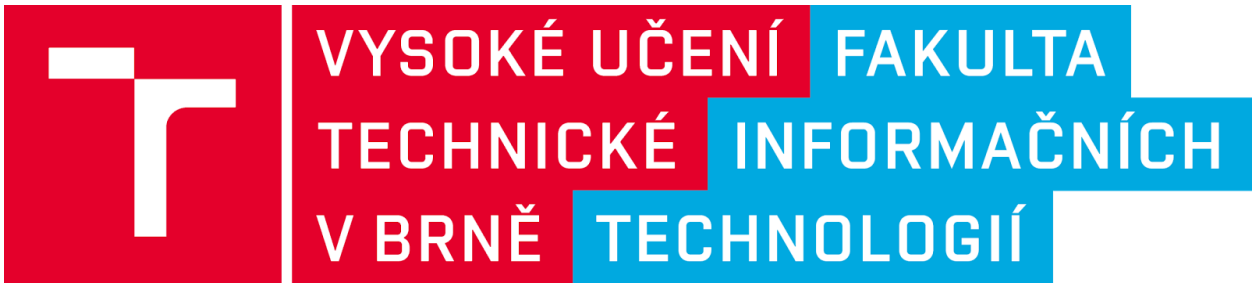
Vysoké Učení Technické v Brně



Analyzátor paketů

Dokumentace k projektu do předmětu ISA

Autor: Martin Bobčík, xbobci00
Datum: 19. Listopadu 2017

Obsah

Úvod

V tomto dokumentu budou postupně popsány jednotlivé použité protokoly a samotná implementace analýzy síťového provozu. Program funguje jako konzolová aplikace pro systém Linux.
V první kapitole je krátce nastíněno zadání. Druhá kapitola sebude zabývat jednotlivými vrstvami modelu TCP/IP a jejich protokoly. A ve třetí kapitole bude popsána implementace programu.

Klíčová slova: ethernet, sít, internet, ip, ipv6, linková vrstva, síťová vrstva, transportní vrstva, pcap, wireshark

Zadání

Úlohou tohoto projektu bylo vytvořit konzolovou aplikaci pro offline analýzu síťového provozu obsahující vybrané protokoly různých vrstev modelu TCP/IP. Bylo proto potřeba nastudovat zadané protokoly, jejich formát a obsah jejich hlaviček. Výsledný program bude číst pakety ze souboru a zobrazovat informace o nich na standardní výstup. Program dále umožňuje agregaci těchto informací.

Úvod do problematiky

Vrstva síťového rozhraní

Vrstva síťového rozhraní zajišťuje komunikaci mezi dvěma nebo více uzly propojenými stejným datovým spojem, jako například Ethernet, jejímž rámcům se věnuje následující podkapitola. Tato vrstva vytváří z informací z vyšších vrstev rámce, přidělí jim fyzickou MAC adresu a předá je fyzické vrstvě k odeslání.

Ethernetový rámec

Ethernetový rámec je datová jednotka vrstvy síťového rozhraní. Jeho hlavičce předchází preambule a oddělovač začátku rámce. V hlavičce pak najdeme cílovou a zdrojovou MAC adresu, a typ následujícího protokolu.

Bajtové odsazení	0-5B	6-11B	12-13B
0	Cílová adresa	Zdrojová adresa	Následující hlavička

IEEE 802.1Q a IEEE 802.1ad

Ethernetový rámec může volitelně obsahovat VLAN hlavičky, které umožňují rozdělit jednu ethernetovou síť na více logických sítí. Zatímco standard 802.1Q umoňuje vložit pouze jednu VLAN hlavičku, standard 802.1ad jich umožnujě vložit více.
Tyto hlavičky obsahují 2 bajtový VLAN identifikátor, a 2 bajtový typ dalšího protokolu, což může být opět 802.1ad, nebo protokol vyšší vrstvy.

Síťová vrstva

Síťová vrstva zajišťuje směrování a adresování systémů skrz jednu nebo více sítí. Tato vrstva příjme data od vyšších vrstev, zabalí je do paketu s cílovou adresou a předá je vrstvě síťového rozhraní.

IPv4

Internet Protocol směruje pakety z jednoho síťového rozhraní na jiné. Cesty těchto paketů jsou na sobě zcela nezávislé. Protokol nezaručuje správné doručení, jedná se tedy o nespolehlivou službu. Spolehlivost se tedy musí zajistit na vyšší vrstvě.
Adresování a směrování je zajištěno pomocí jednoznačných adres každého rozhraní, takzvaných IP adres. 4. Verze tohoto protokolu má adresy 32 bitové, které se standardně vypisují po bajtech v desítkové soustavě oddělené tečkou.

Bajtové odsazení	0		1	2	3
0-3	verze	velikost hlavičky	typ služby	celková délka	
4-7	identifikace			příznaky (3 bity)	offset fragmentu (13 bitů)
8-11	TTL		číslo protokolu	kontrolní součet hlavičky	
12-15	zdrojová adresa				
16-19	cílová adresa				
20 - ((VH * 4) - 1)	rozšířená nepovinná nastavení				
...	data				

Hlavička IPv4 obsahuje verzi internetového protokolu, velikost hlavičky v půlbajtech, typ služby, celkovou délku paketu v bytech, jednoznačný identifikátor paketu, příznaky a odsazení fragmentu.
Příznaky:

- Rezervovaný příznak - Musí být vždy nulový
- DF - pokud je nastaven, je zakázáno tento paket fragmentovat
- MF - pokud je nastaven, pak je tento paket fragmentovaný, a tento fragment není poslední

Odsazení fragmentu udává, na jaké pozici leží data tohoto fragmentu v původním paketu.
Dále hlavička obsahuje čas života paketu (TTL), číslo následujícího protokolu, kontrolní součet hlavičky a zdrojovou a cílovou adresu.

IPv6

Verze 6 internetového protokolu nahrazuje IPv4. IPv4 je nedostačující zejména kvůli, pro dnešní potřeby, malému adresnímu prostoru. IPv6 adresa má totiž 128 bitů, narozdíl od 32 bitů u IPv4. IPv6 adresa se standardně vypisuje po 4 hexadecimálních číslicích v dohromady osmi skupinách oddělených dvojtečkou. Pro zkrácení výpisu je možno vynechat úvodní nuly z každé skupiny a pokud adresa obsahuje několik, po sobě jdoucích nulových skupin, je možno místo nich vypsat pouze "::". Toto zkrácení je však možné využít jen jednou.

<table>
<tr><th>Bajtové odsazení</th><th colspan="2">0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>0–3</td><td>Verze</td><td>Třída provozu</td><td colspan="3">Značka toku</td></tr>
<tr><td>4–7</td><td colspan="3">Délka dat</td><td>Další hlavička</td><td>Max. skoků</td></tr>
<tr><td>8–11</td><td colspan="5" rowspan="4">Zdrojová adresa</td></tr>
<tr><td>12–15</td></tr>
<tr><td>16–19</td></tr>
<tr><td>20–23</td></tr>
<tr><td>24–27</td><td colspan="5" rowspan="4">Cílová adresa</td></tr>
<tr><td>28–31</td></tr>
<tr><td>32–35</td></tr>
<tr><td>36–39</td></tr>
</table>

Hlavička IPv6 obsahuje opět verzi internetového protokolu, třídu provozu určující prioritu paketu, značku toku pro správu kvality služeb (QoS), délku těla paketu, číslo následujícího protokolu, maximální počet skoků paketu a zdrojovou a cílovou adresu. Po adresách následují užitečná data.

IPv6 také podporuje rozšiřující hlavičky, které se vkládají mezi původní IPv6 hlavičku a hlavičku vyšší vrstvy. Existence takové hlavičky se udává v poli pro následující protokol. Každá taková hlavička obsahuje typ dalšího protokolu (eventuelně další rozšiřující hlavičky), délku hlavičky v bajtech (nezahrnující prvních 8 bajtů) a data specifické pro každý typ hlavičky.

ICMP

Internet Control Message Protocol využívají operační systémy k výměně služebních informací, jako například nedostupnost určité služby. Verze 4 se používá nad protokolem IPv4 a verze 6 nad IPv6. Ikdyž jsou tyto zprávy, stejně jako TCP a UDP, posílány nad IP protokolem, jsou součástí síťové vrstvy, protože nejsou přímo používány síťovými aplikacemi. ICMP zprávy jsou generovány na základě různých událostí.

Bitové odsazení	0-7b	8-15b	16-32b
0	Typ	Kód	Kontrolní součet
32	Tělo zprávy		

Hlavička ICMPv4 se skládá ze dvou 8 bitových čísel značící typ a kód ICMP zprávy. Kód upřesňuje typ zprávy. Následuje 16 bitový kontrolní součet. Zbytek hlavičky má různý význam pro různé typy zprávy.
Hlavička ICMPv6, stejně jako 4 verze, začíná dvěma 8 bitovými čísly a 16 bitovým kontrolním součtem. Konec hlavičky opět slouží k různým účelům, v závislosti na typu zprávy. Významy i hodnoty typů a kódů verze 6 se od verze 4 liší.

Transportní vrstva

Transportní vrstva umožňuje výměnu dat přímo mezi aplikacemi. Adresování služeb probíhá pomocí 16 bitových bez-znaménkových čísel portů. Každý port jednoznačně identifikuje službu na daném počítači. Tato vrstva se nestará o směrování.

TCP

Transmission Control Protocol zajišťuje spolehlivou, obousměrnou výměnu dat mezi službami. Spolehlivost přenosu dat je zajištěna určením sekvenčního a potvrzovacího čísla, které jsou dohodnuty při každém navázání nového spojení.
Spojení se navazujeve třech krocích.

1. Odesílatel pošle datagram s příznakem SYN, náhodně vygenerovaným sekvenčním číslem x a potvrzovacím číslem 0.
2. Příjemce pošle zpět datagram s příznaky SYN a ACK, potvrzovací číslo x=x+1 a náhodně vygenerované číslo sekvence y.
3. Odesílatel pošle datagram s příznakem ACK, číslo sekvence x=x+1 a číslo potvrzení y=y+1.

<table>
<tr><th>Bity</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td colspan="16">zdrojový port</td><td colspan="16">cílový port</td></tr>
<tr><td>32</td><td colspan="32">číslo sekvence</td></tr>
<tr><td>64</td><td colspan="32">potvrzený bajt</td></tr>
<tr><td>96</td><td colspan="4">offset dat</td><td colspan="6">rezervováno</td><td colspan="6">příznaky</td><td colspan="16">okénko</td></tr>
<tr><td>128</td><td colspan="16">kontrolní součet</td><td colspan="16">Urgent Pointer</td></tr>
<tr><td>160</td><td colspan="32">volby (volitelné)</td></tr>
<tr><td>192</td><td colspan="24">volby (pokračování)</td><td colspan="8">výplň (do 32)</td></tr>
<tr><td>224</td><td colspan="32">data</td></tr>
</table>

Hlavička TCP datagramu obsahuje zdrojový a cílový port v prvních 32 bitech, 32 bitů sekvenčního čísla a 32 bitů potvrzovacího čísla, 4 bitový ukazatel na začátek dat a 6 rezervovaných bitů pro budoucí použití. Následují příznaky, okno, kontrolní součet, ukazatel na urgentní data a volitelné volby, které mohou mít až 320 bitů (délka musí být dělitelná 32). Nakonec jsou samotná data datagramu.
Příznaky:

- NS
- CWR
- ECE
- URG - Udává, že je potřeba se řídit ukazatelem na urgentní data.
- ACK - Udává, že na místě pro potvrzovací číslo je nějaká hodnota. Všechny datagramy od navázání spojení by měly mít tento příznak nastaven.
- PSH - Nastavením tohoto bitu by měla být data přesunuta z vyrovnávací paměti příjmající aplikaci.
- RST - Nastavením tohoto bitu se resetuje spojení
- SYN - Synchronizace sekvenčních čísel. Tento příznak by se měl používat pouze při navázání spojení.
- FIN - Poslední datagram od odesílatele

UDP

User Datagram Protocol zajišťuje spojení dvou služeb bez záruky na správné doručení. Při přenosu pomocí UDP mohou datagramy dorazit v jiném pořadí, vícenásobně, nebo vůbec. Proto je tento protokol vhodný pro aplikace vyžadující jednoduchost a nezáleží jim na možných ztracených datech, jako je například služba DNS, VoIP, online hry, nebo přehrávání videa v reálném čase.

Bitové odsazení	0-15b	16-31b
0	Zdrojový port	Cílový port
32	Délka	Kontrolní součet
64	Data	

Hlavička UDP datagramu je, narozdíl od TCP hlavičky, mnohem jednodušší. Obsahuje volitelné číslo zdrojového portu, číslo cílového portu, celková délka datagramu a volitelný kontrolní součet. Za hlavičkou následují samotná data.

Popis řešení

Program byl implementován za pomoci výše uvedených znalostí a zdrojů uvedených na konci práce. Implementační jazyk je C++11

Ovládání programu

Program funguje jako konzolová aplikace. Při spuštění se špatnou kombinací parametrů, nebo bez nich se vypíše nápověda a automaticky se ukončí.
Jediným poviný parametr je platná cesta k *.pcap souboru, který obsahuje záznam síťové komunikace. Neexistující soubor, nebo soubor, který nelze otevřít vyústí v ukončení programu s návratovým kódem 1. Ukončení takovouto cestou nastává i v případě zadání několika souborů z něhož i pouze jeden je neexistující či nejde otevřít. Tímto je zabráněno špatnému pořadí paketů, do kterého by nebyly započítány pakety z neotevřených souborů (například kvůli nízkým uživatelským právům).
Jako nepovinné parametry lze zadat -h pro zobrazení nápovědy. Přepínačem -a aggr-key se síťová komunikace agreguje podle klíče aggr-key.
Dostupné agregační klíče:

- srcmac - Zdrojová MAC adresa
- dstmac - Cílová MAC adresa
- srcip - Zdrojoví IP adresa
- dstip - Cílová IP adresa
- srcport - Zdrojový port (pouze u TCP a UDP paketů)
- dstport - Cílový port (pouze u TCP a UDP paketů)

Pokud je zadán nepodporovaný agregační klíč, program končí s návratovým kódem 3.
Přepínač -s sort-key zajistí sestupné třídění výstupních záznamů podle klíče sort-key.
Dostupné klíče třídění

- packets - Třídění podle počtu paketů (pouze u agregace)
- bytes - třídění podle počtu bajtů paketu

Přepínač -l limit zajistí, že počet vypsaných záznamů je maximálně roven číslu v limit.
Zadáním přepínače -f filter-expression program zpracuje pouze pakety vyhovující filtru ve formátu pcap, známého z aplikace WireShark. Pokud řeťezec neodpovídá formátu pcap filtru, program končí s návratovým kódem 2.

Popis implementace

O zpracování parametrů či přepínačů programu se stará metoda ProcessParams, která využívá funkce getopt. Pro jednoduchou práci s hodnotami parametrů jsou všechny uloženy jako C++ řetězce ve struktuře Params. Seznam načítaných souborů je také uložen v této struktuře jako vektor řetězců. Tento způsob byl vybrán kvůli jednoduchosti práce s vektory a řetězci C++.

Po zpracování parametrů si program vytvoří škálu pomocných proměnných, které se používají téměř celou dobu běhu programu. Pro příklad můžeme jmenovat struktury hlaviček jednotlivých protokolů, či asociativní pole pro překlad čísel protokolů na jejich jména.
Program dále prochází soubor po souboru a v každém souboru projde všechny pakety. Informace o každém protokolu si uloží do struktury, ve které jsou pouze proměnné nutné pro výstupní záznamy. Každá vyplněná struktura je poté uložena do asociativního polevšech paketů. (Vektor by byl nejspíš vhodnější struktura, ale toho jsem si všiml pozdě. Navíc asociativní pole v ničem neomezovalo.)

Pro každý protokol je v linuxových knihovnách vytvořena nějaká struktura. Do této struktury se paket uloží jako ukazatel na začátek paketu (paket je uložen jako pole znaků) plus odsazení již zpracovaných paketů. Odsazení je většinou také rovno velikosti protokolů nižších vrstev. Velikost odsazení je tedy uložena v proměnné, a pokaždé se do ní přičítají aktuálně zpracovaná velikost.

Pro IEEE 802.1Q a 802.1ad ovšem struktury nejsou, takže informace z nich se dostávají posuvem ukazatele na začátek těchto hlaviček, a poté bitovými operacemi. Toto zajišťují metody getVlanIDFromXthlanHeader a getNextTypeFromXthVlanHeader, které získají identifikátor hlavičky a číslo dalšího protokolu z x-té hlavičky.

Pokud paket obsahuje neznámý protokol, paket se neuloží do mapy a vypíše se hláška na chybový výstup.

Agregaci zajišťuje metoda agregate, která projde všechny struktury paketů uložené v asociativním poli a uloží je do nového asociativního pole určeného pro agregaci. Klíčem tohoto pole je agregační klíč podle kterého se agreguje. Hodnota pole je struktura obsahující celkový počet paketů pro daný agregační klíč a celkový počet bajtů. Pokud metoda narazí na klíč, který v poli není, vytvoří ho, a vloží první hodnoty, pokud tam už je, hodnoty pouze přičte k již existujícímu klíči.

Pro třídění nebyl zvolen žádný obecně známý algoritmus, místo toho bylo využito datové struktury nabízené jazykem C++. Touto strukturou je asociativní pole s duplicitními klíči, multimap. Při vkládání prvků do tohoto pole jsou prvky automaticky tříděny podle klíče. Za klíč byla tedy zvolena hodnota, podle které má program třídit. Bylo tedy nutné jen vložit struktury

paketů či agregátů z jedné abstraktní datové struktury do jiné, a poté je postupně, odzadu přečíst a vytisknout.

Omezení

- Program nepodporuje fragmentované pakety IPv4 protokolu

Závěr

Program byl testován na školním serveru merlin. Byl bezproblému přeložitelný i spustitelný. Program dodržuje formát vstupních parametrů i výstupů. Z časové tísně nebyla dokončena podpora fragmentace u IPv4 paketů.

Použité zdroje

POSTEL, J. *INTERNET STANDARD: RFC 768 - User Datagram Protocol* [online]. USC/Information Sciences Institute, 1980 [cit. 2017-11-20]. Dostupné z: https://tools.ietf.org/html/rfc768. University of Southern California.

INTERNET STANDARD: RFC 793 - Transmission Control Protocol [online]. USC/Information Sciences Institute, 1981 [cit. 2017-11-20]. Dostupné z: https://tools.ietf.org/html/rfc793. University of Southern California.

POSTEL, J. *INTERNET STANDARD: RFC 792 - Internet control message protocol* [online]. USC/Information Sciences Institute, 1981 [cit. 2017-11-20]. Dostupné z: https://tools.ietf.org/html/rfc792. USC/Information Sciences Institute.

CONTA, Alex, Stephen DEERING a Mukesh GUPTA. *INTERNET STANDARD: RFC 4443 - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification* [online]. The Internet Society, 2006 [cit. 2017-11-20]. Dostupné z: https://tools.ietf.org/html/rfc4443

Transmission Control Protocol. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-11-20]. Dostupné z: https://cs.wikipedia.org/wiki/Transmission_Control_Protocol

Ethernetový rámec. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-11-20]. Dostupné z: https://cs.wikipedia.org/wiki/Ethernetov%C3%BD_r%C3%A1mec

IEEE 802.1aq. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-11-20]. Dostupné z: https://en.wikipedia.org/wiki/IEEE_802.1aq

IEEE 802.1Q. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-11-20]. Dostupné z: https://cs.wikipedia.org/wiki/IEEE_802.1Q

IPv4. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-11-20]. Dostupné z: https://cs.wikipedia.org/wiki/IPv4

INTERNET STANDARD: RFC 791 - Internet Protocol [online]. In: . Marina del Rey (CA): University of Southern California, 1981 [cit. 2017-11-20]. Dostupné z: https://tools.ietf.org/html/rfc791

IPv6. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-11-20]. Dostupné z: https://cs.wikipedia.org/wiki/IPv6

IPv6 packet. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-11-20]. Dostupné z: https://en.wikipedia.org/wiki/IPv6_packet