



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

GENEROVÁNÍ NESTANDARDNÍCH PAKETŮ A DATOVÝCH TOKŮ NA SÍTI ETHERNET PRO ÚČELY VÝUKY

GENERATING NON-STANDARD PACKETS AND DATA FLOWS ON ETHERNET NETWORKS FOR
TEACHING PURPOSES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tereza Mošnerová

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jan Jeřábek, Ph.D.

BRNO 2024



Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

Studentka: Bc. Tereza Mošnerová

ID: 210347

Ročník: 2

Akademický rok: 2023/24

NÁZEV TÉMATU:

Generování nestandardních paketů a datových toků na síti Ethernet pro účely výuky

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte z literatury problematiku fungování protokolů TCP/IPv4 a IPv6. Prostudujte také spektrum protokolů zahrnutých v předmětech BPC-KOM a MPC-PKT. Zaměřte se i na různé nástroje generování paketů v různých operačních systémech a zejména pak na program Ostinato, stejně tak jako na program Wireshark pro analýzu zachyceného síťového provozu. V rámci diplomové práce vytvořte návrh několika řešení sestav různých datových toků, které budou vhodné jako zadání samostatných projektů v předmětech BPC-KOM a MPC-PKT. Připravte řešení ve schváleném programu, které bude dané datové toky generovat, a především umožní snadno tato zadání i modifikovat. Kompletní specifikace musí být předem konzultována a schválena vedoucím práce. Vytvořené sestavy toků paketů je nutné vhodně okomentovat. Vytvořte ukázky možného použití. V rámci semestrálního projektu zpracujte teoretickou část. Dále se seznámte s potřebným software vybavením, připravte si testovací prostředí a vygenerujte první testovací sadu protokolů dle domluvy s vedoucím práce.

DOPORUČENÁ LITERATURA:

- [1] Kurose, J. F., Ross, K. W., Computer networking: a top-down approach. 8th global ed. Essex: Pearson, 2022, 820 s. ISBN 978-9356061316.
- [2] JEŘÁBEK, J. Pokročilé komunikační techniky. Skriptum FEKT Vysoké učení technické v Brně, 2023. s. 1-179.

Termín zadání: 5.2.2024

Termín odevzdání: 21.5.2024

Vedoucí práce: doc. Ing. Jan Jeřábek, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zabývá tvorbou nestandardních datových toků, které obsahují skrytou informaci. Pro generování takových paketů je využíván nástroj Ostinato, ve kterém lze volit protokoly na jednotlivých vrstvách síťového modelu TCP/IP dle preferencí, přidávat a libovolně upravovat jednotlivé bajty dat pomocí polí HexDump. Pro skrytí dat jsou využívána různá pole na vrstvě síťové, transportní i aplikační. Vygenerované varianty paketů jsou snadno modifikovatelné a mohou sloužit například jako zadání projektů pro předměty Komunikační technologie a Pokročilé komunikační techniky vyučované na VUT. Kromě návrhu datových toků jsou také testovány limitace velikostí vkládaných dat u jednotlivých možností. Funkčnost generovaných paketů je ověřována programem Wireshark. Výsledkem je celá řada souborů formátu *.ostm* a *.pcapng* včetně textového souboru s jejich přehledným popisem.

KLÍČOVÁ SLOVA

TCP/IP, Ethernet, generování nestandardních paketů, datový tok, skrytá informace, pole, DHCP, DNS, ICMP, MLD, QUIC, UDP, Ostinato, Wireshark

ABSTRACT

This diploma thesis is aimed at the process of generating non-standard data streams that contain hidden information. For this purpose a software called Ostinato is used. Ostinato allows to select protocols on the individual layers of the TCP/IP network model according to the needs and preferences, and moreover, data bytes can be inserted and modified arbitrarily using the HexDump fields. Various fields of the TCP/IP protocols are used to hide data on the network, transport and application layers. The generated data packets can be easily modified which enables to create several other versions of them. These can serve, for example, as a semestral project assignment for the subjects Communication Technology and Modern Communication Techniques taught at BUT. In addition, the size limits of the inserted data of the individual data packet options are tested. The functionality of the generated packets is verified by Wireshark. As a result, several *.ostm* and *.pcapng* files including a text file with their clear description are provided.

KEYWORDS

TCP/IP, Ethernet, non-standard packet generation, data flow, hidden information, field, DHCP, DNS, ICMP, MLD, QUIC, UDP, Ostinato, Wireshark

MOŠNEROVÁ, Tereza. *Generování nestandardních paketů a datových toků na síti Ethernet pro účely výuky*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. Vedoucí práce: doc. Ing. Jan Jeřábek, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Tereza Mošnerová
VUT ID autora:	210347
Typ práce:	Diplomová práce
Akademický rok:	2023/24
Téma závěrečné práce:	Generování nestandardních paketů a datových toků na síti Ethernet pro účely výuky

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno
.....
podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu diplomové práce panu doc. Ing. Janu Jeřábkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	14
1 TCP/IP	15
1.1 Vrstva síťového rozhraní	15
1.1.1 MAC adresy	16
1.1.2 Ethernet	16
1.1.3 ARP (Address Resolution Protocol)	17
1.2 Vrstva internetová (síťová)	18
1.2.1 IPv4 (Internet Protocol version 4)	19
1.2.2 IPv6 (Internet Protocol version 6)	20
1.2.3 ICMP (Internet Control Message Protocol)	21
1.2.4 ICMPv6 (Internet Control Message Protocol version 6)	22
1.2.5 Směrování	23
1.2.6 Multicast	23
1.3 Vrstva transportní	24
1.3.1 TCP (Transmission Control Protocol)	24
1.3.2 UDP (User Datagram Protocol)	25
1.4 Vrstva aplikační	26
1.4.1 DHCP (Dynamic Host Configuration Protocol)	26
1.4.2 DNS (Domain Name System)	27
1.4.3 mDNS (Multicast Domain Name System)	27
1.4.4 LLMNR (Link-Local Multicast Name Resolution)	28
1.4.5 HTTP (Hypertext Transfer Protocol)	28
1.5 Zapouzdřování a další způsoby přenosu dat	29
1.5.1 Zapouzdřování	29
1.5.2 Přenos dat skrytým kanálem	30
1.5.3 Steganografie	30
2 Nástroje pro generování paketů	31
2.1 Nástroje	31
2.2 Ostinato	34
2.2.1 Seznámení s uživatelským rozhráním Ostinato	35
2.2.2 Generování paketů v programu Ostinato	36
2.3 Srovnání nástrojů	41
2.4 Rešerše podobných prací	42

3	Vytvoření testovacího prostředí	43
3.1	VMware Workstation	43
3.2	Kali Linux	43
3.3	Wireshark	44
3.3.1	Seznámení s uživatelským rozhraním Wireshark	44
4	Návrhy nestandardních datových toků	46
4.1	DHCP – Unassigned option	46
4.2	DNS Response - TXT, RRSIG	53
4.3	MLD – Address Exclude	56
4.4	QUIC – Connection IDs	56
4.5	UDP – IPv6 Hop-by-Hop Option	57
4.6	Srovnání vytvořených nestandardních paketů	59
4.7	Kombinace možností skrývání informace	61
4.8	Zadání projektu pro předmět MPC-PKT	64
4.9	Shrnutí	67
5	Testování paketů na reálné síti	70
5.1	ICMPv6 ping - IPv6 Hop-by-Hop Option	70
5.2	DNS - IPv6 Fragment Header	71
5.3	DNS - Additional Records	72
5.4	mDNS - IPv6 Destination Options	72
5.5	LLMNR - IPv6 Routing Header	72
5.6	ICMPv6 ping - IPv6 Hop-by-Hop Option	73
5.7	UDP - IPv6 Authentication Header	74
5.8	Shrnutí	74
	Závěr	75
	Literatura	77
	Seznam symbolů a zkratk	81
	Seznam příloh	83
A	Datové provozy a vytvořené soubory	85
A.1	<i>bgp.ostm / bgp.pcapng</i>	85
A.2	<i>dhcp.ostm / dhcp.pcapng</i>	85
A.3	<i>dns.ostm / dns.pcapng</i>	85
A.4	<i>ftp.ostm / ftp.pcapng</i>	86
A.5	<i>icmpv6.ostm / icmpv6.pcapng</i>	86

A.6	<i>mdns.ostm /mdns.pcapng</i>	86
A.7	<i>mld.ostm / mld.pcapng</i>	86
A.8	<i>quic.ostm / quic.pcapng</i>	87
A.9	<i>tcp.ostm / tcp.pcapng</i>	87
A.10	<i>udp.ostm / udp.pcapng</i>	87
A.11	Popis souborů: <i>popis.docx</i>	87
B	Limitace velikostí variant pro skrytí dat	88
B.1	<i>auth_header.ostm / auth_header.pcapng</i>	88
B.2	<i>bgp_optional.ostm / bgp_optional.pcapng</i>	88
B.3	<i>dhcp_option.ostm / dhcp_option.pcapng</i>	88
B.4	<i>dns_add_records.ostm / dns_add_records.pcapng</i>	88
B.5	<i>dns_txt.ostm / dns_txt.pcapng</i>	88
B.6	<i>dns_rrsig.ostm / dns_rrsig.pcapng</i>	89
B.7	<i>dns_signature.ostm / dns_signature.pcapng</i>	89
B.8	<i>hop_by_hop_opt.ostm / hop_by_hop_opt.pcapng</i>	89
B.9	<i>mdns.ostm / mdns.pcapng</i>	89
B.10	<i>mld.ostm / mld.pcapng</i>	89
B.11	<i>quic_ids.ostm / quic_ids.pcapng</i>	89
B.12	<i>routing_header.ostm / routing_header.pcapng</i>	89
B.13	<i>tcp_option.ostm / tcp_option.pcapng</i>	90
C	Kombinace variant pro skrytí dat	91
C.1	<i>kombinace_ipv6.ostm / kombinace_ipv6.pcapng</i>	91
C.2	<i>komb_vrstvy.ostm / komb_vrstvy.pcapng</i>	91
C.3	Popis souborů: <i>popis_kombinace.docx</i>	91
D	Další verze projektu pro MPC-PKT	92
D.1	<i>dns.ostm / dns.pcapng</i>	92
D.2	<i>icmpv6_fragmenty.ostm / icmpv6_fragmenty.pcapng</i>	92
D.3	<i>mdns.ostm / mdns.pcapng</i>	92
D.4	<i>mld_v1.ostm / mld_v1.pcapng</i>	92
D.5	<i>mld_v2.ostm / mld_v2.pcapng</i>	93
D.6	<i>projekt_pkt.ostm / projekt_pkt.pcapng</i>	93
D.7	Popis souborů: <i>popis_projekt.docx</i>	93
E	Testování paketů na reálné LAN síti	94
E.1	<i>test_dns.pcapng</i>	94
E.2	<i>test_dns_ar.pcapng</i>	94
E.3	<i>test_echo_reply.pcapng</i>	94

E.4	<i>test_llmnr.pcapng</i>		94
E.5	<i>test_mdns.pcapng</i>		94
E.6	<i>test_ping.pcapng</i>		95
E.7	<i>test_udp.pcapng</i>		95

Seznam obrázků

1.1	Modely pro popis počítačových sítí [2]	15
1.2	Formát rámce Ethernet II [4]	17
1.3	Struktura ARP paketu [1]	18
1.4	Záhlaví IPv4 paketu [5]	19
1.5	Záhlaví IPv6 paketu [1]	20
1.6	Záhlaví ICMP paketu [9]	22
1.7	Záhlaví TCP segmentu [14]	25
1.8	Záhlaví UDP datagramu [3]	26
1.9	Proces zapouzdřování [11]	29
2.1	Ukázky práce s nástrojem Scapy	31
2.2	Odeslání deseti ICMP paketů pomocí nástroje hping3	32
2.3	Odeslání tří ICMP paketů pomocí nástroje Nping v režimu Echo	33
2.4	Uživatelské rozhraní nástroje Packet Sender	34
2.5	Hlavní okno nástroje Ostinato	35
2.6	Sestavení paketu (Ostinato)	36
2.7	Volba dalších parametrů (Ostinato)	37
2.8	Sekce Stream Control (Ostinato)	37
2.9	Záložka Packet View (Ostinato)	38
2.10	Spuštění odesílání paketů (Ostinato)	38
2.11	Přidání pole HexDump (Ostinato)	39
2.12	Vložení obsahu do pole HexDump (Ostinato)	40
2.13	Obsah pole HexDump (Ostinato)	40
3.1	Hlavní okno programu Wireshark	45
4.1	Simulovaná DHCP komunikace	47
4.2	DHCP Options zprávy DHCP Discover	47
4.3	DHCP Unassigned option (Wireshark)	48
4.4	DHCP Unassigned option (Ostinato)	49
4.5	Diagram DHCP komunikace (běžná vs. skrytá komunikace)	49
4.6	Informace skrytá ve zprávě DHCP Discover	50
4.7	Informace skrytá ve zprávě DHCP Offer	50
4.8	Informace skrytá ve zprávě DHCP Request	50
4.9	Informace skrytá ve zprávě DHCP ACK	51
4.10	Rovnice vložené do DHCP Offer zpráv	51
4.11	Jednotlivá pole HexDump v Ostinato	52
4.12	Tvorba Variable Fields v Ostinato	52
4.13	Pakety ilustrující DNS komunikaci	53
4.14	Informace skrytá v TXT poli	54

4.15	DNS Additional records – části pole TXT	54
4.16	Informace skrytá v RRSIG záznamu, v položce Signature	55
4.17	Informace skrytá v DNS RRSIG polích	55
4.18	Informace skrytá ve zprávě MLDv2 report	56
4.19	Informace skrytá v QUIC Connection a Source ID	57
4.20	Informace skrytá v záhlaví Hop-by-Hop Option v poli PadN	58
4.21	Informace skrytá v poli PadN a datové části protokolu UDP	58
4.22	Skrytí informace v několika IPv6 rozšiřujících záhlavích	61
4.23	Skrytí informace na 3 vrstvách modelu TCP/IP - část 1 (vrstva L3 a L4)	62
4.24	Skrytí informace na 3 vrstvách modelu TCP/IP - část 2 (vrstva L7)	63
4.25	Skrytí informace v rozšiřujícím IPv6 záhlaví	64
4.26	Skrytí informace v DNSSEC RRSIG	65
4.27	Fragmentované IPv6 pakety	65
4.28	Skrytí informace v datové části IPv6 paketu	66
4.29	Skrytí informace v mDNS odpovědi	66
4.30	Volba protokolů na jednotlivých vrstvách TCP/IP (Ostinato)	67
4.31	Hexadecimální reprezentace protokolu (Ostinato)	68
5.1	Zpráva Echo request se skrytou informací	70
5.2	Odpověď Echo reply	71
5.3	DNS dotaz se skrytou informací	71
5.4	DNS dotaz se záznamem Additional records	72
5.5	mDNS dotaz se skrytou informací	73
5.6	Odpověď na mDNS dotaz	73

Seznam tabulek

1.1	IPv6 adresy	21
2.1	Přehled nástrojů pro generování paketů	41
4.1	Výsledky ověřování rozsahu polí a podporovaná verze IP protokolu . .	59
4.2	Zjištěné a ověřené limitace délek dat vkládaných do postranního kanálu	60

Úvod

Komunikační protokoly síťového modelu TCP/IP jsou základem internetového provozu a celosvětového propojení zařízení. TCP (Transmission Control Protocol) je odpovědný za spolehlivý přenos dat, zatímco IP (Internet Protocol) se stará o adresaci a směrování dat. Protokoly modelu TCP/IP představují soubor pravidel, která specifikují, jak mezi sebou mají zařízení na internetu komunikovat, čímž vytvářejí standardizaci a umožňují komunikaci zařízení různých výrobců. Model TCP/IP hierarchicky strukturuje proces výměny dat do několika vrstev. Každá ze čtyř vrstev má specifické úlohy. Některé protokoly zajišťují integritu a doručení informací bez ztrát či poškození.

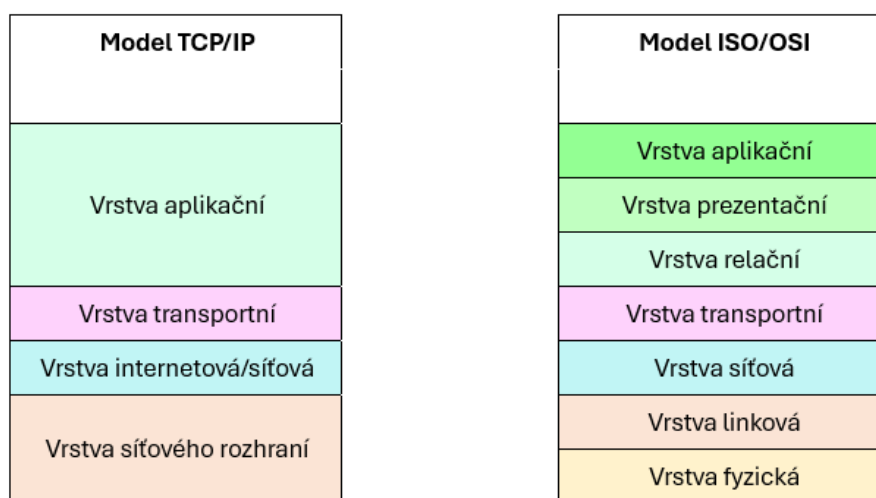
Výuka komunikačních protokolů je velmi důležitá, a to z toho důvodu, že znalost principů komunikačních protokolů umožňuje správně navrhovat, implementovat a bezpečně udržovat síťové infrastruktury. S rozvojem nových technologií vznikají nové požadavky na přenos dat a jejich zabezpečení vůči nově vznikajícím hrozbám. Nové aplikace mohou vyžadovat specifické funkce, či optimalizaci výkonu pro rychlejší výměnu dat, a proto se protokoly modelové sady TCP/IP neustále vyvíjejí. Je tedy klíčové komunikačním protokolům rozumět a sledovat jejich vývoj.

Cílem této diplomové práce je tvorba několika různých variant nestandardních datových toků v nichž je skrývána informace. Nestandardní pakety představují odchylky od obvyklých formátů používaných na síti Ethernet. Obecně mohou být tyto pakety záměrně vytvářeny za účelem testování, vývoje, analýzy bezpečnostních opatření, či pro účely kybernetických útoků. V případě této diplomové práce mají vygenerované datové toky sloužit jako variabilní základ pro sestavení semestrálního projektu zadávaného studentům bakalářského předmětu Komunikační technologie (BPC-KOM) či magisterského předmětu Pokročilé komunikační techniky (MPC-PKT), které je s komunikačními protokoly seznamují.

Diplomová práce je rozdělena do šesti kapitol. První kapitola je teoretickým úvodem, který popisuje základní komunikační protokoly síťového modelu TCP/IPv4 a IPv6 a jejich funkce. V kapitole druhé jsou představeny některé z nástrojů pro generování paketů, například Scapy, Hping a další a též jsou prezentovány informace získané krátkou rešerší prací zabývajících se obdobnou problematikou. Ve třetí kapitole jsou specifikovány softwary využité pro simulaci testovacího prostředí, tvorbu a analýzu vytvářených paketů, například VMware, Wireshark a další. Čtvrtá kapitola obsahuje hlavní praktickou část, v níž je podrobně popsán proces generování paketů za využití nástroje Ostinato a Wireshark a také výsledky testování limitací vytvořených variant. Pátá kapitola se krátce zabývá testováním vytvořených nestandardních paketů na reálné síti. A poté následuje závěr, který shrnuje cíle práce a výsledky v ní dosažené.

1 TCP/IP

Počítačové sítě jsou běžně popisovány dvěma referenčními modely znázorněnými na obrázku 1.1. Modelem ISO/OSI, který má sedm vrstev a modelem TCP/IP, jež má vrstvy čtyři. Hlavním rozdílem mezi modely ISO/OSI a TCP/IP je, že dle modelu ISO/OSI musí být spolehlivost komunikace do jisté míry zajišťována na každé vrstvě, čímž komunikační síť ztrácí část své přenosové kapacity, zatímco u TCP/IP je spolehlivost řešena až na vrstvě transportní. ISO/OSI je spíše konceptuální model, TCP/IP se na druhou stranu používá pro řízení komunikace zařízení na síti [1].



Obr. 1.1: Modely pro popis počítačových sítí [2]

TCP/IP je soustava protokolů, které definují, jak by měly fungovat počítačové sítě, tzn. jak má vypadat odesílání a přijímání informací mezi zařízeními na internetu [2]. Model TCP/IP je založen na dvou základních a velmi důležitých protokolech – TCP (Transmission Control Protocol) a IP (Internet Protocol) a skládá se ze čtyř vrstev. Každá z těchto vrstev splňuje určité funkce nutné pro fungování přenosu dat mezi zařízeními. Jedná se o vrstvu síťového rozhraní, vrstvu internetovou, transportní a aplikační. Pro každou z těchto vrstev jsou klíčové příslušné protokoly. Komunikační protokoly specifikují formát dat posílaných po internetu a zároveň kontrolují veškerou aktivitu na síti.

1.1 Vrstva síťového rozhraní

Vrstva síťového rozhraní je nejnižší vrstvou modelu TCP/IP a spojuje funkce vrstvy fyzické a spojové [3]. Tato vrstva zajišťuje veškeré úkony spojené s řízením konkrétního datového toku a přímým přenosem datových paketů. Na této vrstvě je umožněn

přístup k fyzickému přenosovému médiu, jako je Ethernet či Wi-Fi a řízení datového spoje, nicméně ji dále nelze blíže specifikovat, protože je závislá na přenosové technologii, kterou daná síť používá.

Vrstva síťového rozhraní přijímá data od vyšších vrstev a dělí je na menší části nazývané datové rámce [3]. Datové rámce obsahují informace o zdrojové a cílové MAC adrese a další kontrolní informace. Datové rámce poté předává na fyzickou vrstvu, která se stará o jejich fyzický přenos na síti po síťových linkách z jednoho uzlu do druhého.

Na této vrstvě také dochází k řízení přístupu ke sdílenému médiu [2]. Proces řízení přístupu zahrnuje metody, které podporují komunikaci několika zařízení na stejném fyzickém kanálu bez kolizí. Je zajišťována i detekce a korekce chybových datagramů. Pro detekci chyb je využíván kontrolní součet CRC (Cyclic Redundancy Check), či FCS (Frame Check Sequence). Pomocí jednoho z těchto algoritmů vypočítá odesíla- jící strana kontrolní součet, který připojí k odesílaným datům. Příjemce poté pomocí stejného algoritmu zjistí, zda data byla doručena bezchybně. V případě chybného doručení, tedy pokud kontrolní součet nesedí, je datagram zahozen.

1.1.1 MAC adresy

Vrstva síťového rozhraní pracuje s fyzickými adresami zařízení zvanými MAC (Media Access Control) adresy [3]. MAC adresa je jedinečný identifikátor každého síťového zařízení, které komunikuje v počítačové síti. Je také základní součástí síťového proto- kolu Ethernet a hraje klíčovou roli v přenosu dat v rámci lokální sítě. Skládá se ze 48 bitů (6 oktetů) zapsaných v hexadecimálním formátu. První tři oktety označují vý- robce síťového rozhraní, zatímco poslední 3 oktety představují unikátní identifikátor síťového rozhraní.

Správné použití MAC adres je klíčové pro bezpečné a efektivní fungování sítě. Efektivita je docílena za pomoci switchů, které si udržují adresní tabulky mapující MAC adresy na porty, na kterých jsou připojena zařízení a zajišťují tak efektivní směrování. MAC adresy mohou být také použity pro filtrování provozu, tzn. lze nastavit pravidla, která umožní či zakázají komunikaci určitých zařízení na základě jejich MAC adres. Omezením přístupu neoprávněným zařízením je zároveň zvýšena bezpečnost.

1.1.2 Ethernet

Ethernet je jednou z nejběžnějších a nejdůležitějších síťových technologií vyvinutou za účelem přenosu dat v rámci LAN (Local Area Network). Prvním standardem

pro Ethernet byl IEEE 802.3. Pro připojení k Ethernetu se původně využíval koaxiální kabel s topologií hvězdy, kde všechna zařízení byla připojena k jednomu centrálnímu kabelu. Dnes se častěji používá topologie hvězdy s aktivními síťovými prvky, jako jsou switche a routery [3]. Postupem času se Ethernet stal klíčovou technologií pro široké spektrum sítí, včetně WAN (Wide Area Network), MAN (Metropolitan Area Network) a významných datových center. Došlo také ke vzniku bezdrátového Ethernetu, známého jako Wi-Fi, který umožňuje mobilní komunikaci a přístup k internetu.

Data přenášená v síti Ethernet jsou organizována do rámců (frames) [3]. Každý rámec obsahuje informace o zdrojové a cílové MAC adrese, typu dat a samotná data. Rámce jsou opatřeny kontrolním součtem FCS pro detekci chyb. Dva nejčastěji používané formáty rámce jsou Ethernet II a IEEE 802.3 Ethernet. Jejich struktury vyznačují minimální rozdíly. Formát rámce Ethernet II je zobrazena na obrázku 1.2.

Preamble	Cílová adresa	Zdrojová adresa	Typ	Data	FCS
8 B	6 B	6 B	2 B	46 – 1500 B	4 B

Obr. 1.2: Formát rámce Ethernet II [4]

Ethernet podporuje různé rychlosti přenosu dat, včetně 10 Mbps (Ethernet), 100 Mbps (Fast Ethernet), 1 Gbps (Gigabit Ethernet) a rychlejší.

1.1.3 ARP (Address Resolution Protocol)

Mezi hlavní protokoly vrstvy síťového rozhraní patří především protokoly sloužící k zjištění IP adresy. Jedním z nich je protokol ARP (Address Resolution Protocol), který slouží k získání linkové adresy síťového rozhraní protistrany, a to na základě již známé IP adresy [2]. Například v síti Ethernet je na základě již známé IP adresy zjišťována adresa MAC. V opačném případě, tedy při získávání IP adresy na základě znalosti MAC adresy, hovoříme o IARP (Inverse ARP).

Pokud chce zařízení zjistit MAC adresu jiného zařízení na stejném segmentu sítě, odešle broadcastově ARP paket s dotazem, ve kterém se ptá, jaká je MAC adresa zařízení s konkrétní IPv4 adresou. Na tento dotaz odpovídá pouze to zařízení, kterému daná IPv4 adresa přísluší a v odpovědi poskytuje svou MAC adresu. Pokud zařízení hledá MAC adresu zařízení v jiné síti, pak je mu poskytnuta MAC adresa výchozí brány (default gateway). Získané informace si zařízení ukládá v ARP tabulce. Jednoduchá struktura ARP paketu je znázorněna na obrázku 1.3.

Typ média		Typ protokolu
Délka fyzické adresy	Délka logické adresy	Operace
Fyzická zdrojová adresa		
Logická zdrojová adresa		
Hledaná fyzická adresa		
Hledaná logická adresa		

Obr. 1.3: Struktura ARP paketu [1]

1.2 Vrstva internetová (síťová)

Internetová vrstva je druhou vrstvou modelu TCP/IP a hraje klíčovou roli v síťové komunikaci [1]. Je nezávislá na konkrétní přenosové technologii a zajišťuje směrování mezi dílčími sítěmi, tzn. zajišťuje dopravení dat, v tomto případě IP paketů, od odesílatele až k příjemci v rámci internetu. To obnáší směrování paketů přes různé sítě, které mohou používat odlišné technologie a protokoly. Je proto také úkolem této vrstvy vyrovnat se s rozdílným charakterem adres, či s různou maximální velikostí přenášených paketů.

Nejběžnějším protokolem používaným na internetové vrstvě je IP (Internet Protocol). Existuje několik verzí IP a mezi nejznámější jednoznačně patří protokoly IPv4 (Internet Protocol version 4) a IPv6 (Internet Protocol version 6). IPv4 je paketově orientovaný protokol, který zařízení poskytuje globální unikátní identifikátor (IP adresu). IP adresa je klíčovým prvkem internetové vrstvy. Každé zařízení připojené k internetu má jedinečnou IP adresu, která slouží k identifikaci tohoto zařízení v rámci celé sítě. IPv6 je následníkem a vylepšením protokolu IPv4. Dalšími důležitými protokoly v této vrstvě jsou například ICMP (Internet Control Message Protocol), který se používá pro správu a diagnostiku sítě a IGMP (Internet Group Management Protocol), jenž je důležitý pro správu multicastového provozu, který je popsán v kapitole 1.2.6.

Internetová vrstva se také stará o fragmentaci, neboli rozdělení, datových paketů na menší části pokud jsou příliš velké pro přenos přes síťové médium. Na straně příjemce jsou tyto fragmenty sestaveny zpět do původního paketu.

Internetová vrstva umožňuje implementaci služby QoS (Quality of Service), která určuje prioritu a kvalitu služby pro různé typy datového provozu. Tím pomáhá optimalizovat síťový provoz pro různé aplikace, jako jsou hlasová a videokomunikace, streamování dat nebo klasický datový provoz.

1.2.1 IPv4 (Internet Protocol version 4)

IPv4 je jedna z hlavních verzí protokolů IP a používá se pro adresaci a směrování datových paketů v počítačových sítích. Pro tento účel používá 32-bitové adresy, které identifikují jednotlivá zařízení připojená k síti a umožňují jim tak komunikovat mezi sebou [3].

Základní datovou jednotkou je IP datagram neboli paket, jehož záhlaví se skládá z několika polí, z nichž většina je neměnná [5]. Tento paket je na úrovni síťového rozhraní dále zapouzdřen do rámce příslušné technologie, jež se mění s přechodem paketu přes síť. Základní struktura IPv4 záhlaví paketu je znázorněna na obrázku 1.4.

Bit 0-3	Bit 4-7	Bit 8-15	Bit 16-18	Bit 19-31
IP verze	Délka záhlaví	Typ služby	Celková délka	
Identifikace			Příznaky	Posunutí fragmentu od počátku
Doba života TTL	Protokol		Kontrolní součet	
Zdrojová IPv4 adresa				
Cílová IPv4 adresa				
Volitelné položky záhlaví				
Přenášená data				

Obr. 1.4: Záhlaví IPv4 paketu [5]

Hlavními položkami záhlaví jsou například **Délka záhlaví**, která je proměnná a její maximální velikost může být 60 bajtů, **Typ služby**, jenž slouží pro specifikování požadované kvality přenosu, či **Celková délka** IP datagramu dosahující maximálně 65 536 bajtů. Maximální velikost paketu je dána hodnotou MTU (Maximum Transmission Unit), která je v každé síti rozdílná. Běžně se tedy stává, že při přechodu z jedné sítě do druhé se hodnota MTU mění a musí se tedy změnit i velikost přenášeného paketu. Pro změnu velikosti se používá fragmentace.

Mezi hlavní **Příznaky** paketu patří příznak *DF-bit* (*don't fragment*), který fragmentaci zakazuje, či *MF-bit* (*more fragments*), který naopak upozorňuje na to, že byl datagram fragmentován, a že za již přijatým fragmentem bude následovat další. **Doba života** datagramu neboli TTL (Time to Live) určuje počet skoků (hops), které může paket na své cestě do cíle provést. TTL je směrovačem vždy sníženo o 1 až do doby, kdy hodnotu dále není možné snižovat a paket je zahozen. **Kontrolní součet** záhlaví datagramu se používá ke kontrole, zda došlo k chybě [5].

1.2.2 IPv6 (Internet Protocol version 6)

IPv6 je následovníkem a vylepšením IPv4. S nárůstem počtu zařízení připojených k internetu došlo k vyčerpání adresního prostoru IPv4, a byla proto vyvinuta nová verze IP protokolu – IPv6. IPv6 je nekompatibilní s IPv4, avšak existuje několik mechanismů pro komunikaci mezi zařízeními IPv4 a IPv6, díky kterým může k postupnému přechodu z IPv4 na IPv6 docházet [3]. Patří mezi ně například **souběh protokolů**, kdy IPv4 a IPv6 fungují jako dva nezávislé protokoly, **jednoduché tunelování**, které spočívá v zapouzdření IPv6 paketu do IPv4 záhlaví či naopak, **zjednodušený souběh protokolů**, či **překlad protokolů**, ve kterém dochází k překladu mezi protokoly [1].

IPv6 datagramy

Struktura IPv6 datagramu je velmi podobná paketu IPv4, nicméně byly provedeny změny jež vedly ke zvýšení efektivity přenosu paketu a snížení složitosti [6]. Hlavička IPv6 datagramu, zobrazená na obrázku 1.5, se skládá z jednoho pevného záhlaví. Jeho velikost je na rozdíl od proměnné velikosti 20 až 60 B IPv4 paketu fixní, a to 40 B. Dále již není zahrnuto pole pro fragmentaci, kontrolní součet, informace o délce záhlaví a rozšiřující volby. Kontrolní součet zpomaluje proces směrování, a proto ho IPv6 neimplementuje a využívá se pouze standardní kontrola na linkové vrstvě.

Bit 0-3	Bit 4-7	Bit 8-11	Bit 12-15	Bit 16-19	Bit 20-23	Bit 24-27	Bit 28-31
IP verze	Provozní třída		Identifikace				
Celková délka přenášených dat				Další záhlaví		Limit počtu skoků	
Zdrojová IPv6 adresa							
Cílová IPv6 adresa							
Přenášená data							

Obr. 1.5: Záhlaví IPv6 paketu [1]

IPv6 bylo navrženo tak, aby bylo jednodušší než IPv4 [6]. Jedním z možných zjednodušení je právě omezení fragmentace. Místo fragmentace IPv6 stanovuje, že má každý směrovač na trase komunikovat MTU své sítě. Klienti by měli odesílat takové pakety, které se do této MTU vejdou. IPv6 předpokládá, že většina sítí zvládne MTU 1280 B. Ethernet využívaný na lokální síti zvládá pakety do délky 1500 B.

Řada položek IPv6 záhlaví se shoduje s položkami záhlaví IPv4. Pro identifikaci **Verze** jsou vyčleněny 4 bity, **Třída provozu** slouží pro nastavení priority paketu.

Identifikace toku dat označuje související pakety stejnou hodnotou, což směrovači umožňuje pakety odeslat stejnou cestou a zjednodušuje tak proces směrování. Také je zahrnuta informace **Celková délka přenášených dat**, **Další záhlaví**, jenž je volitelné a následuje za IPv6 adresami, **Limit počtu skoků**, který odpovídá hodnotě TTL u IPv4 a **IPv6 adresa odesílatele a příjemce datagramu**.

IPv6 adresy

IPv6 používá 128-bitové adresy, což je oproti 32-bitovým adresám IPv4 obrovský nárůst, a tudíž IPv6 poskytuje velmi rozsáhlý adresní prostor [7]. Díky tomu může mít každé zařízení jedinečnou IPv6 adresu, což usnadňuje jeho identifikaci a adresaci na síti. To je jeden z důvodů, proč dochází k postupnému přechodu na IPv6. IPv6 adresní prostor je navíc hierarchicky uspořádaný, a tudíž umožňuje rozdělení IP bloků a také efektivnější správu.

128-bitová IPv6 adresa se zapisuje hexadecimálně. Jednotlivé 16-ti bitové bloky jsou odděleny dvojtečkami. Jelikož je IPv6 adresa poměrně rozsáhlá, umožňuje IPv6 redundanci neboli zkracování nulových částí adresy. Pokud je některá z částí adresy tvořena pouze nulami, lze ji zapsat pouze jednou nulou, nebo blok plně vynechat a nahradit ho dvojtečkou. Tento proces zjednodušení je znázorněn v následující tabulce 1.1.

Základní verze	2001:067c:1220:08a1:0000:0000:0000:2628
Zkrácená verze	2001:67c:1220:8a1:0000:0000:0000:2628
Verze bez nul	2001:67c:1220:8a1::2628

Tab. 1.1: Proces zjednodušení IPv6 adresy [7]

IPv6 adresa má 3 části: site prefix, subnet ID, interface ID [8]. *Site prefix* je první část adresy běžně dlouhá 48 bitů, přidělená poskytovatelem internetových služeb. Zařízení se stejným umístěním budou sdílet i stejnou předponu. *Subnet ID* identifikuje administrační podsít a má délku 16 bitů. Nejdelší částí IPv6 adresy je *Interface ID*, které se používá k identifikaci IPv6 rozhraní v podsíti. Toto ID musí být pro zařízení v dané podsíti unikátní a má délku 64 bitů.

1.2.3 ICMP (Internet Control Message Protocol)

Protokol ICMP je protokol chybových hlášení, který slouží k řízení sítě, tzn. označuje chybové stavy, ověřuje dostupnost, síťové zdroje a podobně [9]. ICMP se také používá k provádění diagnostiky sítě pomocí nástroje *Traceroute* či *Ping*. Nástroj *Traceroute* se používá pro monitorování trasy mezi dvěma zařízeními připojenými k

internetu. Tento nástroj zjišťuje cestu z jednoho směrovače do druhého a před přenosem dat kontroluje, zda se na síti nevyskytují nějaké problémy. *Ping*, neboli *Echo request* zpráva je využíván pro zjištění času, který uplyne od odeslání dat, po jejich dosažení cíle a návrat ke zdroji.

Protokol ICMP nenavazuje spojení, nepřenáší žádná uživatelská data a jeho obsah je přenášen přímo v IP datagramech. ICMP paket znázorněný na obrázku 1.6 se skládá z IP záhlaví a ICMP dat [2].

Bit 0-7	Bit 8-15	Bit 16-32
Typ	Kód	Kontrolní součet
Rozšířené záhlaví		
Data (Payload)		

Obr. 1.6: Záhlaví ICMP paketu [9]

Pole **Typ** specifikuje typ ICMP zprávy, tzn. zda se jedná o zprávu chybovou, kontrolní, či informační [9]. Mezi nejznámější zprávy patří *8 – Echo request (Žádost o echo)* a *0 – Echo reply (Echo)*, které slouží pro ověření existence spojení mezi koncovými zařízeními. Poté zpráva *3 – Destination unreachable (Nedoručitelný datagram)*, která informuje o nedosažitelnosti cíle, *5 – Redirect message (Změň směrování)* informující o tom, že je k dispozici lepší cesta do cíle, či *11 – Time exceeded (Čas vypršel)*, identifikující, že hodnota TTL je na nule a další. Pole **Kód** nese dodatečné informace o ICMP zprávě. Následuje pole **Kontrolní součet** verifikující platnost ICMP záhlaví, **Rozšiřující záhlaví**, a poté ICMP **Data** proměnné délky.

1.2.4 ICMPv6 (Internet Control Message Protocol version 6)

ICMPv6 je obdobou ICMP protokolu, nicméně byl navržený přímo pro IPv6. Tento protokol chybových hlášení a správy sítě implementuje narozdíl od ICMP i několik dalších základních funkcí, mezi které patří například *Neighbor Discovery ND (Objevování sousedů)*, jež je využívána pro zjištění přítomnosti dalších uzlů, jejich adres na linkové vrstvě a dále pro zjištění výskytu aktivních směrovačů [10].

Neighbour Discovery je obdobou protokolu ARP používaného v IPv4, a slouží k určení MAC adresy zařízení v lokální síti na základě znalosti jeho IPv6 adresy. Využívá 5 typů ICMPv6 paketů: *Router Solicitation (RS)*, *Router Advertisement (RA)*, *Neighbor Solicitation (NS)*, *Neighbor Advertisement (NA)* a *Redirect* [10]. Zpráva *RS* je zasílána hostem v případě, že žádá směrovače o okamžité zaslání zprávy *RA*. Zprávou *RA* směrovač periodicky oznamuje svou přítomnost na síti a síťové parametry jako je MTU nebo hop limit. Zprávu *NS* zasílá uzel za účelem zjištění linkové

adresy souseda nebo ověření jeho dostupnosti. Odpovědí je zpráva *NA*, ve které uzel poskytuje svou adresu. Směrovače používají zprávu *Redirect*, kterou informují hosty o vhodnější cestě do cíle. Stejně jako ICMP používá i ICMPv6 zprávy typu *Echo request*, *Echo reply* a řadu dalších.

1.2.5 Směrování

Směrování určuje cestu dat z jednoho bodu do druhého v rámci počítačových sítí. Směrovače mají za úkol zvolit optimální cestu (route) z jednoho koncového bodu do druhého. Každý směrovač do jisté míry zná globální topologii, a na základě této znalosti lokálně rozhoduje, kam paket směrovat. Základním problémem směrování však je, že je globální topologie velmi rozsáhlá a dynamická a je tudíž složité o ní veškeré informace shromažďovat [11].

Pro správné směrování je třeba znát: (IP) adresu adresáta, všechny možné cesty do vzdálených sítí, aktuálně zvolenou cestu do cíle, sousední směrovače, způsob, kterým lze získávat informace o routách a jak tyto informace udržovat a aktualizovat. Pro tento účel jsou využívány směrovací tabulky [11].

Směrování probíhá na více úrovních, v rámci jednotlivých autonomních systémů (AS), i mezi nimi [2]. Směrování lze provádět staticky, kdy jsou směrovací informace nakonfigurovány ručně administrátorem, či dynamicky, přičemž směrování probíhá automaticky za pomoci směrovacích protokolů jako je OSPF (Open Shortest Path First), BGP (Border Gateway Protocol) apod. Na směrovací protokoly je kladeno několik základních požadavků: minimalizace velikosti směrovacích tabulek, minimalizace počtu přenášovaných kontrolních zpráv, robustnost a využívání optimálních tras.

1.2.6 Multicast

Multicast je typ skupinové komunikace, při které jsou data doručována několika příjemcům současně [12]. Může docházet k přenosu dat z jednoho zařízení na více, též zvané *one-to-many*, či od vícero odesílatelů několika příjemcům, jinak nazváno jako *many-to-many*. Vícesměrové vysílání je využíváno při videokonferencích, podnikové komunikaci, či distribuci softwaru.

IP multicast efektivně šetří šířku pásma tím, že odesílatel vysílá pouze jeden paket, který je po cestě sítí replikován směrovači a přepínači, které podporují vícesměrové vysílání [12]. Směrovače fungují jako stromy, tzn. rozvětví se tak, aby byla data doručena příjemcům co nejkratší cestou. Cesta paketu od zdroje k cíli se nazývá distribuční strom. Tvorbu těchto stromů zajišťuje protokol PIM (Protocol Independent Multicast). Zařízení, která mají zájem přijímat určitý datový tok zasílaný multicastově se mohou nacházet kdekoli na síti.

V rámci IPv4 se zařízení k multicastové skupině připojují, či se od ní odpojují, pomocí protokolu IGMP [12]. Pokud má zařízení zájem připojit se do konkrétní multicastové skupiny, odešle zprávu IGMP směrovačům v síti, které poté doručují odesílaná data. Pro taková zařízení je vyčleněna třída IPv4 adres D v rozsahu *224.0.0.0* až *239.255.255.255*.

V IPv6 řídí práci s multicastovými skupinami protokol MLD (Multicast Listener Discovery), který spadá pod protokol ICMPv6. MLD existuje ve dvou verzích – MLDv1 a MLDv2 přičemž MLDv1 odpovídá IGMPv2 a MLDv2 verzi IGMPv3. Jednou z hlavních výhod MLDv2 je filtrování příjmu multicastu podle zdroje. Hlavní zprávy využívané protokolem MLDv2 jsou *MLD Query*, *Report* a *Done* [13].

1.3 Vrstva transportní

Hlavním úkolem transportní vrstvy je zajištění přenosu dat mezi dvěma koncovými zařízeními též nazývanými aplikační programy [2]. Na základě požadavků komunikujících programů dokáže tato vrstva regulovat tok dat oběma směry či měnit charakter přenosu z nespojového na spojový. Transportní vrstva poskytuje logickou komunikaci, tzn. že komunikující zařízení mohou být na různě vzdálených místech, propojeny přes několik routerů, ale komunikace probíhá jako kdyby byly připojeny přímo. Na této vrstvě jsou zavedeny tzv. porty, které představují adresy v rámci konkrétního hosta, a umožňují tak odlišovat síťový provoz příslušející různým aplikacím.

Transportní spojení probíhá přímo mezi koncovými body, bez ohledu na mezilehlou síť. Na jedné straně komunikačního kanálu se nachází odesílatel. Na straně odesílatele je na transportní vrstvě zpráva z aplikační vrstvy převedena na segment tak, že je rozdělena na několik částí a ke každé je přidáno záhlaví. Data jsou poté poslána na síťovou vrstvu, kde se opět přidá záhlaví a vznikne datagram. Příjemce, který se nachází na druhé straně komunikačního kanálu provádí s daty opačný postup.

Nejpoužívanějšími protokoly na této vrstvě jsou TCP (Transmission Control Protocol) a UDP (User Datagram Protocol).

1.3.1 TCP (Transmission Control Protocol)

Protokol TCP je nejběžněji používaným protokolem pro přenos dat. Je spojově orientovaný, což znamená, že před samotným přenosem dat je vytvářeno spojení mezi odesílatelem a příjemcem. Tento proces navázání spojení se nazývá *3 way-handshake* a zahrnuje tři základní kroky: navázání spojení, přenos dat a ukončení spojení [3].

Klient pošle zprávu bez uživatelských dat s příznakem *SYN*, server odpovídá zprávou *SYN* a *ACK*, na kterou klient finálně odpovídá *ACK*.

TCP je známý pro svou spolehlivost. Zajímá se o to, zda byla data doručena a přijata ve správném pořadí. Pokud tomu tak není, pak přenos opakuje. Další z vlastností protokolu TCP je full duplex, tedy možnost posílat data v obou směrech zároveň, a také řídit tok dat a zahlcení sítě.

TCP používá koncept portů a socketů k identifikaci koncových bodů komunikace [1]. Socket je kombinací IP adresy a portu, která jednoznačně identifikuje konkrétní bod komunikace. Datové jednotce na úrovni transportní vrstvy a využívající TCP se říká segment. Přenášená data jsou ovlivněna parametrem MSS (Maximum Segment Size), který je definován délkou MTU největšího paketu [3]. Struktura záhlaví TCP datagramu je zobrazena na následujícím obrázku 1.7.

Zdrojový port			Cílový port	
Pořadové číslo odesílaného bajtu				
Pořadové číslo potvrzovaného bajtu				
Délka záhlaví	Rezerva	Příznaky	Délka okna	
Kontrolní součet			Ukazatel naléhavých dat	
Volitelné položky				
Data				

Obr. 1.7: Záhlaví TCP segmentu [14]

Záhlaví protokolu TCP obsahuje několik parametrů. Jak již bylo zmíněno, TCP se zajímá, zda byla data doručena ve správném pořadí, proto využívá **Pořadové číslo odesílaného/potvrzovaného bajtu**. Mezi nejpoužívanější **Příznaky** patří například *URG* (*urgent*), který informuje o urgentnosti dat, či *RST* (*reset the connection*) pro odmítnutí spojení v případě duplikovaných segmentů. Další parametry TCP záhlaví jsou popsány v mnoha dalších zdrojích [2], [14],... a proto nebudou v této práci podrobně rozepisovány.

Obdobným způsobem jako navázání spojení probíhá i jeho ukončení, které se nazývá *4-way handshake* a využívá příznaky *FIN* a *ACK* [3].

1.3.2 UDP (User Datagram Protocol)

Na rozdíl od TCP představuje UDP nespojový a nespolehlivý protokol transportní vrstvy [3]. To znamená, že před samotným přenosem dat nenavazuje spojení mezi ode-

sílatelem a příjemcem a datové pakety neboli datagramy jsou tudíž odesílány nezávisle a mohou putovat po síti různými cestami. Datagramy tak mohou být doručeny v nesprávném pořadí, či dokonce ztraceny. Potvrzení přijetí dat, ani jejich opětované doručení není prováděno. Oproti TCP je naopak UDP rychlejší, a jelikož nenavazuje spojení, ani neudržuje jeho aktivní stav, je jeho výhodou nižší latence. UDP je tak vhodným protokolem pro generování různých speciálních paketů, a z uvedených důvodů byl v této práci zvolen jako hlavní protokol použitý na transportní vrstvě.

Stejně jako TCP i UDP používá porty. Většina dat je přenášena systémem klient/server přes port UDP 53 [3]. Záhlaví protokolu UDP znázorněné na obrázku 1.8 je velmi jednoduché.

Zdrojový port	Cílový port
Délka	Kontrolní součet
Data	

Obr. 1.8: Záhlaví UDP datagramu [3]

1.4 Vrstva aplikační

Aplikační vrstva představuje rozhraní mezi uživateli a síťovými službami [11]. Umožňuje jednotlivým aplikačním programům přístup k médiu a zároveň podporuje jejich spolupráci. Jednotlivé koncové programy komunikují přímo s transportní vrstvou, a služby prezentační a relační vrstvy, které v modelu TCP/IP na rozdíl od modelu ISO/OSI chybí, si musí zajistit samy.

Aplikační vrstva zahrnuje širokou škálu síťových protokolů a služeb, které podporují různé aplikace, včetně webových prohlížečů, e-mailových klientů, souborových přenosových nástrojů, hlasové a videokomunikace a mnoho dalších. Tyto aplikace mohou využívat různé aplikační protokoly, které jsou dostupné v rámci aplikační vrstvy TCP/IP. Na aplikační vrstvě lze také využívat šifrování a autentizační mechanismy za účelem zajištění ochrany citlivých informací.

1.4.1 DHCP (Dynamic Host Configuration Protocol)

Protokol DHCP se používá pro dynamické přidělování IP adres a pro konfiguraci síťových parametrů zařízení, které se připojují do sítě [3]. Celý proces přidělování IP adres a konfigurace sítě zjednodušuje a automatizuje. Proces přidělení IP adresy funguje na základě žádosti *DHCP Discovery*, kterou zasílá broadcastově host

za tím účelem, aby našel server, se kterým bude interagovat. Příslušené servery pak odpovídají klientovi zprávou *DHCP Offer*, ve které nabídnou IP adresu a uvedou informaci o *IP address lease time*, tedy informaci o tom, jak dlouho bude zapůjčení této IP adresy k dispozici. Klient podle své volby poté zasílá zprávu *DHCP Request* na příslušný server, a ten odpovídá zprávou *DHCP (N)ACK* a adresu klientovi (ne)přiřazuje. Kromě IP adres může DHCP přidělovat i další síťové parametry, jako jsou výchozí brána (default gateway), DNS servery, maska podsítě a další. To umožňuje klientům v síti okamžitě začít komunikovat bez potřeby ruční konfigurace.

1.4.2 DNS (Domain Name System)

DNS je hierarchický distribuovaný systém, jenž umožňuje překlad doménových jmen na IP adresy a opačně [3]. Každé zařízení připojené k síti má svou IP adresu, která je ale těžce zapamatovatelná, a proto je pro uživatele přeložena na doménové, snadno zapamatovatelné, jméno právě pomocí DNS. DNS pracuje na UDP portu 53.

DNS zahrnuje několik úrovní domén, které jsou uspořádány od nejvyšší po nejnižší a odděleny tečkami. Mezi ty nejvyšší patří tzv. kořenové domény jako je *.com*, *.org*, či *.net*, a také národní domény *.us*, *.uk* a jiné. Do dalších úrovní pak spadají například *google.com*, *bing.com* a další.

Existuje hned několik typů DNS záznamů. *A (Address)* záznamy překládají doménové jméno na IPv4 adresy, *AAAA (IPv6 Address)* záznamy na adresy IPv6. *MX (Mail Exchanger)* záznamy slouží k identifikaci mailového serveru pro danou doménu a podobně. Pokaždé, když uživatel zadá do svého prohlížeče doménové jméno, je odeslán DNS dotaz na DNS server, který hledanou doménu obsluhuje. Tento server poté najde záznam pro tuto doménu a vrací odpověď. Tímto umožňuje prohlížeči spojení s webovým serverem.

Existují různé typy DNS serverů. Autoritativní servery uchovávají a spravují záznamy, zatímco rekurzivní servery provádějí DNS dotazy a odpovědi. V případě, že rekurzivní servery nemají k dispozici potřebný záznam, obrací se na kořenový server.

1.4.3 mDNS (Multicast Domain Name System)

Protokol multicast DNS je používán zejména pro překlad lokálních jmen a funguje na portu 5353. Je doplněním klasického DNS, avšak s jeho hierarchií a strukturou přímo nesouvisí. MDNS umožňuje koncovým zařízením vysílat dotaz multicastově a zároveň umožňuje všem stanicím na dané lokální síti na daný dotaz odpovídat [1].

S použitím mDNS přichází řada benefitů. Jelikož jsou dotaz i odpověď zasílány multicastově, vidí a mohou si je uložit všechna zařízení v síti. Pokud chce stanice odeslat dotaz na překlad lokálního jména, které již v síti dotazováno bylo, nemusí

dotaz vysílat opětovaně, ale může si informaci najít ve své cache paměti. Probíhá také pasivní kontrola chyb, při které může stanice odstranit ze své cache paměti zastaralá data v případě, že na konkrétní dotaz nevidí příslušející odpověď. Další výhody mDNS jsou popsány v dokumentu RFC 6762 [15].

1.4.4 LLMNR (Link-Local Multicast Name Resolution)

Protokol LLMNR funguje obdobně jako protokol DNS, nicméně nezávisle na něm, konkrétně na lokálních linkách, kde je přístup k DNS omezen. LLMNR využívá port 5355 a multicastové zprávy. Multicastová IPv4 adresa, na kterou jsou zasílány odpovědi na dotazované překlady lokálních doménových jmen je 224.0.0.252. Pro IPv6 rozsah je odpovídající adresa ff02::1:3 [16].

1.4.5 HTTP (Hypertext Transfer Protocol)

HTTP je hlavním protokolem pro přenos dat na internetu a základem pro fungování www (World Wide Web) [3]. Je to textový protokol založený na principu žádost – odpověď. HTTP umožňuje klientům, typicky webovým prohlížečům, komunikovat se servery a žádat o obsah ve formě hypertextu, obrázků, videí a podobně. Proto, aby klienti mohli provádět operace na serveru jsou definovány různé metody, například metoda *GET* se používá pro získání obsahu a metoda *POST* pro odeslání dat na server.

Před přenosem dat pomocí protokolu HTTP musí být navázáno spojení. To probíhá tím způsobem, že klient odešle paket s příznakem *SYN* na server. Server přidá příznak *ACK* a vrací zprávu klientovi, který poté odešle *ACK* a požadavek na konkrétní http objekt. Webový server ve svých odpovědích na žádosti klienta zahrnuje stavový kód. Pokud byl požadovaný objekt úspěšně poskytnut prohlížeči klienta, pak zasílá server zprávu *200 OK*. Pokud server nemá oprávnění k poskytnutí požadovaného objektu, odešle zprávu *403 Forbidden – přístup zakázán*. Zpráva *404 Not Found* znamená, že objekt nebyl nalezen. HTTP je stavově nezávislý protokol, což znamená, že každá zpráva je nezávislá na předchozích žádostech. To usnadňuje škálování a správu na serveru.

Pro zabezpečení komunikace mezi klientem a serverem se používá rozšíření HTTPS (HTTP Secure), které data šifruje. Důvěrné informace jsou tak chráněny před nepovoleným přístupem, a navíc HTTPS zajišťuje integritu dat. To znamená, že data nemohou být během přenosu upravena či poškozena. Použití HTTPS na webových stránkách je standardem pro poskytnutí důvěryhodného prostředí pro komunikaci.

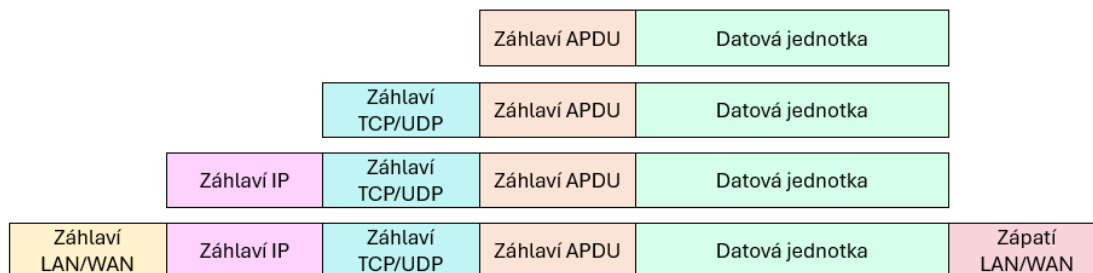
1.5 Zapouzdřování a další způsoby přenosu dat

Zapouzdřování je způsob přenosu dat, při kterém jsou data zabalena do jiných datových jednotek s hlavičkami, které obsahují informace potřebné pro jejich přenos. Mezi další techniky přenášení dat patří například přenos dat skrytým kanálem, kryptografie, či steganografie.

1.5.1 Zapouzdřování

Procesem zvaným zapouzdřování je docílen souběh protokolů [11]. Zapouzdřování zahrnuje vytvoření vrstevné struktury, ve které jsou data umístěna do balíčků zvaných pakety nebo rámce s hlavičkami obsahujícími informace potřebné pro směrování a doručení dat do cílové destinace. Zapouzdřování umožňuje rozlišit různé úrovně a protokoly v síťové komunikaci.

Proces zapouzdřování je zobrazen na obrázku 1.9. Na aplikační vrstvě vzniká aplikační PDU (Protocol Data Unit), tj. jednotka která se skládá ze záhlaví APDU a datové jednotky aplikace neboli dat. Tato jednotka je utvořena s pomocí aplikačního protokolu a data jsou poté předána na transportní vrstvu. Na transportní vrstvě je nejčastěji využíván protokol TCP. Po přidání TCP hlavičky k těmto datům vzniká segment, neboli datová jednotka transportní vrstvy. Segment je poté předán internetové vrstvě, kde je opět přidána hlavička, v tomto případě IP hlavička, a vzniká IP datagram též zvaný paket. IP paket následovně putuje přes síť k cílovému zařízení, kde je proveden opačný postup, tedy paket je postupně rozbalován.



Obr. 1.9: Proces zapouzdřování [11]

Zapouzdřování paketů umožňuje síťovým protokolům mezi sebou efektivně komunikovat, protože každý protokol rozumí své hlavičce a každá vrstva má svou roli při zapouzdřování a deencapslování paketů, čímž se vytváří hierarchie pro komunikaci v počítačových sítích. Pakety mohou mít i několik hlaviček stejného typu, např. více IP či ICMP hlaviček.

Obdobně jako je tomu na obrázku 1.9 probíhá skládání paketu i v softwarech pro generování paketů, které jsou rozebírány v kapitole 2. Tyto softwary umožňují

uživatelům skládat si jednotlivá záhlaví paketu dle preference a zápatí pak vytváří automaticky za uživatele.

1.5.2 Přenos dat skrytým kanálem

Skrytý kanál neboli Covert channel je označení pro libovolný komunikační kanál, jenž softwarové procesy, které nejsou oprávněny v rámci systému komunikovat, využívají pro přenos informací. Takový přenos nevyužívá standardní legitimní mechanismy přenosu dat a je v rozporu s bezpečnostními předpisy pro meziprocetovou komunikaci [17].

Skrytý kanál je používán především za účelem ochrany soukromí. Pro skrytí informací využívá mechanismus, který logickým kanálem vytvořeným přes existující veřejný kanál zasílá skryté informace, které nejsou autorizovány ani zaznamenávány běžnými bezpečnostními mechanismy [18]. Pro vytvoření skrytého kanálu je nezbytné programování a mnohdy i přístup k systému souborů na zdrojovém konci komunikace. S použitím skrytého kanálu ovšem přichází i jistá rizika jako je navazování zakázaných spojení, či únik důvěrných informací. Detekci skrytého kanálu je možné zajistit podrobnou analýzou, avšak je velmi obtížná [17].

Existují dva typy skrytých kanálů: *Skrytý úložný kanál (Covert storage channel)* a *Skrytý časový kanál (Covert timing channel)*. Skrytý úložný kanál zahrnuje zápis informace jedním procesem a čtení této informace procesem jiným. Ve skrytém časovém kanálu zasílá jeden proces informaci jinému procesu za přizpůsobení jeho systémových zdrojů jako je čas CPU (Central Processing Unit) [19].

Jako součást skrytého kanálu je často vnímána i steganografie blíže popsaná v následující kapitole 1.5.3.

1.5.3 Steganografie

Steganografie je technika používaná ke skrytí části informace v jiné informaci [20]. Obecně se používá pro skrytou komunikaci mezi odesílatelem a příjemcem, kdy odesílatel může vložit tajemství do krycího média tak, že odposlech nebude vnímat přítomnost přenášeného tajemství. Steganografie je často zmiňována spolu s kryptografií [21]. Obě metody se používají pro skrytí či ochranu tajných dat. Zatímco kryptografie zavádí šifrování tajných dat a činí je tak nečitelnými pro neoprávněné uživatele, steganografie existenci takových tajných zpráv skrývá. To znamená, že uživatel, který o skryté informaci neví, ji bez hledání nezaznamená. Steganografie skrývá tajná data jejich vložením například do audio, video, obrázkového nebo textového souboru.

2 Nástroje pro generování paketů

Datové pakety mohou být generovány pro různé účely, včetně testování a ladění sítí, či výzkumu a bezpečnostních analýz. Pro generování lze použít několik nástrojů. První možností je manuální generování paketů pomocí nástrojů pro programování. Pro tento účel je využíván například programovací jazyk Python či nástroj Scapy. Dále je možné generovat pakety pomocí nástrojů Nping či Hping3 v příkazovém řádku, či v programech s vlastním uživatelským rozhraním GUI (Graphical User Interface) jako je Ostinato či Packet Sender. Ať už jsou pakety generovány jakýmkoliv způsobem, je nutné proces provádět zákonně a eticky.

2.1 Nástroje

Scapy

Scapy je flexibilní interaktivní nástroj napsaný v programovacím jazyce Python, který umožňuje manipulovat s pakety [22]. Je schopen pakety generovat, odesílat a přijímat na různých síťových vrstvách. Často je používán pro vytváření vlastních paketů, skenování, tracerouting, či pro testování síťových aplikací a je určen pro pokročilé uživatele či profesionály.

Ve Scapy mohou být pakety vytvářeny téměř libovolně. Tento nástroj také umožňuje definovat parametry na každé vrstvě a postupně paket skládat. Odeslané a přijaté pakety jsou spárovány. Scapy narozdíl od jiných nástrojů poskytuje kompletní data, tzn. informace o doručených, nedoručených, či nepřirazených paketech, což uživateli napomáhá k pozitivnímu vývoji analýzy.



```
Scapy 2.5.0
Soubor Akce Úpravy Zobrazit Nápvěda

>>> IP()
<IP  >
>>> IP()/TCP()
<IP  frag=0 proto=tcp |<TCP  |>>
>>> Ether()/IP()/TCP()
<Ether type=IPv4 |<IP  frag=0 proto=tcp |<TCP  |>>>
>>>
>>> send(IP(dst="192.168.0.139",ttl=(1,4))/ICMP(), return_packets=True)
....
Sent 4 packets.
<PacketList: TCP:0 UDP:0 ICMP:4 Other:0>
>>>
>>> sr1(IP(dst="8.8.8.8")/UDP()/DNS(qd=DNSQR(qname="www.seznam.cz")))
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
<IP  version=4 ihl=5 tos=0x0 len=91 id=63674 flags= frag=0 ttl=128 proto=udp checksum=0x539d src=8.8.8.8 dst=192.168.29.139 |<UDP  sport=domain dport=domain len=71 checksum=0xd4c5 |<DNS  id=0 qr=1 opcode=QUERY aa=0 tc=0 rd=1 ra=1 z=0 ad=0 cd=0 rcode=ok qdcount=1 ancount=2 nscount=0 arcount=0 qd=<DNSQR  qname='www.seznam.cz.' qtype=A qclass=IN > an=<DNSRR  rrtype='www.seznam.cz.' type=A rclass=IN ttl=13 rdlen=4 rdata=77.75.77.222 |<DNSRR  rrtype='www.seznam.cz.' type=A rclass=IN ttl=13 rdlen=4 rdata=77.75.79.222 |>> ns=None ar=None |>>>
>>>
```

Obr. 2.1: Ukázky práce s nástrojem Scapy

Na obrázku 2.1 jsou znázorněny některé ze základních funkcí nástroje Scapy. Nejprve (1) je ukázáno, jak lze skládat protokoly jednotlivých vrstev, u nichž je poté možné specifikovat parametry. Dále (2) je zobrazeno odeslání čtyř ICMP paketů a nakonec (3) je znázorněn DNS dotaz na doménu *seznam.cz*.

Hping

Hping je dalším nástrojem pro generování a analýzu síťových paketů, který je součástí Kali Linux [23]. Umožňuje různé formy testování, včetně skenování portů, analýzy síťových protokolů a simulace síťových podmínek. Tento nástroj umožňuje odesílat vlastní ICMP, UDP, či TCP pakety a zobrazovat odpovědi. Povoluje libovolnou velikost paketů a lze jej použít i k přenosu souborů. Vychází z příkazu ping, ale také zvládá fragmentaci a režim traceroute.

Novější verzí z roku 2005 je hping3. Pomocí hping3 lze testovat pravidla brány firewall a výkon sítě, provádět skenování portů, zjišťovat cestu MTU, experimentovat s IP protokoly a podobně. Hping3 je skriptovatelný pomocí jazyka Tcl [23].

Některé z příkazů nástroje hping3: [24]

- -0 -rawip ... mód IP
- -1 -icmp ... mód ICMP
- -C -icmptype ... typ ICMP
- -2 -udp ... mód UDP
- -8 -scan ... mód skenování
- -S -syn ... nastavení příznaku SYN

Například odeslání deseti ICMP paketů je znázorněno na obrázku 2.2.

```
(root@kali)-[/home/user]
# hping3 -1 -c 10 192.168.0.139
HPING 192.168.0.139 (eth1 192.168.0.139): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.0.139 ttl=128 id=65464 icmp_seq=0 rtt=7.0 ms
len=46 ip=192.168.0.139 ttl=128 id=65465 icmp_seq=1 rtt=9.7 ms
len=46 ip=192.168.0.139 ttl=128 id=65466 icmp_seq=2 rtt=8.6 ms
len=46 ip=192.168.0.139 ttl=128 id=65467 icmp_seq=3 rtt=8.6 ms
len=46 ip=192.168.0.139 ttl=128 id=65468 icmp_seq=4 rtt=9.5 ms
len=46 ip=192.168.0.139 ttl=128 id=65469 icmp_seq=5 rtt=8.2 ms
len=46 ip=192.168.0.139 ttl=128 id=65470 icmp_seq=6 rtt=4.0 ms
len=46 ip=192.168.0.139 ttl=128 id=65471 icmp_seq=7 rtt=2.3 ms
len=46 ip=192.168.0.139 ttl=128 id=65472 icmp_seq=8 rtt=5.3 ms
len=46 ip=192.168.0.139 ttl=128 id=65473 icmp_seq=9 rtt=7.4 ms

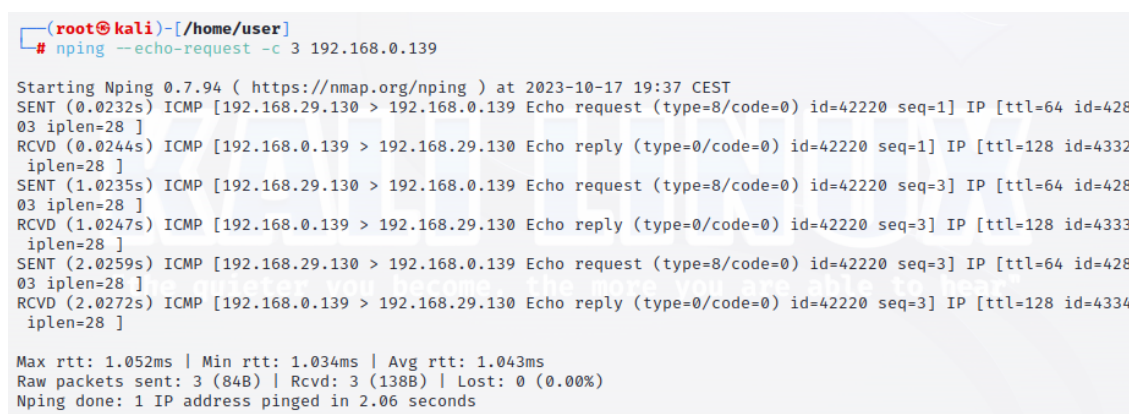
— 192.168.0.139 hping statistic —
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.3/7.0/9.7 ms
```

Obr. 2.2: Odeslání deseti ICMP paketů pomocí nástroje hping3

Nping

Obdobou nástroje Hping je poněkud novější Nping [25]. Nping je součástí Nmap, open-source nástroje pro pokročilé testování a analýzu sítí. Díky integraci Nping s funkcemi Nmap mohou uživatelé provádět pokročilé analýzy sítí, které mohou zahrnovat testování dostupnosti a odezvy, generování různých typů provozů a podobně. Nping je vhodný i pro zátěžové testování sítě a simulaci útoků jako DoS (Denial of Service), či ARP spoofing.

Kromě toho Nping nabízí speciální režim provozu nazvaný *Echo*, který slouží k simulaci ICMP Echo request a Echo reply paketů. Tím zařízení zjišťuje, zda je cílový počítač dostupný a jaký je čas jeho odezvy. Tato funkce pracuje obdobně jako nástroj ping, ale umožňuje provádět pokročilejší analýzy sítě pomocí dalších Nping funkcí. Echo mód je ukázaný na obrázku 2.3.



```
(root@kali)-[/home/user]
# nping --echo-request -c 3 192.168.0.139

Starting Nping 0.7.94 ( https://nmap.org/nping ) at 2023-10-17 19:37 CEST
SENT (0.0232s) ICMP [192.168.29.130 > 192.168.0.139 Echo request (type=8/code=0) id=42220 seq=1] IP [ttl=64 id=42803 iplen=28 ]
RCVD (0.0244s) ICMP [192.168.0.139 > 192.168.29.130 Echo reply (type=0/code=0) id=42220 seq=1] IP [ttl=128 id=4332 iplen=28 ]
SENT (1.0235s) ICMP [192.168.29.130 > 192.168.0.139 Echo request (type=8/code=0) id=42220 seq=3] IP [ttl=64 id=42803 iplen=28 ]
RCVD (1.0247s) ICMP [192.168.0.139 > 192.168.29.130 Echo reply (type=0/code=0) id=42220 seq=3] IP [ttl=128 id=4333 iplen=28 ]
SENT (2.0259s) ICMP [192.168.29.130 > 192.168.0.139 Echo request (type=8/code=0) id=42220 seq=3] IP [ttl=64 id=42803 iplen=28 ]
RCVD (2.0272s) ICMP [192.168.0.139 > 192.168.29.130 Echo reply (type=0/code=0) id=42220 seq=3] IP [ttl=128 id=4334 iplen=28 ]

Max rtt: 1.052ms | Min rtt: 1.034ms | Avg rtt: 1.043ms
Raw packets sent: 3 (84B) | Rcvd: 3 (138B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 2.06 seconds
```

Obr. 2.3: Odeslání tří ICMP paketů pomocí nástroje Nping v režimu Echo

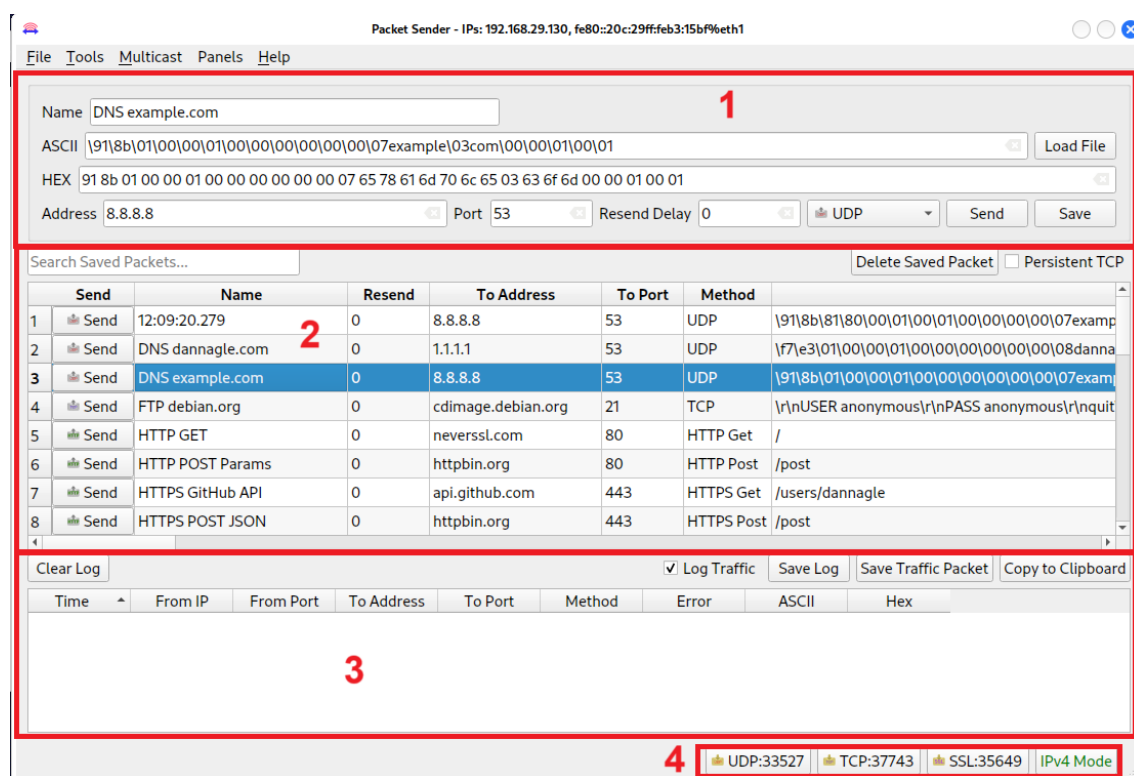
Packet Sender

Packet Sender je nástroj pro jednoduché odesílání a přijímání síťových paketů [26]. Je vhodný pro testování síťových scénářů prostřednictvím vlastních síťových paketů. Využívá příkazového řádku, ale disponuje i přehledným uživatelským grafickým rozhraním a je snadno použitelný i pro začátečníky, ale zároveň užitečný pro vývojáře, administrátory a testery sítí.

Umožňuje uživatelům vytvářet a odesílat vlastní síťové pakety s přesně definovanými parametry, a také příchozí pakety přijímat, což je užitečné pro diagnostiku komunikace mezi zařízeními. Packet Sender podporuje různé síťové protokoly, jako jsou TCP, UDP, ARP a další, a pakety tak mohou být generovány pro různé účely. Výhodou Packet Senderu je podpora vytváření šifrovaných spojení přes SSL (Secure

Sockets Layer). Podporovaný je též multicastový přenos a k dispozici je také nástroj pro výpočet IPv4 podsítí. Mezi další výhody patří možnost ukládat vytvořené pakety na Packet Sender Cloud, odkud mohou být v případě potřeby znovu získány.

Uživatelské rozhraní Packet Senderu je znázorněno na obrázku 2.4. V horní části okna (1) lze specifikovat název paketu, jeho cílovou adresu, zvolit port a další související data. Poté následuje část s uloženými pakety (2), ve které lze zvolit a znovu použít či upravit předem vytvořené pakety. Níže (3) se zobrazují všechny odeslané pakety. V dolní pravé části okna (4) je zobrazen stav a porty serveru UDP, TCP a SSL. Kliknutím na některý z protokolů je jeho stav aktivován/deaktivován. K dispozici je také tlačítko pro přepínání verze IP protokolu. Servery Packet Senderu jsou nakonfigurovány tak, že podporují buď IPv4 nebo IPv6, ale ne obojí současně, je proto nutné mezi protokoly přepínat.



Obr. 2.4: Uživatelské rozhraní nástroje Packet Sender

Mezi nástroje pro generování paketů patří i Ostinato. Tento nástroj byl využit pro tuto práci, a proto mu je věnována celá následující podkapitola 2.2.

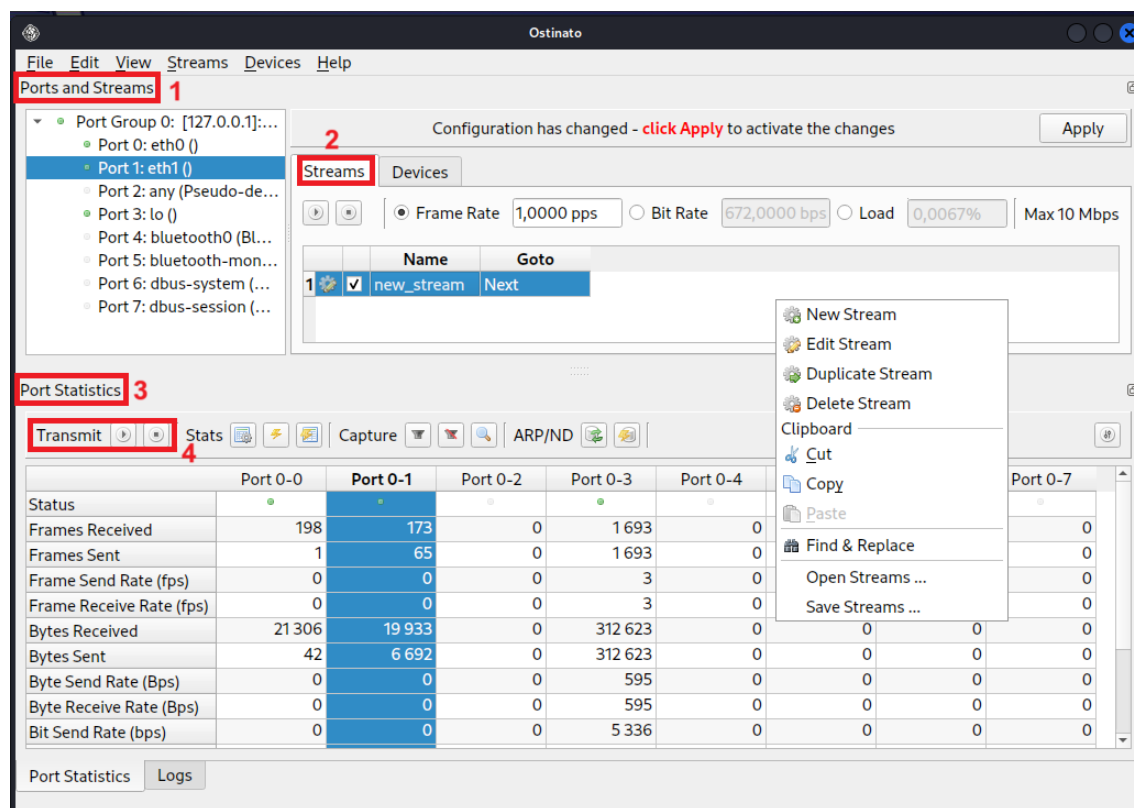
2.2 Ostinato

Ostinato je nástroj pro generování síťových paketů, který je k dispozici pro různé operační systémy [27]. Ostinato nabízí svým uživatelům velké množství funkcí pro vy-

tváření, úpravu a odesílání síťových paketů na různých vrstvách sítě. Tento nástroj je ale také využíván pro účely testování, ladění sítí, diagnostiky a výzkumu síťového provozu, a proto je vhodný taktéž pro profesionály v oblasti síťové komunikace a bezpečnosti.

2.2.1 Seznámení s uživatelským rozhraním Ostinato

Ostinato disponuje uživatelsky přívětivým grafickým rozhraním, které se skládá z jednoho hlavního okna zobrazeného na obrázku 2.5.



Obr. 2.5: Hlavní okno nástroje Ostinato

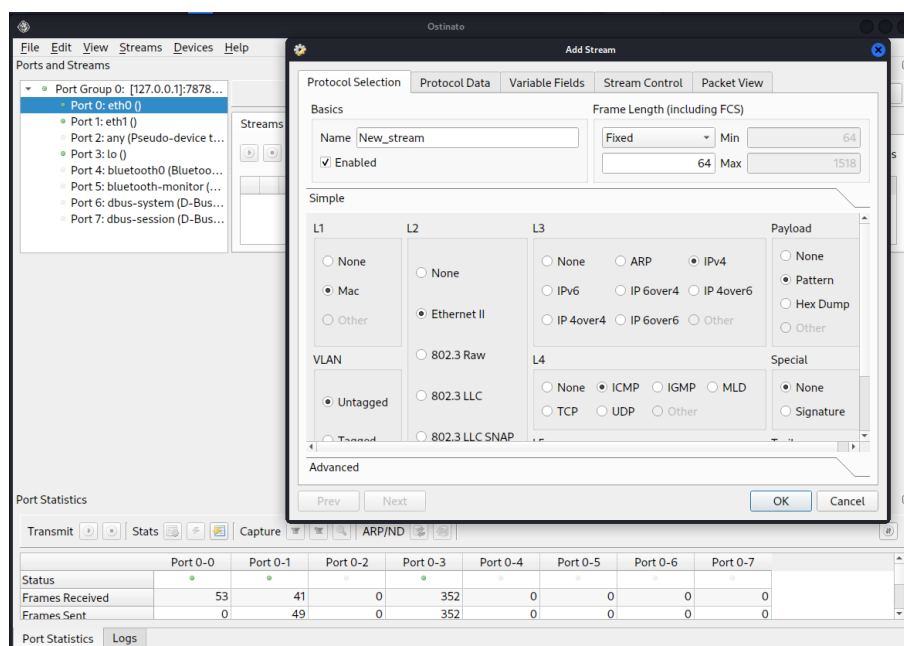
Horní menu obsahuje funkce pro provádění různých akcí jako je otevření, uložení či exportování projektu. Jak lze vidět na obrázku 2.5, v horní části hlavního okna se nachází sekce *Ports and Streams* (1), ve které uživatel zprvu volí rozhraní, na kterém bude pakety generovat a v části *Streams* (2) poté samotné pakety vytváří či importuje. V dolní polovině okna se nachází oblast *Port Statistics* (3) s několika záložkami pro různé operace s pakety. Záložka *Capture/Transmit* (4) spouští generování paketů. Záložka *Streams* umožňuje vytvářet a spravovat různé toky síťových paketů. *Statistics* poskytuje informace o statistikách spojených s generováním a odesíláním paketů, včetně odezvy a rychlosti přenosu a *Console* zobrazuje informace o chování programu, včetně chyb, upozornění a jiných událostí.

Ostinato podporuje mnoho protokolů včetně Ethernet, IP, TCP, UDP, ICMP a dalších. Jeho uživatelé mohou vytvářet vlastní síťové pakety a upravovat již existující, což umožňuje testování různých aspektů síťové komunikace. Pro vytváření a úpravu síťových paketů se využívá panel nastavení paketů, který umožňuje definovat různé atributy, jako jsou zdrojové a cílové adresy, protokoly, porty, obsah dat a podobně. Další výhodou je, že Ostinato lze integrovat s nástrojem Wireshark pro analýzu zachyceného síťového provozu, což umožňuje detailní studium chování sítě.

2.2.2 Generování paketů v programu Ostinato

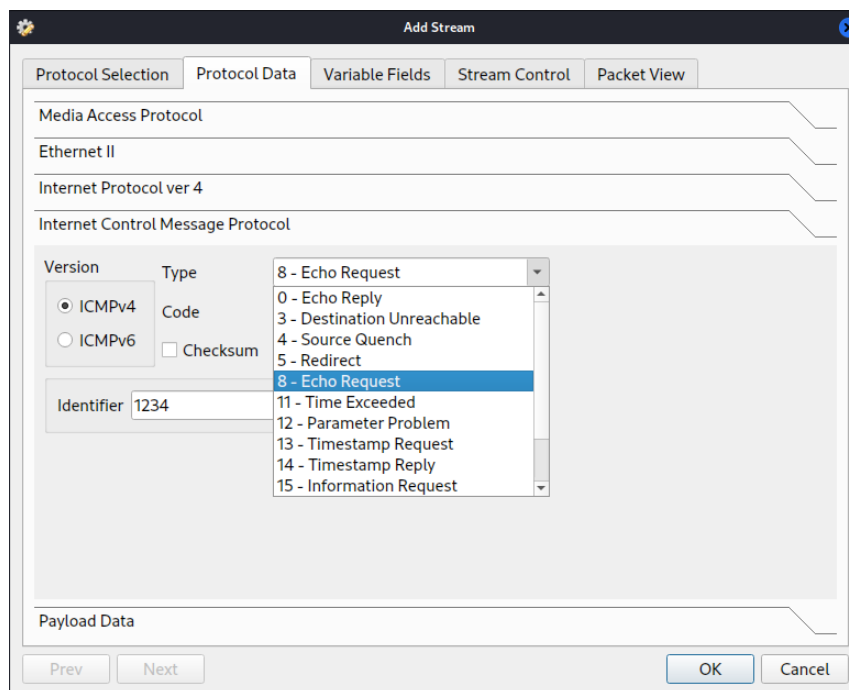
Proces generování paketů v programu Ostinato je popsán a znázorněn na obrázcích 2.6 až 2.10.

Pro vytvoření nového datového toku (streamu) je nejprve nutné zvolit jeden z nabízených portů. Po volbě portu je pak možné nový datový tok vytvořit (*New Stream*), či existující soubor nahrát a upravit (*Open Streams*). Při zvolení možnosti *New Stream* se zobrazí nové okno, ve kterém lze sestavit paket dle preferencí uživatele. V záložce *Protocol Selection* si uživatel volí protokoly na jednotlivých vrstvách *L1* až *L5* jak lze vidět na obrázku 2.6.



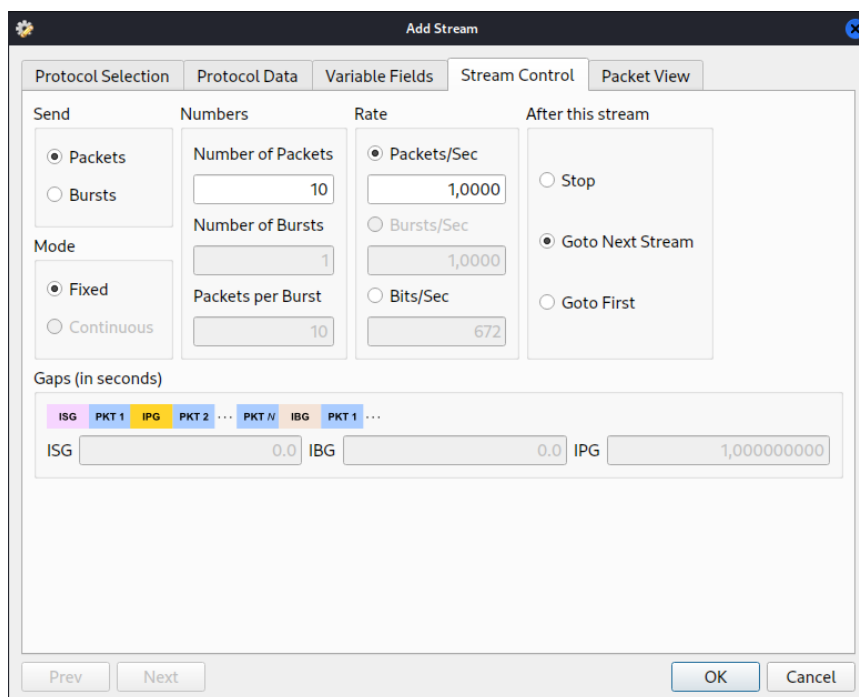
Obr. 2.6: Sestavení paketu (Ostinato)

V záložce *Protocol Data* (obrázek 2.7) lze specifikovat další parametry jednotlivých protokolů. Například se zde nastavuje zdrojová a cílová MAC a IP adresa, či typy zpráv ICMP.



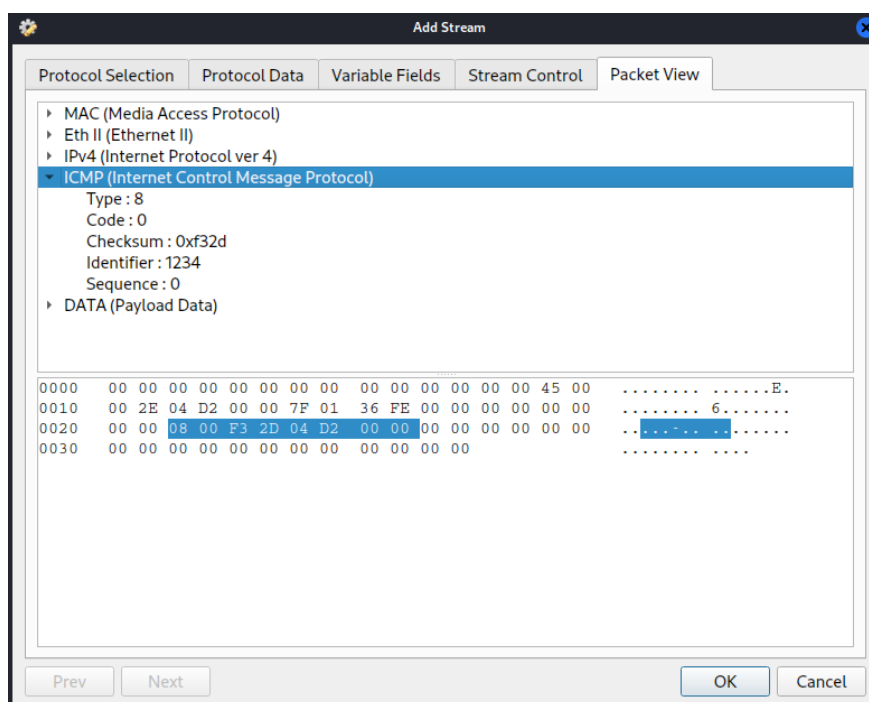
Obr. 2.7: Volba dalších parametrů (Ostinato)

V sekci *Variable Fields* je možné nastavit další proměnná políčka a v podokně *Stream Control* (obrázek 2.8) jsou specifikovány požadavky na odesílání jako je celkový počet paketů, počet paketů odeslaných za jednu sekundu, a podobně.



Obr. 2.8: Sekce Stream Control (Ostinato)

V poslední záložce *Packet View* (obrázek 2.9) je možné si sestavený paket zobrazit.



Obr. 2.9: Záložka Packet View (Ostinato)

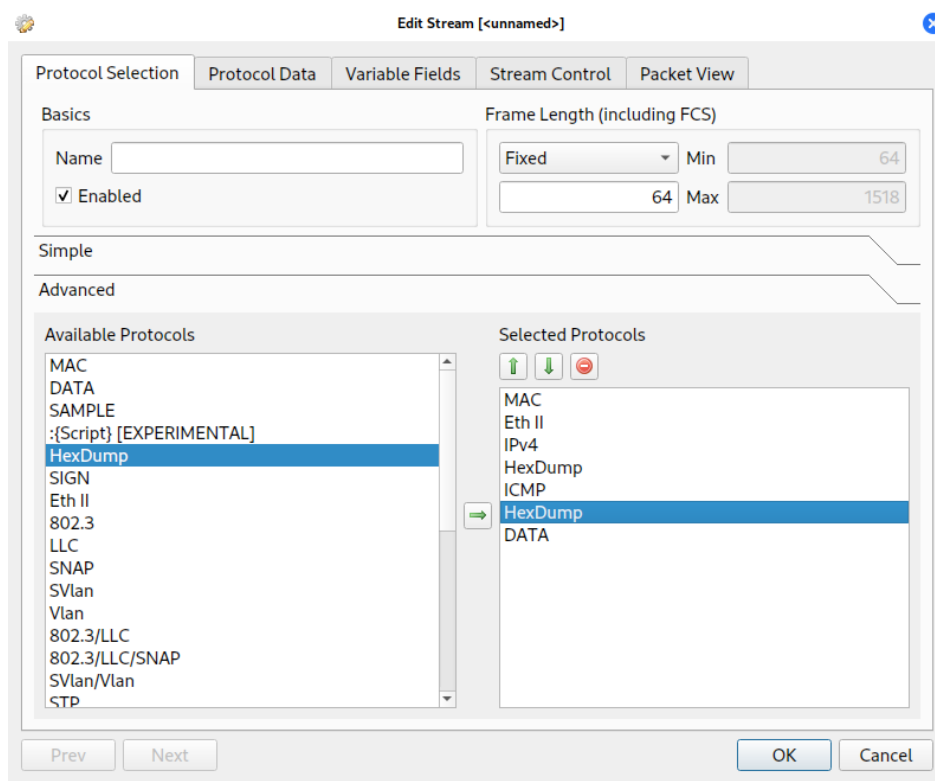
Následně je v dolní části okna *Port Statistics* (obrázek 2.10) potřeba zvolit vybraný port a spustit odesílání vytvořených paketů.

Port Statistics							
Transmit Stats ARP/ND							
	Port 0-1	Port 0-2	Port 0-3	Port 0-4	Port 0-5	Port 0-6	Port 0-7
Status	●	○	●	○	○	○	○
Frames Received	65	0	751	0	0	0	0
Frames Sent	61	0	751	0	0	0	0
Frame Send Rate (fps)	0	0	3	0	0	0	0
Frame Receive Rate (fps)	0	0	3	0	0	0	0
Bytes Received	10 162	0	132 924	0	0	0	0
Bytes Sent	8 840	0	132 924	0	0	0	0
Byte Send Rate (Bps)	0	0	591	0	0	0	0
Byte Receive Rate (Bps)	0	0	591	0	0	0	0
Bit Send Rate (bps)	0	0	5 304	0	0	0	0

Obr. 2.10: Spuštění odesílání paketů (Ostinato)

Zachycený provoz je pak možné si zobrazit a dále analyzovat v programu Wireshark.

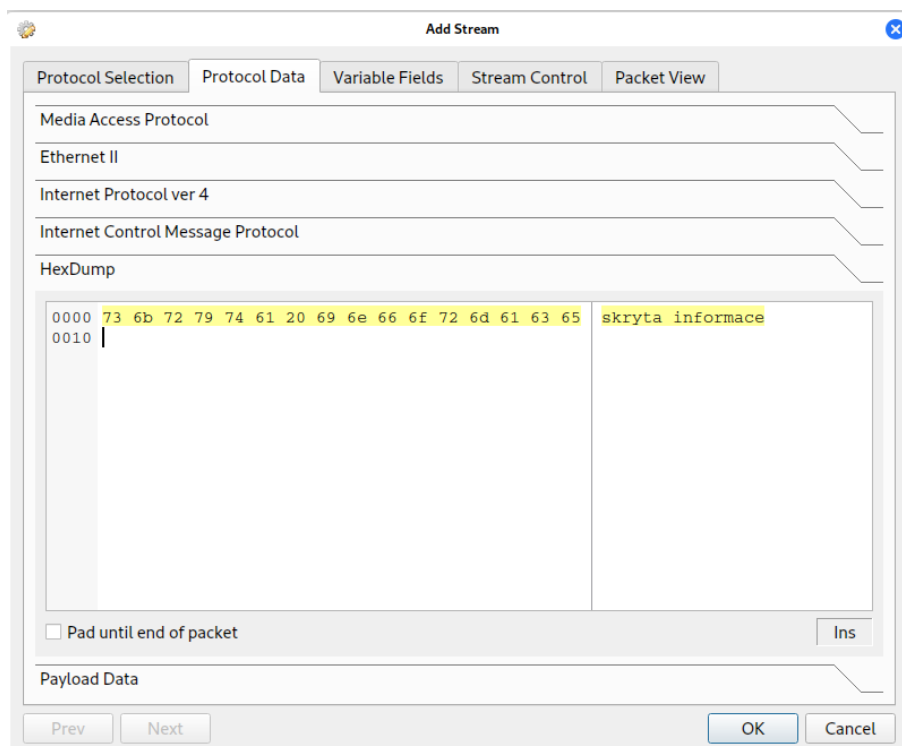
V Ostinatu je též možné při tvorbě paketů využít funkci HexDump, která umožňuje skrytí informace. Pole HexDump je možné vložit libovolně mezi jednotlivá záhlaví v sekci *Protocol Selection* viz obrázek 2.11.



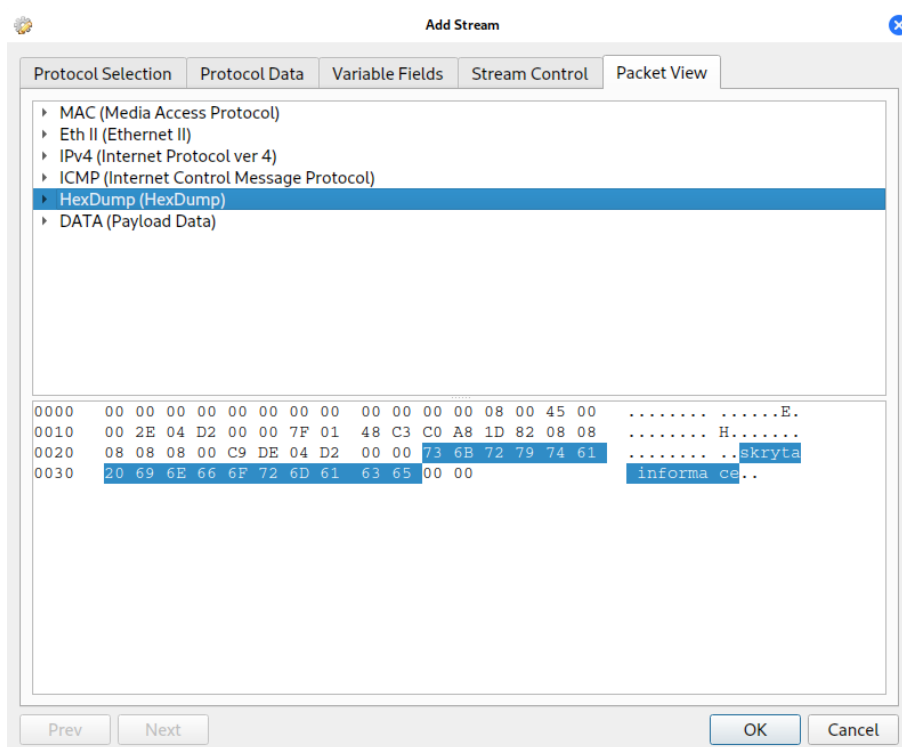
Obr. 2.11: Přidání pole HexDump (Ostinato)

Na záložce *Protocol Data* se vkládá obsah HexDumpu zapsaný hexadecimálně (obrázek 2.12).

Jakým způsobem se bude obsah HexDumpu zobrazovat je možné vidět v záložce *Packet View* (obrázek 2.13).



Obr. 2.12: Vložení obsahu do pole HexDump (Ostinato)



Obr. 2.13: Obsah pole HexDump (Ostinato)

2.3 Srovnání nástrojů

Všechny výše zmíněné nástroje jsou vhodné pro generování provozu na síti a testování sítě.

Scapy je nástroj umožňující provádění složitých úkonů s pakety a vytváření vlastních skriptů v programovacím jazyce Python, a tak je vhodný spíše pro profesionály. Naopak Packet Sender či Ostinato jsou vhodné i pro méně zkušené uživatele, protože disponují uživatelsky přívětivým a lehce ovladatelným rozhraním. Nping a Hping jsou nástroje vhodné pro skenování síťových portů.

Scapy poskytuje oproti nástrojům Hping a Nping velkou výhodu v tom, že odezva na odeslání paketu není redukována pouze na informaci *otevřeno/uzavřeno/filtrováno*, ale jsou zobrazeny veškeré informace. Velkým plusem nástrojů Scapy a Ostinato je podpora několika protokolů na různých vrstvách sítě. V nástrojích Hping a Packet Sender je manipulace s pakety omezena především na vrstvu internetovou a transportní za podpory pouze několika základních protokolů.

Ostinato nepodporuje stavová TCP připojení, nicméně jeho funkce jsou dostačující a vhodné pro generování paketů pro účely této práce. Hlavní výhodou je možnost skládání jednotlivých záhlaví paketu, což je užitečná funkce pro skrytí tajné informace, kterou budou mít studenti za úkol při vypracování samostatného projektu objevit.

Hping3 je co se týče srovnání parametrů téměř identický s Nping, a proto byl v tabulce srovnávající tyto nástroje z důvodu přehlednosti vynechán.

Tab. 2.1: Přehled nástrojů pro generování paketů

	Scapy	Nping	Packet Sender	Ostinato
Open-source	ano	ano	ano	ano†
Programovatelný	ano	ne	ne	ano
Platformy*	L, W, M	L, W, M	L, W, M	L, W, M
GUI	ne	ne	ano	ano
Skenování portů	ano	ano	ne	ano
Protokoly L3	různé	IPv4, IPv6, ICMP	IPv4, IPv6, ICMP	různé
Protokoly L4	různé	TCP, UDP	TCP, UDP, SSL	různé

* L – Linux, W – Windows, M – Mac OS

†Od dubna roku 2024 Ostinato již není open-source.[28].

2.4 Rešerše podobných prací

Jedním ze zájmů v této práci bylo zjistit, zda se obdobnému tématu, tedy tématu generování nestandardních paketů na síti Ethernet za využití nástrojů Ostinato a Wireshark, již někdo věnoval. Nejprve byl prohledán archiv závěrečných prací Vysokého Učení Technického a poté byl využit vyhledávač Google Scholar, kde bylo nalezeno několik souvisejících prací či článků.

V rámci VUT v minulých letech vzniklo několik prací zameraných na vytváření simulačních úloh v programech NS2, NS3, GNS3, Scapy, či IT Guru, které sloužily především k výuce počítačových cvičení předmětů BPC-KOM či MPC-PKT, nebo pro závěrečné ověření znalostí studentů, kteří jeden z těchto předmětů absolvovali. Tyto práce byly soustředěny na širokou škálu protokolů modelu TCP/IP. Vznikla například práce věnovaná generování IPv6 a ICMPv6 paketů [8], či práce zahrnující protokoly FTP a HTTP [29].

Jeden ze studentů se též zaměřil na práci s programem Wireshark pro analýzu síťového provozu. Výstupem jeho prací [30] a [31] jsou laboratorní scénáře, které studenty předmětu Komunikační technologie seznamují s prostředím programu Wireshark a soubory se zachycenými pakety pro analýzu protokolů ICMP, IP, DNS, TCP, DoH a mnoho dalších.

Na Google Scholar bylo nalezeno několik odborných prací, které se zaměřují jednak na problematiku generování provozu na síti, a také na vytvoření vlastního generátoru síťového provozu či na srovnání různých již existujících generátorů viz [32] a [33]. Mezi porovnávané nástroje byl mnohdy zahrnut i program Ostinato, který byl převážně hodnocen kladně, a tím bylo utvrzeno rozhodnutí, že byl vhodně zvolen pro tuto diplomovou práci.

Velmi zřídka lze najít práce věnující se steganografickým metodám využívaných pro odesílání IP paketů [34]. Častěji se vyskytují práce na téma obrázkové steganografie, ty však pro účely této práce nejsou relevantní.

3 Vytvoření testovacího prostředí

Pro účely generování nestandardních paketů je nutné vytvořit testovací prostředí. Generování nestandardních paketů je prováděno z důvodu bezpečnosti a snadné změny konfigurace ve virtualizačním prostředí VMware Workstation. Jako operační systém je využíván Kali Linux a pro práci s pakety, respektive jejich vytváření a analýzu, jsou využity programy Ostinato a Wireshark.

3.1 VMware Workstation

Virtualizační prostředí VMware představuje komplexní řešení pro virtualizace serverů či desktopů, tzn. umožňuje vytvářet virtuální instance operačních systémů a aplikací na jednom fyzickém serveru nebo desktopu [35]. Hlavními výhodami virtualizačních softwarů je efektivní využití zdrojů, kdy na jednom fyzickém serveru může běžet hned několik virtuálních serverů. Virtuální servery jsou od sebe izolovány, což zvyšuje bezpečnost. Virtualizace dále snižuje náklady na provoz datových center a umožňuje centralizovanou správu dat.

V rámci této diplomové práce bylo využito VMware Workstation 17 Player, bezplatná verze virtualizačního software společnosti VMware. Tato verze VMware Workstation je vhodná pro výuku, vývoj a testování softwaru, ať už jde o operační systémy, aplikace nebo síťová řešení. Je to užitečný nástroj pro všechny, kteří chtějí vytvářet a spravovat virtuální prostředí na svém počítači. Uživatelé mohou vytvářet virtuální stroje různých operačních systémů, včetně Windows, Linux, macOS a dalších. Virtuální stroje mohou být spouštěny přímo na hostitelském počítači, což umožňuje běh více operačních systémů současně.

3.2 Kali Linux

Kali Linux je specializovaná linuxová distribuce navržená pro potřeby penetračního testování, bezpečnostního výzkumu a digitální forenzní analýzy [36]. Nabízí širokou škálu nástrojů pro profesionály, kteří se zabývají testováním a zabezpečováním počítačových systémů a sítí.

Tento software je aktivně vyvíjen a aktualizován, a navíc si uživatelé snadno mohou Kali Linux přizpůsobit svým potřebám. Je také jednoduchý na použití. Je možné ho spustit z virtuálních strojů či USB, a také obsahuje různé penetrační a testovací scénáře, což samotné testování usnadňuje. Je určen pouze pro legální účely jako je etické hackování či bezpečnostní výzkum.

Hlavními důvody, proč byl pro tuto práci zvolen právě Kali Linux jsou ty, že tato linuxová distribuce založená na Debianu je open-source a zdarma. Kali Linux je vyvíjen v zabezpečeném prostředí a je přizpůsobitelný svým uživatelům. Podporuje několik bezdrátových zařízení a také dodržuje standard FHS (Filesystem Hierarchy Standard), který umožňuje uživatelům Linuxu snadno najít binární soubory, podpůrné soubory, knihovny atd.

3.3 Wireshark

Wireshark je populární nástroj pro analýzu síťového provozu [37]. Jedná se o open-source software, který slouží k zachycení, rozboru a vizualizaci provozu na různých úrovních sítě. Umožňuje zachytit pakety, které zahrnují různé protokoly a komunikační data. U zachycených dat je možné zobrazit podrobné informace o všech použitých protokolech, či vyfiltrovat pouze tu část obsahu, která má být podrobena analýze. Navíc lze zachycená data zobrazit i v podobě grafů, stromových diagramů, či tabulek, což analýzu usnadňuje. Kromě zachytávání dat také umožňuje načtení dat ze souboru.

Wireshark najde využití pro monitoring sítě a diagnostiku její bezpečnosti, pod kterou spadá identifikace problémů jako jsou ztráty paketů či chyby v komunikaci, anebo také síťové útoky.

3.3.1 Seznámení s uživatelským rozhraním Wireshark

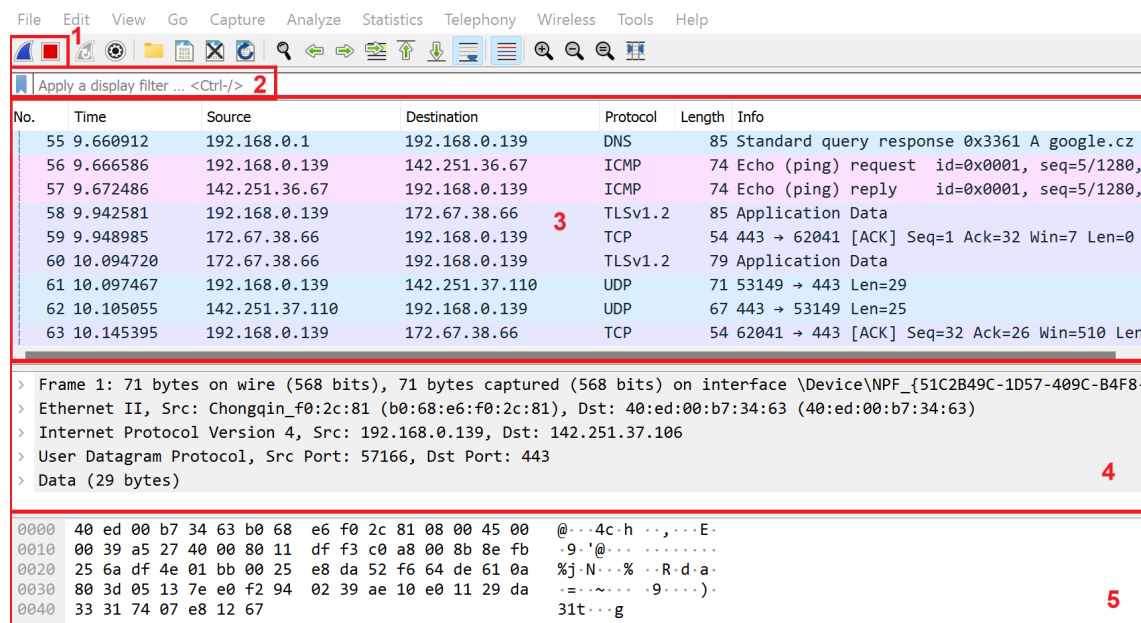
Po spuštění Wiresharku se zobrazí výpis dostupných síťových rozhraní. Je třeba zvolit to, na kterém má být provoz zachytáván či analyzován. Po volbě rozhraní se zobrazí hlavní okno Wiresharku, které je znázorněno na obrázku 3.1. Pro zahájení zachytávání provozu na zvoleném rozhraní slouží tlačítko *Start (capture)*, pro ukončení tlačítko *Stop (capture)* (1). Zachycená data se zobrazí v hlavním okně Wiresharku ve formě seznamu paketů, kdy každý řádek v seznamu představuje jeden síťový paket a obsahuje informace o číslu paketu, času jeho zachycení, jeho zdrojové a cílové IP adrese, použitém protokolu a dalších detailech (3). Pomocí dialogu *Preferences* lze nastavit, které sloupce z tohoto jednořádkového výpisu mají být zobrazovány.

Pro zjednodušení je možné zachycené pakety filtrovat pomocí různých filtrů. Například lze filtrovat data na základě zdrojové IP adresy, cílového portu nebo použitého protokolu. Filtr může vypadat například takto: *ip.addr == 192.168.1.1* (zobrazení komunikace s určitou IP adresou) a vkládá se do příslušného pole nad seznamem paketů (2) [38].

Po kliknutí na konkrétní paket se v dolní části okna zobrazí okno stromu protokolů, které byly během přenosu použity (4). Dále je pro každý z těchto protokolů

možné zobrazit pole s podrobnými informacemi o hlavičkách paketu, jeho obsahu, délce apod.

Poslední částí grafického rozhraní je okno pro zobrazení dat, které poskytuje pohled na surová data paketu označeného v souhrnném okně (5).



Obr. 3.1: Hlavní okno programu Wireshark

Kromě filtrování dat je také možné data graficky zobrazit, uložit pro budoucí analýzu, či exportovat do různých formátů. Wireshark také nabízí různé statistiky síťového provozu. Statistiku lze vytvořit přes tlačítko *Statistics* v hlavním menu. Lze zobrazit statistiky o použití protokolů, zdrojových a cílových IP adresách, portech a dalších aspektech síťového provozu.

Analýza síťového provozu v programu Wireshark pomáhá identifikovat bezpečnostní rizika, problémy se zahlcením sítě, či jiné anomálie v komunikaci.

4 Návrhy nestandardních datových toků

Praktická část diplomové práce představuje a podrobně popisuje některá z navržených řešení sestav datových toků se skrytými daty. Skrytá informace, která může být vložena do různých částí paketu, nemá ovlivňovat běžnou komunikaci zvoleným protokolem.

Pro vytvoření paketů byl využit nástroj Ostinato a pro jejich monitorování a kontrolu funkčnosti nástroj Wireshark. Celkem v rámci této práce bylo navrženo a sestrojeno deset variant datových toků:

- BGP: Optional Parameters: Authentication
- DHCP: Options: Unassigned
- TCP: Options: Experimental
- DNS: TXT, RRSIG
- FTP: IPv6 Authentication Header
- ICMPv6: IPv6 Routing Header
- mDNS: AAAA
- MLD: Multicast Address Exclude
- QUIC: Connection IDs
- UDP: IPv6 Hop-by-Hop Option

V této kapitole je podrobně popsáno pouze pět z výše vyjmenovaných variant. V některých případech jsou prezentované varianty rozpracovány do více možností, které připadají v úvahu. Všechny varianty, respektive soubory *.pcapng* a *.ostm* reprezentující vytvořené pakety, jsou poskytnuty jako příloha A ve výpisu příloh 5.8 kde jsou také popsány.

Základními požadavky pro tvorbu nestandardních paketů byly různorodost použitých protokolů, skrytí informace v takových částech paketů, kde se standardně nevyskytuje a především skládání jednotlivých záhlaví paketů tak, aby je bylo možné dle potřeby snadno modifikovat a využít například pro výuku předmětů BPC-KOM a MPC-PKT jako zadání semestrálního projektu.

V této kapitole jsou postupně představeny jednotlivé varianty simulovaných paketů – DHCP (kapitola 4.1), DNS (kapitola 4.2), MLD (kapitola 4.3), QUIC (kapitola 4.4) a UDP (kapitola 4.5). Proces generování a zachycení datových toků je ilustrován několika obrázky.

4.1 DHCP – Unassigned option

Jak lze vidět na obrázku 4.1, kterým je snímek obrazovky programu Wireshark, datový tok znázorňuje komunikaci, ve které zařízení žádají DHCP server o přidělení IPv4 adresy. Komunikace se skládá ze zpráv *DHCP Discover*, *Offer*, *Request* a *ACK*

a tyto zprávy se v ní opakují celkem čtyřikrát, tzn. čtyři zařízení žádají o přidělení IP adresy. Souběžně s klasickou DHCP komunikací probíhá také skrytá komunikace zahrnující výpočet rovnice. Tato skrytá komunikace žádným způsobem nemá ovlivňovat komunikaci základní.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0.0.0.0	255.255.255.255	DHCP	330	DHCP Discover - Transaction ID 0x7d159732
2	0.010009189	0.0.0.0	255.255.255.255	DHCP	330	DHCP Discover - Transaction ID 0x7d159732
3	0.020008902	0.0.0.0	255.255.255.255	DHCP	330	DHCP Discover - Transaction ID 0x7d159732
4	0.030015516	0.0.0.0	255.255.255.255	DHCP	330	DHCP Discover - Transaction ID 0x7d159732
5	0.040010574	192.168.29.254	192.168.29.129	DHCP	359	DHCP Offer - Transaction ID 0x7d159732
6	0.050061111	192.168.29.254	192.168.29.130	DHCP	359	DHCP Offer - Transaction ID 0x7d159732
7	0.060110007	192.168.29.254	192.168.29.131	DHCP	359	DHCP Offer - Transaction ID 0x7d159732
8	0.070113067	192.168.29.254	192.168.29.132	DHCP	359	DHCP Offer - Transaction ID 0x7d159732
9	0.080227277	0.0.0.0	255.255.255.255	DHCP	335	DHCP Request - Transaction ID 0x7d159732
10	0.090568028	0.0.0.0	255.255.255.255	DHCP	335	DHCP Request - Transaction ID 0x7d159732
11	0.100598954	0.0.0.0	255.255.255.255	DHCP	335	DHCP Request - Transaction ID 0x7d159732
12	0.110527312	0.0.0.0	255.255.255.255	DHCP	335	DHCP Request - Transaction ID 0x7d159732
13	0.120527440	192.168.29.254	192.168.29.129	DHCP	359	DHCP ACK - Transaction ID 0x7d159732
14	0.130523517	192.168.29.254	192.168.29.130	DHCP	359	DHCP ACK - Transaction ID 0x7d159732
15	0.140525711	192.168.29.254	192.168.29.131	DHCP	359	DHCP ACK - Transaction ID 0x7d159732
16	0.150526887	192.168.29.254	192.168.29.132	DHCP	359	DHCP ACK - Transaction ID 0x7d159732

Obr. 4.1: Simulovaná DHCP komunikace

Pro skrytí informace je využito DHCP pole *Option*, do něhož lze vložit až 255 bajtů dat. DHCP Options zprávy *Discover* jsou znázorněny na obrázku 4.2.

```

Dynamic Host Configuration Protocol (Discover)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x7d159732
  Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: VMware_b3:15:bf (00:0c:29:b3:15:bf)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Discover)
  > Option: (50) Requested IP Address (0.0.0.0)
  > Option: (12) Host Name
  > Option: (55) Parameter Request List
  > Option: (215) Unassigned
  > Option: (255) End

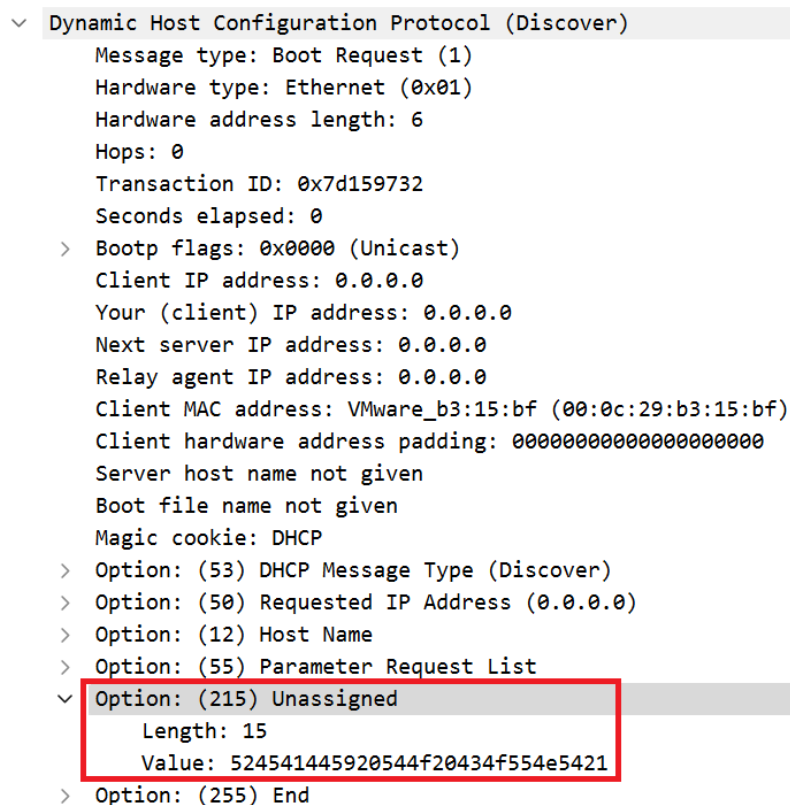
```

Obr. 4.2: DHCP Options zprávy DHCP Discover

Existuje několik známých a běžně používaných hodnot DHCP options, které definuje organizace IANA (Internet Assigned Numbers Authority) a mezi které patří

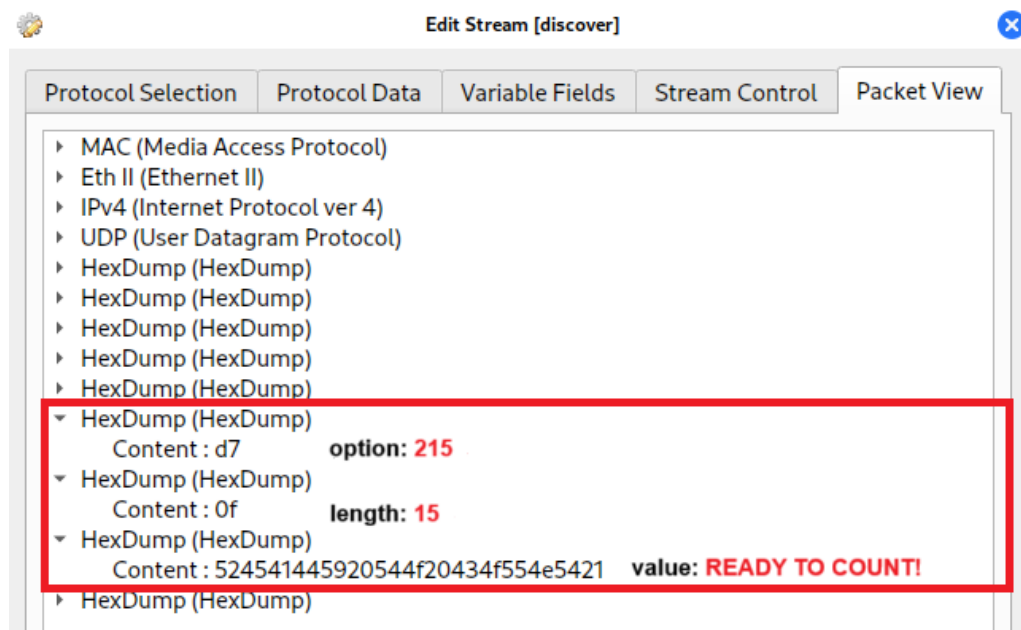
například: 3 – *Gateway*, 12 – *Hostname*, či 51 – *Lease Time*. K dispozici je naopak i několik málo tzv. *unassigned* hodnot, ke kterým žádná DHCP option přiřazená není, například: 111, 149, 178-207, 222, apod. Právě tyto hodnoty lze využít například pro účely skrytí dat [39].

Pro tvorbu vlastní DHCP (unassigned) option je třeba analyzovat hexadecimální reprezentaci jedné z již existujících DHCP options a dle jejího vzoru sestavit v programu Ostinato DHCP option další. Řetězec hexadecimálních znaků představující pole DHCP Option se skládá ze tří částí, kterými jsou *hodnota (option)*, *délka (length)* a *obsah (value)* viz obrázek 4.3.

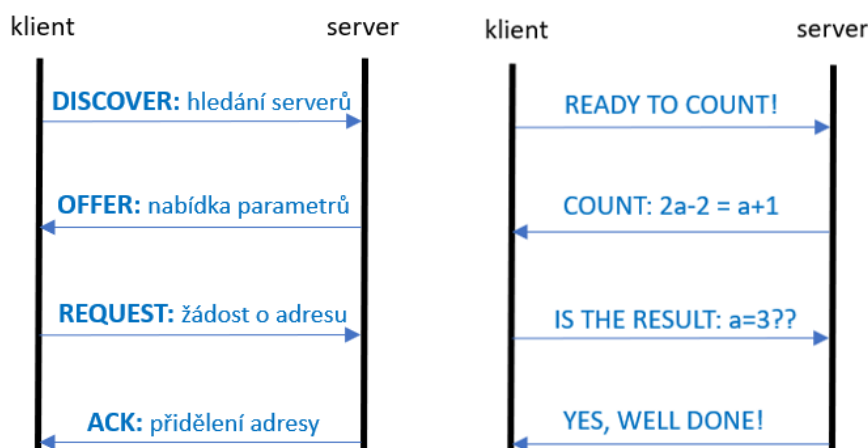


Obr. 4.3: DHCP Unassigned option (Wireshark)

Na obrázku 4.4 je zobrazeno rozdělení DHCP Option do tří polí HexDump v programu Ostinato. Jednotlivá pole jsou rozdělena proto, aby mohla být snadno měněna. Při tomto uspořádání lze jednoduše přepsat hodnotu pole, délku, i samotná data. V této komunikaci je pro DHCP Option zvolena hodnota **215**. Pole skrývající informaci je vloženo do každé ze zpráv DHCP komunikace, která simuluje výměnu zpráv klienta s DHCP serverem zahrnující výpočet rovnice. Komunikaci znázorňují ve zjednodušené formě diagramy na obrázku 4.5 a následně screenshoty z nástroje Wireshark.



Obr. 4.4: DHCP Unassigned option (Ostinato)



Obr. 4.5: Diagram DHCP komunikace (běžná vs. skrytá komunikace)

Ve zprávě DHCP Discover klient indikuje zprávou **READY TO COUNT**, že je připraven od DHCP serveru přijmout příklad k vypočítání (obrázek 4.6).

Server odpovídá zprávou s rovnicí **COUNT: $2a-2=a+1$** (obrázek 4.7).

Na rovnici poté zařízení odpovídá ve zprávě Request tak, že se dotazuje, zda příklad spočítal správně **IS THE RESULT $a=3??$** (obrázek 4.8).

Na závěr DHCP server potvrzuje správnost výpočtu skrytou zprávou **YES, WELL DONE!** (obrázek 4.9) a přiřazuje klientovi požadovanou IPv4 adresu.

Time	Source	Destination	Protocol	Length	Info
1 0.00000000	0.0.0.0	255.255.255.255	DHCP	330	DHCP Discover - Transaction ID 0x7d159732
2 0.010009189	0.0.0.0	255.255.255.255	DHCP	330	DHCP Discover - Transaction ID 0x7d159732
> Bootp flags: 0x0000 (Unicast) Client IP address: 0.0.0.0 Your (client) IP address: 0.0.0.0 Next server IP address: 0.0.0.0 Relay agent IP address: 0.0.0.0 Client MAC address: VMware_b3:15:bf (00:0c:29:b3:15:bf) Client hardware address padding: 000000000000000000 Server host name not given Boot file name not given Magic cookie: DHCP > Option: (53) DHCP Message Type (Discover) > Option: (50) Requested IP Address (0.0.0.0) > Option: (12) Host Name > Option: (55) Parameter Request List > Option: (215) Unassigned Length: 15 Value: 524541445920544f20434f554e5421 > Option: (255) End					
0000	ff ff ff ff ff ff 00 0c	29 b3 15 bf 08 00 45 00			).....E
0010	01 3c 04 d2 00 00 7f 11	35 e0 00 00 00 00 ff ff			<.....5.....}
0020	ff ff 00 44 00 43 01 28	03 2e 01 01 06 00 7d 15			...D C (.....}
0030	97 32 00 00 00 00 00 00	00 00 00 00 00 00 00 00			2.....}
0040	00 00 00 00 00 00 0c 29	b3 15 bf 00 00 00 00 00			)
0050	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00c0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0100	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0110	00 00 00 00 00 63 82	53 63 35 01 31 32 04 00			...c Sc5...2..
0120	00 00 00 0c 04 6b 61 6c	69 37 0d 01 1c 02 03 0f			...kal i7...
0130	06 77 0c 2c 2f 1a 79 2a	d7 0f 52 45 41 44 59 20			...w.../y*...READY
0140	54 4f 20 43 4f 55 4e 54	21 ff			TO COUNT I

Obr. 4.6: Informace skrytá ve zprávě DHCP Discover

Time	Source	Destination	Protocol	Length	Info
5 0.040010574	192.168.29.254	192.168.29.129	DHCP	359	DHCP Offer - Transaction ID 0x7d159732
6 0.050061111	192.168.29.254	192.168.29.130	DHCP	359	DHCP Offer - Transaction ID 0x7d159732
Client IP address: 0.0.0.0 Your (client) IP address: 192.168.29.129 Next server IP address: 192.168.29.254 Relay agent IP address: 0.0.0.0 Client MAC address: VMware_b3:15:bf (00:0c:29:b3:15:bf) Client hardware address padding: 000000000000000000 Server host name not given Boot file name not given Magic cookie: DHCP > Option: (53) DHCP Message Type (Offer) > Option: (54) DHCP Server Identifier (192.168.29.254) > Option: (51) IP Address Lease Time > Option: (1) Subnet Mask (255.255.255.0) > Option: (28) Broadcast Address (192.168.29.255) > Option: (3) Router > Option: (15) Domain Name > Option: (6) Domain Name Server > Option: (215) Unassigned Length: 15 Value: 434f554e543a2e32612d323d612b31 > Option: (255) End					
0000	00 0c 29 b3 15 bf 00 50	56 e9 f1 8e 08 00 45 10			...P V.....E
0010	01 59 04 d2 00 00 10 11	ec c5 c0 a8 1d fe c0 a8			Y.....}
0020	1d 81 00 43 00 44 01 45	1e a7 02 01 06 00 7d 15			...C D E.....}
0030	97 32 00 00 00 00 00 00	00 00 c0 a8 1d 81 c0 a8			2.....}
0040	1d fe 00 00 00 00 0c 29	b3 15 bf 00 00 00 00 00			)
0050	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00c0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0100	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0110	00 00 00 00 00 63 82	53 63 35 01 02 36 04 c0			...c Sc5...6..
0120	a8 1d fe 33 04 00 00 07	08 01 04 ff ff ff 00 1c			...3.....}
0130	04 c0 a8 1d ff 03 04 c0	a8 1d 02 0f 0b 6c 6f 63			...loc
0140	61 6c 64 6f 6d 61 69 6e	06 04 c0 a8 1d 02 d7 0f			aldomain
0150	43 4f 55 4e 54 3a 2e 32	61 2d 32 3d 61 2b 31 ff			COUNT: 2 a-2=a+1
0160	00 00 00 00 00 00 00 00				}

Obr. 4.7: Informace skrytá ve zprávě DHCP Offer

Time	Source	Destination	Protocol	Length	Info
9 0.080227277	0.0.0.0	255.255.255.255	DHCP	335	DHCP Request - Transaction ID 0x7d159732
10 0.090568028	0.0.0.0	255.255.255.255	DHCP	335	DHCP Request - Transaction ID 0x7d159732
> Bootp flags: 0x0000 (Unicast) Client IP address: 0.0.0.0 Your (client) IP address: 0.0.0.0 Next server IP address: 0.0.0.0 Relay agent IP address: 0.0.0.0 Client MAC address: VMware_b3:15:bf (00:0c:29:b3:15:bf) Client hardware address padding: 000000000000000000 Server host name not given Boot file name not given Magic cookie: DHCP > Option: (53) DHCP Message Type (Request) > Option: (50) Requested IP Address (192.168.29.1) > Option: (12) Host Name > Option: (55) Parameter Request List > Option: (215) Unassigned Length: 19 Value: 49532054484520524553554c543a613d3333f > Option: (255) End					
0000	ff ff ff ff ff ff 00 0c	29 b3 15 bf 08 00 45 00			).....E
0010	01 41 04 d2 00 00 7f 11	35 db 00 00 00 00 ff ff			A.....5.....}
0020	ff ff 00 44 00 43 01 2d	62 92 01 01 06 00 7d 15			...D C (.....}
0030	97 32 00 00 00 00 00 00	00 00 00 00 00 00 00 00			2.....}
0040	00 00 00 00 00 00 0c 29	b3 15 bf 00 00 00 00 00			)
0050	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			aldomain
00c0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
00f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0100	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			}
0110	00 00 00 00 00 63 82	53 63 35 01 03 32 04 c0			...c Sc5...2..
0120	a8 1d 81 0c 04 6b 61 6c	69 37 0d 01 1c 02 03 0f			...kal i7...
0130	06 77 0c 2c 2f 1a 79 2a	d7 13 49 53 20 54 48 45			...w.../y*...IS THE
0140	20 52 45 53 55 4c 54 3a	61 3d 33 3f 3f ff 00			RESULT: a=3??

Obr. 4.8: Informace skrytá ve zprávě DHCP Request

Time	Source	Destination	Protocol	Length	Info
13 0.120527440	192.168.29.254	192.168.29.129	DHCP	359	DHCP ACK - Transaction ID 0x7d159732
14 0.130523517	192.168.29.254	192.168.29.130	DHCP	359	DHCP ACK - Transaction ID 0x7d159732

Client IP address: 0.0.0.0	0000	00 0c 29 b3 15 bf 00 50	56 e9 f1 8e 08 00 45 10	..)....P V....E
Your (client) IP address: 192.168.29.129	0010	01 59 04 d2 00 00 10 11	ec c5 c0 a8 1d fe c0 a8	-Y.....
Next server IP address: 192.168.29.254	0020	1d 81 00 43 00 44 01 3f	1e a7 02 01 06 00 7d 15	...C-D-?}
Relay agent IP address: 0.0.0.0	0030	97 32 00 00 00 00 00 00	00 00 c0 a8 1d 81 c0 a8	-2.....
Client MAC address: VMware_b3:15:bf (00:0c:29:b	0040	1d fe 00 00 00 00 00 0c	29 b3 15 bf 00 00 00 00	)
Client hardware address padding: 00000000000000	0050	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
Server host name not given	0060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
Boot file name not given	0070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
Magic cookie: DHCP	0080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
	0090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (53) DHCP Message Type (ACK)	00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (54) DHCP Server Identifier (192.168.29	00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (51) IP Address Lease Time	00c0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (1) Subnet Mask (255.255.255.0)	00d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (28) Broadcast Address (192.168.29.255)	00e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (3) Router	00f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (15) Domain Name	0100	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
> Option: (6) Domain Name Server	0110	00 00 00 00 00 00 63 82	53 63 35 01 05 36 04 c0	c- Sc5- 6-
> Option: (215) Unassigned	0120	a8 1d fe 33 04 00 00 07	08 01 04 ff ff ff 00 1c	...3.....
> Option: (15) Domain Name	0130	04 c0 a8 1d ff 03 04 c0	a8 1d 02 0f 0b 6c 6f 63	loc
> Option: (215) Unassigned	0140	61 6c 64 6f 6d 61 69 6e	06 04 c0 a8 1d 02 d7 0f	aldomain
> Option: (215) Unassigned	0150	59 45 53 2c 20 57 45 4c	4c 20 44 4f 4e 45 21 ff	YES, WEL L DONE!
> Option: (255) End	0160	00 00 00 00 00 00 00 00		

Obr. 4.9: Informace skrytá ve zprávě DHCP ACK

Rovnice, jež je součástí DHCP komunikace, je pomocí programu Ostinato generována takovým způsobem, že se s každou další DHCP Offer zprávou mění. Konkrétně se v každé Offer zprávě proměnná **a** mění na následující písmenko abecedy a hodnota čísel vzrůstá o **1**. Rovnice schované ve čtyřech po sobě jdoucích DHCP Offer zprávách znázorňuje obrázek 4.10.

COUNT: .2 a-2=a+1

COUNT: .2 b-3=b+2

COUNT: .2 c-4=c+3

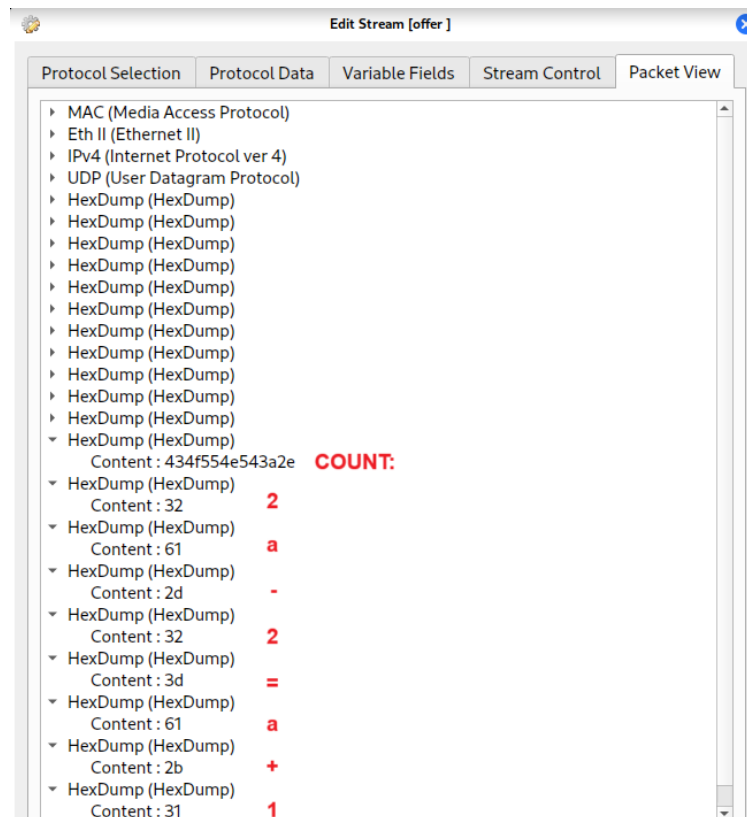
COUNT: .2 d-5=d+4

Obr. 4.10: Rovnice vložené do DHCP Offer zpráv

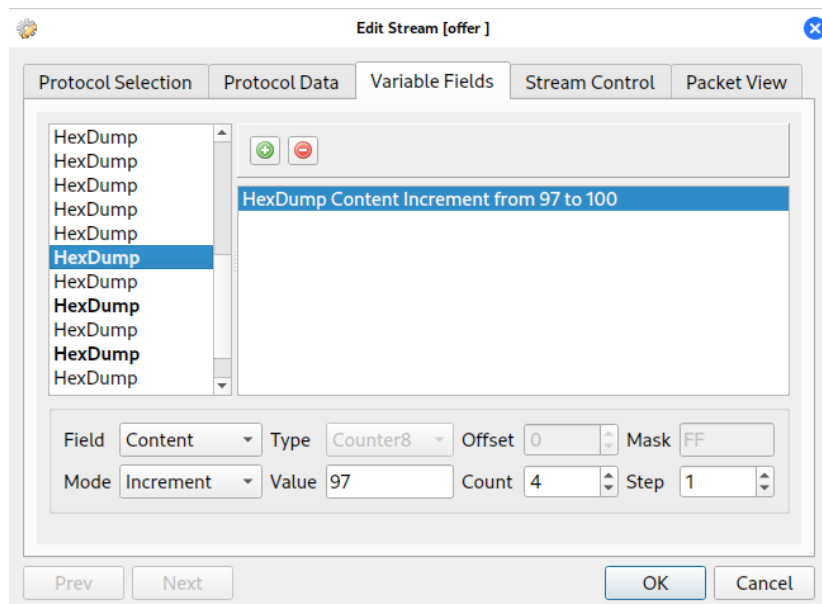
Změny znaků rovnice je dosaženo pomocí polí HexDump a funkce *Variable fields*. Jednotlivé znaky rovnice jsou rozděleny do samostatných polí HexDump (viz obrázek 4.11), což umožňuje použití funkce *Variable fields*, která mění hodnotu proměnné dle nastavených parametrů. Obrázek 4.12 ilustruje nastavení proměnnosti hodnoty **a** na hodnoty **b, c, d,...** Pole **Mode** je nastaveno na hodnotu *Increment* a pole **Step** na hodnotu *1*. Změna začíná na hodnotě **Value 97**, což je ASCII hodnota pro písmeno *a*. Pole **Count**, které je nastavené na hodnotu *4* znamená, že k nárůstu hodnoty o +1 bude docházet celkem čtyřikrát.

Obdobný princip návrhu paketu lze použít i jiným způsobem a u jiných protokolů. Výše uvedené rovnice a jejich uspořádání do jednotlivých paketů DHCP komunikace jsou pouze jednou možnou ukázkou. Pro návrh je důležité právě rozložení

dat do editovatelných bajtů či polí, která lze v programu Ostinato díky grafickému rozhraní snadno vizualizovat.



Obr. 4.11: Jednotlivá pole HexDump v Ostinato



Obr. 4.12: Tvorba Variable Fields v Ostinato

4.2 DNS Response - TXT, RRSIG

Dalším protokolem použitým pro přenos skrytých dat je protokol DNS. Simulovaná komunikace je jednoduchá. Zařízení zasílají na DNS server žádosti o poskytnutí různých typů záznamů jako je *A/AAAA*, *ANY* či *TXT*. Na dotazy DNS server odpovídá zprávami, v nichž požadované záznamy poskytuje. V rámci odpovědi je možné nalézt hned několik variant pro skrytí dat. V této komunikaci jsou takové možnosti celkem čtyři, a to odpověď typu *TXT*, nevyžádaný *TXT* záznam jako součást *Additional records*, pole *Signature* podpisu *RRSIG*, či více polí podpisu *RRSIG* (*Original TTL*, *Signature Expiration*, *Signature Inception*, *Key Tag*). Komunikaci obsahující tyto varianty ilustruje obrázek 4.13.

No.	Time	Source	Destination	Length	Protocol	Info
1 0...		192.168.1.102	192.168.1.1	69	DNS	Standard query 0x0003 AAAA seznam.cz
2 0...		192.168.1.1	192.168.1.102	125	DNS	Standard query response 0x0003 AAAA seznam.cz AAAA 2a02:598:2::1222 AAAA 2a02:598:2::1222
3 0...		192.168.1.102	192.168.1.1	94	DNS	Standard query 0x62fe TXT fekt.vut.cz OPT
4 0...		192.168.1.1	192.168.1.102	207	DNS	Standard query response 0x62fe TXT fekt.vut.cz TXT OPT
5 0...		192.168.1.102	192.168.1.1	68	DNS	Standard query 0x0002 A vutbr.cz
6 0...		192.168.1.1	192.168.1.102	113	DNS	Standard query response 0x0002 A vutbr.cz A 147.229.2.90 TXT
7 0...		192.168.1.102	192.168.1.1	94	DNS	Standard query 0xcc3e A eu-teams.events.data.microsoft.com
8 0...		192.168.1.1	192.168.1.102	220	DNS	Standard query response 0xcc3e A eu-teams.events.data.microsoft.com CNAME eu-teams.events.data.microsoft.com
9 0...		192.168.1.104	192.168.1.1	70	DNS	Standard query 0x0002 ANY novinky.cz
0...		192.168.1.1	192.168.1.104	234	DNS	Standard query response 0x0002 ANY novinky.cz SOA ans.seznam.cz RRSIG
0...		192.168.1.104	192.168.1.1	70	DNS	Standard query 0x0002 ANY novinky.cz
0...		192.168.1.1	192.168.1.104	234	DNS	Standard query response 0x0002 ANY novinky.cz AAAA 2a02:598:a::78:151 RRSIG

Obr. 4.13: Pakety ilustrující DNS komunikaci

První možností, která je v rámci DNS využita pro skrytí informace, je záznam *TXT*, jenž může být jednak součástí odpovědi či jako nevyžádaný záznam s doplňujícími informacemi v sekci *Additional records*. Tyto dvě varianty jsou téměř totožné, a proto obrázek 4.14 znázorňuje pouze jednu z nich, konkrétně textové pole jako součást *Additional records*. Pro účel skrytí informace v *TXT* poli je potřeba vytvořit DNS pole typu **16** a jeho délku přizpůsobit délce skrývaného textu. Celková délka paketu se při tomto způsobu skrytí informace mění. U změny obsahu pole *RRSIG* je délka paketu naopak neměnná. Délka *RRSIG* je totiž definovaná použitým algoritmem.

Hexadecimální reprezentaci pole *Additional record* typu *TXT* a bajty představující jednotlivé části tohoto pole znázorňuje obrázek 4.15.

Obrázek 4.16 ukazuje, že pro skrytí tajné informace lze v případě protokolu DNS využít také obsáhlý *RRSIG Signature*, který slouží jako digitální podpis DNS záznamu a pomáhá ověřit jeho autenticitu. Jednou z variant v rámci *RRSIG* je vložení informace přímo do pole *Signature*. Podpis je dlouhý několik desítek až stovek bajtů, a tak do něho lze skrýt informaci poměrně obsáhlou. Vložení dat do podpisu je v nástroji Ostinato dosaženo upravením hexadecimální reprezentace *RRSIG*. Narozdíl od ostatních variant, změna *RRSIG* základní komunikaci ovlivní. Zásah

No.	Time	Source	Destination	Length	Protocol	Info
1	0...	192.168.29.254	1.1.1.1	87	DNS	Standard query 0x1198 A utko.fekt.vut.cz OPT
2	0...	1.1.1.1	192.168.29.254	121	DNS	Standard query response 0x1198 A utko.fekt.vut.cz A 147.229.71.224 TXT

> Frame 2: 121 bytes on wire (968 bits),	0000	00 50 56 e9 f1 8e 00 50	56 e2 b8 a2 08 00 45 00	·PV···P V····E·
> Ethernet II, Src: VMware_e2:b8:a2 (00:	0010	00 6b 57 c7 00 00 80 11	02 13 01 01 01 01 c0 a8	·kW·····
> Internet Protocol Version 4, Src: 1.1.	0020	1d fe 00 35 eb a5 00 57	92 14 11 98 81 a0 00 01	···5···W ······
> User Datagram Protocol, Src Port: 53,	0030	00 01 00 00 00 01 04 75	74 6b 6f 04 66 65 6b 74	·····u tko·fekt
> Domain Name System (response)	0040	03 76 75 74 02 63 7a 00	00 01 00 01 c0 0c 00 01	·vut·cz· ······
Transaction ID: 0x1198	0050	00 01 00 00 37 25 00 04	93 e5 47 e0 00 00 10 00	·····7%·· ··G·····
> Flags: 0x81a0 Standard query response	0060	01 00 00 01 00 00 12 11	53 4b 52 59 54 41 20 49	······· SKRYTA I
Questions: 1	0070	4e 46 4f 52 4d 41 43 45	20	NFORMACE
Answer RRs: 1				
Authority RRs: 0				
Additional RRs: 1				
> Queries				
> Answers				
> Additional records				
> <Root>: type TXT class IN				
Name: <Root>				
Type: TXT (16) (Text strings)				
Class: IN (0x0001)				
Time to live: 256 (4 minutes, 16				
Data length: 18				
TXT Length: 17				
TXT: SKRYTA INFORMACE				

[Request In: 1]

Obr. 4.14: Informace skrytá v TXT poli

> Answers	0000	00 50 56 e9 f1 8e 00 50	56 e2 b8 a2 08 00 45 00	·PV···P V····E·
> Additional records	0010	00 6b 57 c7 00 00 80 11	02 13 01 01 01 01 c0 a8	·kW·····
> <Root>: type TXT, class IN	0020	1d fe 00 35 eb a5 00 57	92 14 11 98 81 a0 00 01	···5···W ······
Name: <Root>	0030	00 01 00 00 00 01 04 75	74 6b 6f 04 66 65 6b 74	·····u tko·fekt
Type: TXT (16) (Text strings)	0040	03 76 75 74 02 63 7a 00	00 01 00 01 c0 0c 00 01	·vut·cz· ······
Class: IN (0x0001)	0050	00 01 00 00 37 25 00 04	93 e5 47 e0 00 00 10 00	·····7%·· ··G·····
Time to live: 256 (4 minutes, 16 seconds)	0060	01 00 00 01 00 00 12 11	53 4b 52 59 54 41 20 49	······· SKRYTA I
Data length: 18	0070	4e 46 4f 52 4d 41 43 45	20	NFORMACE
TXT Length: 17				
TXT: SKRYTA INFORMACE				

Obr. 4.15: DNS Additional records – části pole TXT

do obsahu *RRSIG* totiž znamená, že podpis nebude dále validní, což však v případě kdy klient neověřuje autentičnost záznamů nevadí.

Druhou možností pro skrytí informace v rámci *RRSIG*, jež zachycuje obrázek 4.17, je její skrytí přes více polí, konkrétně přes pole *Original TTL*, *Signature Expiration*, *Signature Inception* a *Key Tag*. Tato pole mají pevně danou a omezenou délku a celkem je do nich možné vložit pouze 14 B. I tímto způsobem se v závislosti na vkládané informaci podpis zpravidla opět stane nevalidním.

No.	Time	Source	Destination	Length	Protocol	Info
9	0.088...	192.168.1.104	192.168.1.1	70	DNS	Standard query 0x0002 ANY novinky.cz
10	0.089...	192.168.1.1	192.168.1.104	234	DNS	Standard query response 0x0002 ANY novinky.cz SOA ans.seznam.cz RRSIG
Frame 10: 234 bytes on wire (1872 bits), 234 bytes captured on interface 0						
Ethernet II, Src: ASUSTekCOMPU_1f:bb:a4 (3c:7c:3f:1f:bb:a4), Dst: 192.168.1.1						
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.104						
User Datagram Protocol, Src Port: 53, Dst Port: 63199						
Domain Name System (response)						
Transaction ID: 0x0002						
Flags: 0x8180 Standard query response, No error						
Questions: 1						
Answer RRs: 2						
Authority RRs: 0						
Additional RRs: 0						
Queries						
Answers						
novinky.cz: type SOA, class IN, mname ans.seznam.cz						
novinky.cz: type RRSIG, class IN						
Name: novinky.cz						
Type: RRSIG (46) (Resource Record Signature)						
Class: IN (0x0001)						
Time to live: 112 (1 minute, 52 seconds)						
Data length: 82						
Type Covered: SOA (6) (Start Of a zone of Authority)						
Algorithm: ECDSA Curve P-256 with SHA-256 (13)						
Labels: 2						
Original TTL: 300 (5 minutes)						
Signature Expiration: Mar 12, 2024 12:12:00.000000						
Signature Inception: Feb 27, 2024 10:42:00.000000						
Key Tag: 23961						
Signer's name: novinky.cz						
Signature: d21cdf4c2ba88d97fbc3047c7c202020534b525e						

Obr. 4.16: Informace skrytá v RRSIG záznamu, v položce Signature

No.	Time	Source	Destination	Length	Protocol	Info
9	0.082...	192.168.1.104	192.168.1.1	70	DNS	Standard query 0x0002 ANY novinky.cz
10	0.083...	192.168.1.1	192.168.1.104	234	DNS	Standard query response 0x0002 ANY novinky.cz SOA ans.seznam.cz RRSIG
Frame 10: 234 bytes on wire (1872 bits), 234 bytes captured on interface 0						
Ethernet II, Src: ASUSTekCOMPU_1f:bb:a4 (3c:7c:3f:1f:bb:a4), Dst: 192.168.1.1						
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.104						
User Datagram Protocol, Src Port: 53, Dst Port: 63199						
Domain Name System (response)						
Transaction ID: 0x0002						
Flags: 0x8180 Standard query response, No error						
Questions: 1						
Answer RRs: 2						
Authority RRs: 0						
Additional RRs: 0						
Queries						
Answers						
novinky.cz: type SOA, class IN, mname ans.seznam.cz						
novinky.cz: type RRSIG, class IN						
Name: novinky.cz						
Type: RRSIG (46) (Resource Record Signature)						
Class: IN (0x0001)						
Time to live: 112 (1 minute, 52 seconds)						
Data length: 82						
Type Covered: SOA (6) (Start Of a zone of Authority)						
Algorithm: ECDSA Curve P-256 with SHA-256 (13)						
Labels: 2						
Original TTL: 1413565006 (16360 days, 16 hours, 56 minutes, 46 seconds)						
Signature Expiration: Sep 16, 2004 10:26:46.000000						
Signature Inception: Mar 3, 2012 17:56:33.000000						
Key Tag: 17221						
Signer's name: novinky.cz						
Signature: d21cdf4c2ba88d97fbc3047c7c202020e109d9a						

Obr. 4.17: Informace skrytá v DNS RRSIG polích

4.3 MLD – Address Exclude

Tajná informace může být též skrytá v rámci zprávy *MLDv2 Report*, která stanici umožňuje oznámit zájem o připojení do multicastové skupiny. Tento typ zprávy přenáší protokol ICMPv6. Jedním z polí této zprávy je pole *Multicast Address Exclude*, které slouží pro specifikaci multicastových adres, jež mají být vyloučeny z multicastových skupin, kterým se host chystá naslouchat. Záznamů *Multicast Address Exclude* může být v paketu uvedeno hned několik a v každém z nich lze skrývat 16 B dat.

Obrázek 4.18 ilustruje opět ve Wiresharku zachycenou komunikaci, která znázorňuje žádost několika zařízení o přístup k multicastové skupině ff02::16. Právě v jedné z těchto zpráv je záznam *Multicast Address Exclude* upraven tak, že místo adresy je vložen text **SKRYTA INFORMACE**.

No.	Time	Source	Destination	Length	Protocol	Info
1 0...		fe80::230:48ff:fe2b:d298	ff02::1	90	ICMPv6	Multicast Listener Query
2 0...		fe80::943e:325c:ecdd:7ae9	ff02::16	130	ICMPv6	Multicast Listener Report Message v2
3 0...		fe80::a6fe:115c:eaed:2a3a	ff02::16	130	ICMPv6	Multicast Listener Report Message v2
4 0...		fe80::943e:315c:acda:e932	ff02::16	110	ICMPv6	Multicast Listener Report Message v2
5 0...		fe80::913e:5e5c:edad:2e7a	ff02::16	90	ICMPv6	Multicast Listener Report Message v2
6 0...		fe80::943e:125c:aabc:7aea	ff02::16	110	ICMPv6	Multicast Listener Report Message v2
7 0...		fe80::943b:51ec:e09d:6efd	ff02::16	130	ICMPv6	Multicast Listener Report Message v2

> Frame 2: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0	0000	33 33 00 00 00 16 00 50 56 a2 77 7b 86 dd 60 00	33 33 00 00 00 16 00 50 56 a2 77 7b 86 dd 60 00
> Ethernet II, Src: VMware_a2:77:7b (00:50:56:a2:77:7b), Dst: ff02::1 (01:00:5e:00:00:02)	0010	00 00 00 4c 00 01 fe 80 00 00 00 00 00 94 3e	00 00 00 4c 00 01 fe 80 00 00 00 00 00 94 3e
> Internet Protocol Version 6, Src: fe80::943e:325c:ecdd:7ae9, Dst: ff02::16	0020	32 5c ec dd 7a e9 ff 02 00 00 00 00 00 00 00	32 5c ec dd 7a e9 ff 02 00 00 00 00 00 00 00
> Internet Control Message Protocol v6	0030	00 00 00 00 00 16 3a 00 05 02 00 00 01 00 8f 54	00 00 00 00 00 16 3a 00 05 02 00 00 01 00 8f 54
> Type: Multicast Listener Report Message v2 (Code: 84)	0040	f7 d2 00 00 00 03 02 00 00 00 ff 02 00 00 00 00	f7 d2 00 00 00 03 02 00 00 00 ff 02 00 00 00 00
> Checksum: 0xf7d2 [correct]	0050	00 00 00 00 00 00 00 01 00 02 02 00 00 00 ff 02	00 00 00 00 00 00 00 01 00 02 02 00 00 00 ff 02
> [Checksum Status: Good]	0060	00 00 00 00 00 00 00 00 00 00 01 00 03 02 00	00 00 00 00 00 00 00 00 00 00 01 00 03 02 00
> Reserved: 0000	0070	00 00 53 4b 52 59 54 41 20 49 4e 46 4f 52 4d 41	00 00 53 4b 52 59 54 41 20 49 4e 46 4f 52 4d 41
> Number of Multicast Address Records: 3	0080	43 45	43 45
> Multicast Address Record Exclude: ff02::1:2			
> Multicast Address Record Exclude: ff02::1:3			
> Multicast Address Record Exclude: 534b:5259:5441:2049:4e4f:524d:4120			
> Record Type: Exclude (2)			
> Aux Data Len: 0			
> Number of Sources: 0			
> Multicast Address: 534b:5259:5441:2049:4e4f:524d:4120			

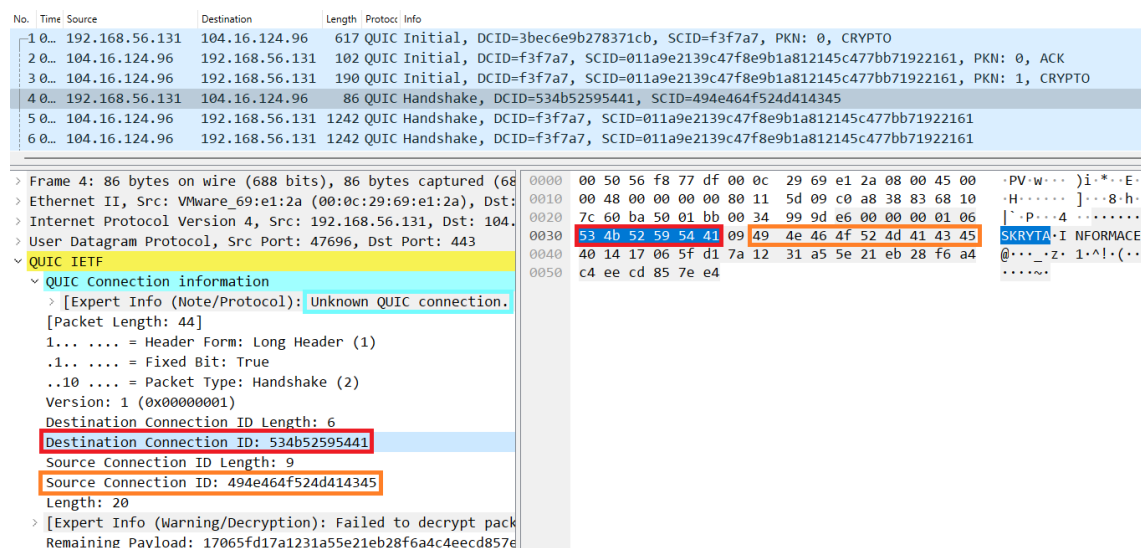
Obr. 4.18: Informace skrytá ve zprávě MLDv2 report

4.4 QUIC – Connection IDs

Další z možných variant, kam v rámci paketů skrývat data jsou dvě po sobě následující pole *Destination Connection ID* a *Source Connection ID* zprávy *Handshake* protokolu QUIC. Tato IDs slouží k identifikaci jednotlivých QUIC spojení a v každém z nich může být skryto až 255 B dat. Celkem je tedy pro vložení dat k dispozici 510 B, a proto je tento typ paketu vhodný, potřebujeme-li přenést větší množství informací. Pro vložení informace je třeba specifikovat délku jednotlivých polí a následně libovolně upravit jejich hexadecimální reprezentaci tak, aby skrytou informaci

obsahovaly. Je však nutné si uvědomit, že jakmile dojde ke změně jednoho z Connection IDs, bude Wireshark upozorňovat hláškou "Unknown QUIC Connection", že dané spojení nezná, a to z toho důvodu, že upravené Connection ID nebude odpovídat ID předchozímu, ani následujícímu (pokud tedy nebudou stejným způsobem upravena všechna IDs dané komunikace).

Obrázek 4.19 znázorňuje využití těchto dvou polí pro skrytí textu **SKRYTA INFORMACE**.



Obr. 4.19: Informace skrytá v QUIC Connection a Source ID

4.5 UDP – IPv6 Hop-by-Hop Option

Tato varianta simuluje komunikaci dvou zařízení, která si vyměňují data přes transportní protokol UDP. Pro skrytí informace je zde využito jednak rozšiřující IPv6 záhlaví typu *Hop-by-Hop Option* a také datová část protokolu UDP. Konkrétně v této komunikaci zasílá zařízení s adresou 2001:db8:85a3::8a2e:370:7334 druhému zařízení v každé zprávě jednu pětinu skryté informace.

Rozšiřující IPv6 záhlaví je poměrně vhodnou variantou pro skrytí informace. Existuje několik typů IPv6 extension headers a ve většině případů umožňují skrytí až 2048 B dat. Jednotlivé druhy rozšířených IPv6 hlaviček jsou popsány v dokumentu RFC 8200 [40].

Pro skrytí informace v rámci *IPv6 Hop-by-Hop Option* je příhodné použít pole *PadN (Padding)*, které se skládá z částí *typ*, *délka* a *data*. Obrázek 4.20 ukazuje rozšiřující záhlaví a pole *PadN* se skrytým textem **1/5: INFORMACE (look down)**. Tento text je pro všech pět paketů stejný a spojení "look down" upozorňuje

studenta, aby další informaci hledal níže, tedy v datové části. Informace v datové části UDP je proměnná, respektive je v ní skrytá vždy 1/5 věty **TVUJ TAJNY KOD JE: AE35M**. Poslední část věty je kód, který je variabilní, tzn. každé studijní skupině může být vygenerovaný jiný kód, čehož je dosaženo v nástroji Ostinato pomocí funkce *Variable fields*, která již byla představena dříve. Celou komunikaci ilustruje obrázek 4.21.

No.	Time	Source	Destination	Length	Protocol	Info
1	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	478	UDP	61243 → 63947 Len=376
2	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220

> Frame 1: 478 bytes on wire (3824 bits)	0030	ae 2e 00 00 43 28 11 04	05 02 00 00 01 20 20 20	C(.. ..)
> Ethernet II, Src: a4:20:c8:5b:c2:41 (a	0040	20 31 2f 35 20 3a 49 4e	46 4f 52 4d 41 43 45 20	1/5 :IN FORMACE
> Internet Protocol Version 6, Src: 2001	0050	28 6c 6f 6f 6b 20 64 6f	77 6e 29 20 20 20 ef 3b	(look do wn)
0110 = Version: 6	0060	f9 cb 01 80 a3 59 72 64	65 66 67 68 69 6a 6b 6c	Yrd efghijkl
> 0000 0000 = Flow	0070	6d 6e 6f 70 71 72 73 74	75 76 77 61 62 63 64 65	mnpqrst uvwabcde
.... 0000 0000 0000 0000 0000 = Flow	0080	66 67 68 69 6a 6b 6c 6d	6e 6f 70 71 72 73 74 75	fghijklm nopqrstu
Payload Length: 424	0090	76 77 61 62 63 64 65 66	67 68 69 6a 6b 6c 6d 6e	vwabcedf ghijklmn
Next Header: IPv6 Hop-by-Hop Option	00a0	6f 70 71 72 73 74 75 76	77 61 62 63 64 65 66 67	opqrstuv wabcedfg
Hop Limit: 100	00b0	68 69 6a 6b 6c 6d 6e 6f	70 71 72 73 74 75 76 77	hijklmno pqrstuvw
Source Address: 2001:db8:85a3::8a2e:	00c0	61 62 63 64 65 66 67 68	69 6a 6b 6c 6d 6e 6f 70	abcd efgh ijklmnop
Destination Address: 2001:db8:65a3:e	00d0	71 72 73 74 75 76 77 61	62 63 64 65 66 67 68 69	qrstuvw a bcedefghi
> IPv6 Hop-by-Hop Option	00e0	6a 6b 6c 6d 6e 6f 70 71	72 73 74 75 76 77 61 62	jklmnopq rstuvwab
Next Header: UDP (17)	00f0	63 64 65 66 67 68 69 6a	6b 6c 6d 6e 6f 70 71 72	cdefghij klmnopqr
Length: 4	0100	73 74 75 76 77 61 62 63	64 65 66 67 68 69 6a 6b	stuvwabc defghijk
[Length: 40 bytes]	0110	6c 6d 6e 6f 70 71 72 73	74 75 76 77 61 62 63 64	lmnopqrs tuvabcd
> Router Alert	0120	65 66 67 68 69 6a 6b 6c	6d 6e 6f 70 71 72 73 74	efghijkl mnpqrstu
> PadN	0130	75 76 77 61 62 63 64 65	66 67 68 69 6a 6b 6c 6d	vwabcedf ghijklmn
> Type: PadN (0x01)	0140	6e 6f 70 71 72 73 74 75	76 77 61 62 63 64 65 66	nopqrstu vwabcedf
Length: 32	0150	67 68 69 6a 6b 6c 6d 6e	6f 70 71 72 73 74 75 76	ghijklmn opqrstuv
PadN: 202020312f35203a494e464f52	0160	77 61 62 63 64 65 66 67	68 69 6a 6b 6c 6d 6e 6f	wabcedf ghijklmn
> User Datagram Protocol, Src Port: 6124	0170	70 71 72 73 74 75 76 77	61 62 63 64 65 66 67 68	pqrstuvw abcdefgh
> Data (376 bytes)	0180	69 6a 6b 6c 6d 6e 6f 70	71 72 73 74 75 76 77 61	ijklmnop qrstuvw
Data [truncated]: 726465666768696a6b	0190	62 63 64 65 66 67 68 69	6a 6b 6c 6d 6e 6f 70 71	bcdefghi jklmnopq
[Length: 376]	01a0	72 73 74 75 76 77 61 62	63 64 65 66 67 68 69 6a	rstuvwab cdefghij
	01b0	6b 6c 6d 6e 6f 70 71 72	73 74 75 76 77 61 62 63	klmnopqr stuvwabc
	01c0	64 65 66 67 68 69 6a 6b	6c 6d 6e 6f 70 71 72 73	defghijk lmnopqrs
	01d0	74 20 20 31 2f 35 20 3a	20 54 56 55 4a 00	t 1/5 : TVUJ.

Obr. 4.20: Informace skrytá v záhlaví Hop-by-Hop Option v poli PadN

No.	Time	Source	Destination	Length	Protocol	Info
1	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	478	UDP	61243 → 63947 Len=376 TVUJ
2	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
3	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
4	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	479	UDP	61243 → 63947 Len=377 TAJNY
5	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
6	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	477	UDP	61243 → 63947 Len=375 KOD
7	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
8	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
9	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
0...	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	476	UDP	61243 → 63947 Len=374 JE
0...	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
0...	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220
0...	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	500	UDP	61243 → 63947 Len=398 AE35M
0...	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	500	UDP	61243 → 63947 Len=398 BF46N
0...	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	500	UDP	61243 → 63947 Len=398 CG57O
0...	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	500	UDP	61243 → 63947 Len=398 DH68P
0...	0...	2001:db8:85a3::8a2e:370:7334	2001:db8:65a3:e2a0:0:ae2e:0:4328	500	UDP	61243 → 63947 Len=398 EI79Q
0...	0...	2001:db8:65a3:e2a0:0:ae2e:0:4328	2001:db8:85a3::8a2e:370:7334	282	UDP	63947 → 61243 Len=220

Obr. 4.21: Informace skrytá v poli PadN a datové části protokolu UDP

4.6 Srovnání vytvořených nestandardních paketů

V předchozí části této kapitoly bylo představeno několik variant vytvořených nestandardních paketů. Tvorba takových paketů byla ve všech případech podobná, avšak každá z variant používá pro vložení skrytého textu odlišná pole a odlišný počet polí, a tudíž má odlišné limitace pro délky vkládaných dat. Pro každou z těchto variant byly limitace testovány a ověřovány a následující dvě tabulky 4.1 a 4.2 tyto zjištěné limitace prezentují a srovnávají. Tabulka 4.1 zobrazuje přehled konkrétních polí použitých pro skrytí textu (v tabulce jako *Typ pole*) a pro každý návrh je též specifikováno přes kolik polí lze informaci vkládat (*Počet polí*). Dále tabulka zobrazuje informaci o tom, zda je daný způsob skrytí dat podporován pouze jedním z protokolů IPv4 a IPv6, či oběma verzemi (*Verze IP protokolu*). V tabulce 4.2 jsou srovnány minimální a maximální možné velikosti vkládaných dat (*Min./ Max. délka pole*) a též počty násobků, po nichž je nutno data do polí vkládat (*Vkládání dat po **).

Tab. 4.1: Výsledky ověřování rozsahu polí a podporovaná verze IP protokolu

Protokol a pole	Typ pole	Počet polí	Verze IP protokolu
IPv6 Fragment Header	více polí *	4	IPv6
IPv6 Authentication Header	více polí †	4	IPv6
IPv6 Routing Header	Address	1	IPv6
IPv6 Hop-by-Hop Option	PadN	1	IPv6
IPv6 Destination Options	PadN	1	IPv6
DNS RR Signature	Signature	1	IPv4, IPv6
DNS RRSIG	více polí ‡	4	IPv4, IPv6
DNS response	TXT	1	IPv4, IPv6
mDNS response	Address	1	IPv4, IPv6
DHCP Option	Unassigned	1	IPv4, IPv6
BGP Optional Parameters	Authentication	1	IPv4, IPv6
TCP Options	Unknown	2	IPv4, IPv6
QUICK Handshake	2 Connection IDs	2	IPv4, IPv6
MLDv2 Report	Address exclude	1	IPv6
ICMP ping	datová část	1	IPv4, IPv6

* IPv6 Fragment Header: více polí – Next header, Reserved octet, Offset, Reserved bits, More fragments, Identification

† IPv6 Authentication Header: více polí – Reserved, AH SPI, AH Sequence, AH ICV

‡ DNS RRSIG: více polí – Original TTL, Signature Expiration, Signature Inception, Key Tag

Ve většině případů je skrytý text vkládán přes jedno pole paketu. Například

v BGP Optional Parameters je text vkládán do samostatného pole *Authentication data*, v případě protokolu QUIC do pole *Handshake*, a v rámci mDNS odpovědi do záznamu *AAAA*. V několika málo situacích je však možné, nebo dokonce nutné, text vkládat přes více polí. Například IPv6 Fragment Header má velikost omezenou na pouhých 8 B, je proto žádoucí pro skrytí informace využít všechny tyto bajty, a z toho důvodu je delší řetězec nutné vkládat přes všechna pole fragmentovaného rozšiřujícího záhlaví, tj. konkrétně přes pole *Next header*, *Reserved octet*, *Offset* a *Identification*.

Tab. 4.2: Zjištěné a ověřené limitace délek dat vkládaných do postranního kanálu

Protokol a pole	Min. délka dat	Max. délka dat	Vkládání dat po *
	[B]	[B]	[B]
IPv6 Fragment Header	8	8	-
IPv6 Authentication Header	14	1026	4
IPv6 Routing Header	16	16	16
IPv6 Hop-by-Hop Option	4	255	8
IPv6 Destination Options	4	255	8
DNS RR Signature	1	<MTU 1500	1
DNS RRSIG	14	14	–
DNS TXT	1	255	1
mDNS A/AAAA	4/16	4/16	4/16
DHCP Option	1	255	1
BGP Optional Parameters	1	255	1
TCP Option	1	37	4
QUIC Src.ID + Dst.ID	1+1	255+255	1+1
MLDv2 Report	16	16	16
ICMP ping	1	<MTU 1500	1

* Data je do polí možné vkládat pouze po určitých násobcích B.

V tabulce 4.2 se minimální délkou dat rozumí minimální počet vkládaných B, při kterém bude dané pole mít správný tvar a základní komunikace hlavním kanálem zůstane funkční. Maximální délka určuje horní limit vkládaných B a je uvedena vždy pro jedno pole daného typu. Například jeden textový záznam DNS odpovědi může obsahovat maximálně 255 B. Je však umožněno vložit do paketu více polí zvoleného typu za sebou, a to nejen v případě DNS.

Obdobné platí i pro IPv6 extension headers. Velikosti jednotlivých polí, jako jsou *Address* či *PadN*, jsou sice omezeny, ale ve většině případů je maximální velikost celých rozšiřujících záhlaví až 2048 B, tudíž je možné polí zvoleného typu

do záhlaví vkládat více za sebou. Nicméně jelikož IPv6 standardně neprovádí fragmentaci paketů, je velikost paketu omezena hodnotou MTU, která bývá 1500 B [40]. Do obsahu rozšiřujícího záhlaví je proto možné vkládat takový počet polí (PadN, adres, či jiných), který dovolí limit MTU.

Je nutné podotknout, že hodnoty uvedené v těchto tabulkách vycházejí z parametrů konkrétních paketů generovaných v rámci této práce. Například pro TCP Options: *Experimental - Unknown* je v tomto případě možné vložit 37 B dat. Obecně však možná velikost vkládaných dat závisí na dalších položkách TCP záhlaví, které může být rozšířeno nejvýše na 60 B.

Všechny výše uvedené minimální i maximální hodnoty byly prakticky vyzkoušeny a pakety s těmito objemy dat vytvořeny a též jsou součástí elektronické přílohy B této práce.

4.7 Kombinace možností skrývání informace

V rámci této práce bylo prakticky ověřeno, že lze skrytou informaci vkládat do více částí paketu zároveň. Jednou z variant, jak v rámci jednoho paketu skrýt informaci na více místech zároveň, je použití rozšiřujících IPv6 záhlaví. Podle dokumentu RFC 8200 [40] lze za sebe v určitém pořadí skládat více typů IPv6 extension headers. Je však vždy nutné v každém záhlaví uvést typ záhlaví, které bude následovat. Dle kombinace uvedené v dokumentu RFC 8200 bylo vyzkoušeno hlavičky poskládat a výsledkem je paket s pěti rozšiřujícími hlavičkami, v nichž je možné skrývat data. Poskládání jednotlivých rozšiřujících IPv6 záhlaví znázorňuje obrázek 4.22. Z těchto pěti záhlaví však pro vložení textové informace byla využita pouze záhlaví čtyři. *Fragment Header* totiž povoluje využít v tomto případě pro skrytí dat pouze 4 z 8 B, což se nejvíe blíží dostatečnému.

No.	Time	Source	Destination	Length	Protocol	Info
1	0.000...	fe80::20c:3fff:feba:f...	fe80::20e:4cff:fe3b:d...	217	ICMPv6	Echo (ping) request id=0x04d2, seq=0, hop limit=128 (reply in 2)
2	0.001...	fe80::20e:4cff:fe3b:d...	fe80::20c:3fff:feba:f...	217	ICMPv6	Echo (ping) reply id=0x04d2, seq=0, hop limit=128 (request in 1)
Frame 2: 217 bytes on wire (1736 bits), 217 bytes captured on interface 0 Ethernet II, Src: Bermai_3b:df:ca (00:0e:4c:3b:df:ca), Dst: fe80::20c:3fff:feba:f411 (fe80::20c:3fff:feba:f411) Internet Protocol Version 6, Src: fe80::20e:4cff:fe3b:dfca, Dst: fe80::20c:3fff:feba:f411 0110 = Version: 6 0000 0000 = Traffic Class: 0x00000000 0000 0000 0000 0000 = Flow Label: 0x00000000 Payload Length: 116 Next Header: IPv6 Hop-by-Hop Option (0) Hop Limit: 128 Source Address: fe80::20e:4cff:fe3b:dfca Destination Address: fe80::20c:3fff:feba:f411 [Source SLAAC MAC: Bermai_3b:df:ca (00:0e:4c:3b:df:ca)] [Destination SLAAC MAC: CogentDefenc_ba:f4:11 (00:0c:29:ba:f4:11)] IPv6 Hop-by-Hop Option Destination Options for IPv6 Routing Header for IPv6 (RPL Source Route) Fragment Header for IPv6 Authentication Header Internet Control Message Protocol v6						

Obr. 4.22: Skrytí informace v několika IPv6 rozšiřujících záhlavích

Je také možné upravovat obsah paketu na třech vrstvách síťového modelu TCP/IP zároveň, a to na vrstvě síťové, transportní i aplikační a skrýt tak informaci na třech místech paketu aniž by základní komunikace byla ovlivněna. Následující obrázky 4.23 a 4.24 znázorňují paket *DNS response*. Zobrazeny jsou jednotlivé vrstvy paketu, které zahrnují protokol *IPv6* na vrstvě síťové, *TCP* na vrstvě transportní a *DNSSEC* na vrstvě aplikační. Na každé z těchto vrstev je skrytá část informace. V rámci DNSSEC je informace skrývána na dvou místech. Celkem tak tento paket poskytuje možnost skrýt informaci až na čtyřech místech zároveň.

Obrázek 4.23 ukazuje informaci skrytou na síťové vrstvě v *IPv6 Authentication header* a na vrstvě transportní v *TCP Options - Experimental: Unknown*.

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.0000...	2001:db8:acad::	2001:4860:4860::	DNS	201	Standard query 0x5719 A kom-ppkt.utko.fekt.vut.cz TXT
2	0.0014...	2001:4860:4860::	2001:db8:acad::	DNS	528	Standard query response 0x5719 A kom-ppkt.utko.fekt.vut.cz AAAA 93e5:47e0::1 RRSIG TXT

- Internet Protocol Version 6, Src: 2001:4860:4860::8888, 0110 = Version: 6 0000 0000 = Traffic Class 1001 1000 0111 0110 0010 = Flow Label: 0x98762 - Payload Length: 474 - Next Header: Authentication Header (51) - Hop Limit: 127 - Source Address: 2001:4860:4860::8888 - Destination Address: 2001:db8:acad:a::1 - Authentication Header - Transmission Control Protocol, Src Port: 53, Dst Port: 49154 - Destination Port: 49154 [Stream index: 0] [Conversation completeness: Incomplete (11)] [TCP Segment Len: 406] - Sequence Number: 0 (relative sequence number) - Sequence Number (raw): 810569577 [Next Sequence Number: 407 (relative sequence number)] - Acknowledgment Number: 1 (relative ack number) - Acknowledgment number (raw): 4079825204 1011 = Header Length: 44 bytes (11) - Flags: 0x012 (SYN, ACK) - Window: 65535 [Calculated window size: 65535] - Checksum: 0x3579 [unverified] [Checksum Status: Unverified] - Urgent Pointer: 0 - Options: (24 bytes), Experimental, Maximum segment size - TCP Option - Experimental: Unknown - TCP Option - Maximum segment size: 1460 bytes - TCP Option - Window scale: 8 (multiply by 256) - TCP Option - SACK permitted - TCP Option - End of Option List (EOL)	<pre> 0000 00 0c 29 89 cb 18 00 0c 29 b3 15 bf 86 dd 60 09 ... 0010 87 62 01 da 33 7f 20 01 48 60 48 60 00 00 00 ... 0020 00 00 00 00 88 88 20 01 0d b8 ac ad 00 0a 00 ... 0030 00 00 00 00 00 01 06 04 00 00 00 20 00 00 00 ... 0040 0a 31 2f 34 20 49 4e 46 4f 52 4d 41 43 45 00 ... 0050 c0 02 30 50 4f 69 f3 2d 31 34 b0 12 ff ff 35 79 ... 0060 00 00 fd 0e 32 2f 34 49 4e 46 4f 52 4d 41 43 45 ... 0070 02 04 05 b4 03 03 08 04 02 00 01 94 57 19 81 a0 ... 0080 00 01 00 02 00 00 00 01 08 6b 6f 6d 2d 70 70 6b ... 0090 74 04 75 74 6b 6f 04 66 65 6b 74 03 76 75 74 02 ... 00a0 63 7a 00 00 01 00 01 c0 0c 00 1c 00 01 00 00 37 ... 00b0 25 00 10 93 e5 47 e0 00 00 00 00 00 00 00 00 ... 00c0 00 00 01 c0 0c 00 2e 00 01 00 00 37 25 01 1f 00 ... 00d0 01 07 04 00 00 38 40 65 5f 04 b2 65 37 77 b2 dd ... 00e0 33 04 66 65 6b 74 03 76 75 74 02 63 7a 00 4e 62 ... 00f0 06 b9 93 9b 5d 9b f9 c1 04 d9 1a 09 68 61 3f 17 ... 0100 d7 6d d5 2a 10 6a da 91 cf 98 24 68 c9 81 64 01 ... 0110 3b 15 e9 ae 47 cf fa dd 87 38 29 6c 21 2c 39 cd ... 0120 d0 b3 d2 5a 64 ae b4 8c c2 1d cc e4 8b 60 de 4a ... 0130 10 e0 32 6f 10 aa 00 00 00 00 00 00 00 05 00 ... 0140 e8 00 6a 0f 2d 0f 00 00 00 00 00 00 00 00 00 ... 0150 2d 33 2f 34 20 49 4e 46 4f 52 4d 41 43 45 00 00 ... 0160 00 2d e0 a5 2a 2f 00 00 64 e4 00 e5 7c 85 4d 34 ... 0170 0a 2f 04 1f 5a 15 c8 aa fd bc 62 b3 b7 8f 75 81 ... 0180 30 3d d0 47 9e c5 45 3b 3f fe 1e 6b e2 f3 de 16 ... 0190 55 1b 76 21 7f 40 1b 43 7b ef b3 9e 6a f1 7a 47 ... 01a0 09 d5 22 fc 46 a4 5a 37 01 39 8d f4 17 25 ba 57 ... 01b0 e3 29 34 be 4f c3 2d d8 a5 41 2f ee cf df f7 ea ... 01c0 7b df 92 65 f6 d6 af 3d c5 92 ab 14 ef 80 ec ff ... 01d0 d9 34 ff 28 3c de 81 da 2a c7 d6 37 b5 7f d4 df ... 01e0 c9 d4 e4 ea bb 71 fe 70 12 49 3c 66 da 83 00 00 ... 01f0 10 00 01 00 00 01 00 00 17 16 2d 2d 2d 2d 2d ... 0200 34 2f 34 20 49 4e 46 4f 52 4d 41 43 45 2d 2d 2d </pre>
--	---

Obr. 4.23: Skrytí informace na 3 vrstvách modelu TCP/IP - část 1 (vrstva L3 a L4)

Na obrázku 4.24 je prezentována informace skrytá na dvou místech v rámci vrstvy aplikační, a to v poli *RRSIG* a *Additional records - TXT* protokolu DNS.

Soubory, ve kterých je kombinováno více možností skrytí informace jsou poskytnuty jako příloha C.

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.0000...	2001:db8:acad...	2001:4860:48...	DNS	201	Standard query 0x5719 A kom-ppkt.utko.fekt.vut.cz TXT
2	0.0014...	2001:4860:486...	2001:db8:aca...	DNS	528	Standard query response 0x5719 A kom-ppkt.utko.fekt.vut.cz AAAA 93e5:47e0::1 RRSIG TXT

Transmission Control Protocol, Src Port: 53, Dst Port: Domain Name System (response) Length: 404 Transaction ID: 0x5719 Flags: 0x81a0 Standard query response, No error Questions: 1 Answer RRs: 2 Authority RRs: 0 Additional RRs: 1 Queries Answers kom-ppkt.utko.fekt.vut.cz: type AAAA, class IN, ad kom-ppkt.utko.fekt.vut.cz: type RRSIG, class IN Name: kom-ppkt.utko.fekt.vut.cz Type: RRSIG (46) (Resource Record Signature) Class: IN (0x0001) Time to live: 14117 (3 hours, 55 minutes, 17 sec) Data length: 287 Type Covered: A (1) (Host Address) Algorithm: RSA/SHA1 + NSEC3/SHA1 (7) Labels: 4 Original TTL: 14400 (4 hours) Signature Expiration: Nov 23, 2023 08:52:18.0000 Signature Inception: Oct 24, 2023 09:52:18.0000 Key Tag: 56627 Signer's name: fekt.vut.cz Signature [truncated]: 4e6206b9939b5d9bf9c104d91 Additional records <Root>: type TXT, class IN Request In: 1 [Time: 0.001472641 seconds]	0000 00 0c 29 89 cb 18 00 0c 29 b3 15 bf 86 dd 60 09 ..).....)..... 0010 87 62 01 da 33 7f 20 01 48 60 48 60 00 00 00 ..b.3. .H'..... 0020 00 00 00 00 88 88 20 01 0d b8 ac ad 00 0a 00 00 0030 00 00 00 00 01 06 04 00 00 00 00 20 00 00 00 0040 0a 31 2f 34 20 49 4e 46 4f 52 4d 41 43 45 00 35 ..1/4 INF ORMACE-5 0050 c0 02 30 50 4f 69 f3 2d 31 34 b0 12 ff ff 35 79 ..0Poi-- 14....5y 0060 00 00 fd 0e 32 2f 34 49 4e 46 4f 52 4d 41 43 45 2/4I NFORMACE 0070 02 04 05 b4 03 08 04 02 00 01 94 57 19 81 a0-W... 0080 00 01 00 02 00 00 01 08 6b 6f 6d 2d 70 70 6b kom-ppk 0090 74 04 75 74 6b 6f 04 66 65 6b 74 03 76 75 74 02 t-utko-f ekt-vut- 00a0 63 7a 00 00 01 00 01 c0 0c 00 1c 00 01 00 00 37 cz.....7 00b0 25 00 10 93 e5 47 e0 00 00 00 00 00 00 00 00 %.....G..... 00c0 00 00 01 c0 0c 00 2e 00 01 00 00 37 25 01 1f 00 7%... 00d0 01 07 04 00 00 38 40 65 5f 04 b2 65 37 77 b2 dd 8@e...e7w... 00e0 33 04 66 65 6b 74 03 76 75 74 02 63 7a 00 4e 62 3-fekt-v ut-cz-Nb 00f0 06 b9 93 9b 5d 9b f9 c1 04 d9 1a 09 68 61 3f 17].....ha? 0100 d7 6d d5 2a 10 6a da 91 cf 98 24 68 c9 81 64 01 ..m.*.j...\$h..d- 0110 3b 15 e9 ae 47 cf fa dd 87 38 29 6c 21 2c 39 cd ;..G....8)1!,9 0120 d0 b3 d2 5a 64 ae b4 8c c2 1d cc e4 8b 60 de 4a ..Zd.....J 0130 1e 00 32 6f 10 aa 00 00 00 00 00 00 00 05 00 ..2o..... 0140 e8 00 6a 0f 2d 0f 00 00 00 00 00 00 00 00 00 ..j..... 0150 2d 33 2f 34 20 49 4e 46 4f 52 4d 41 43 45 00 00 -3/4 INF ORMACE 0160 00 2d e0 a5 2a 2f 00 00 64 e4 00 e5 7c 85 4d 34 ...*/...d...M4 0170 0a 2f 04 1f 5a 15 c8 aa fd bc 62 b3 b7 8f 75 81 ...Z....b..u... 0180 30 3d d0 47 9e c5 45 3b 3f fe 1e 6b e2 f3 de 16 0=G..E;?...k.... 0190 55 1b 76 21 7f 40 1b 43 7b ef b3 9e 6a f1 7a 47 U-v!@-C {...j-zG 01a0 09 d5 22 fc 46 a4 5a 37 01 39 8d f4 17 25 ba 57 ...F-Z7 -9...%W 01b0 e3 29 34 be 4f c3 2d d8 a5 41 2f ee cf df f7 ea ..)4-0...A/..... 01c0 7b df 92 65 f6 d6 af 3d c5 92 ab 14 ef 80 ec ff {...e...=..... 01d0 d9 34 ff 28 3c de 81 da 2a c7 d6 37 b5 7f d4 df ..4.(<...*.7.... 01e0 c9 d4 e4 ea bb 71 fe 70 12 49 3c 66 da 83 00 00 q-p .I<f.... 01f0 10 00 01 00 00 01 00 00 17 16 2d 2d 2d 2d 2d 2d ..4. INFO RMACE -- 0200 34 2f 34 20 49 4e 46 4f 52 4d 41 43 45 2d 2d 2d
---	---

Obr. 4.24: Skrytí informace na 3 vrstvách modelu TCP/IP - část 2 (vrstva L7)

4.8 Zadání projektu pro předmět MPC-PKT

Jedním z dalších cílů této práce bylo vytvořit další verzi zadání semestrálního projektu pro studenty absolvující předmět Pokročilé komunikační techniky. Jako vzor byly použity tři verze projektu z roku 2023. Tato nová verze je tudíž obdobná, avšak přizpůsobená aktuálním požadavkům předmětu. Celé zadání i jeho dílčí části jsou poskytnuty v příloze D.

Jako zadání projektu je studentům poskytnut soubor *.pcapng*, který se skládá z několika datových toků zahrnujících například protokoly DNS a mDNS, MLD, či ICMPv6. Pro každý z komunikačních scénářů platí, že alespoň v jednom z paketů je ukrytá konkrétní tajná informace, kterou mohou studenti odhalit.

Postup tvorby paketů byl obdobný jako u jednotlivých variant představených v kapitolách 4.1 až 4.5. Informace obsažené v přenášných packagech jsou snadno modifikovatelné, a proto je z této verze zadání možné vytvořit i verze další.

V zadání jsou nejprve simulovány zprávy *RA*, *RS*, *NA* a *NS* přenášené protokolem ICMPv6 a poté následují *MLD Query* a *MLD Reporty*, v nichž koncová zařízení sdělují svůj zájem o přístup do konkrétní multicastové skupiny. Právě ve zprávě *MLD Query* je na síťové vrstvě v rozšiřujícím IPv6 záhlaví typu *Hop-by-Hop Option* v poli *PadN* skrytý text, jak je znázorněno na obrázku 4.25.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000..	fe80::20c:29ff:fe69:e12a	ff02::2	ICMPv6	62	Router Solicitation
2	0.00116..	fe80::8271:1f00:ac02:7000	ff02::1	ICMPv6	198	Router Advertisement from 80:71:1f:02:70:00
3	0.00236..	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b75:cf99:8977	ICMPv6	86	Neighbor Solicitation for 2001:67c:1220:9892:e990:4b75:cf99:8977
4	0.00344..	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d809:e77:b...	ICMPv6	86	Neighbor Advertisement 2001:67c:1220:9892:e990:4b75:cf99:8977
5	0.00458..	fe80::230:48ff:fe2b:d298	ff02::1:ff02:7000	ICMPv6	142	Multicast Listener Query
6	0.00566..	fe80::256:32dc:fe3a:e125	ff02::1:ff02:7000	ICMPv6	78	Multicast Listener Report
7	0.00768..	fe80::20c:29ff:fe73:b1d7	ff02::1:ff02:7000	ICMPv6	78	Multicast Listener Report

> Frame 5: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on Ethernet II, Src: SuperMicroCo_2b:d2:98 (00:30:48:2b:d2:98), Dst: IPv6mcast Internet Protocol Version 6, Src: fe80::230:48ff:fe2b:d298, Dst: ff02::1:ff02:7000	0000 33 33 ff 02 70 00 00 30 48 2b d2 98 86 dd 60 00 33 - p 0 H+ ...
0110 = Version: 6	0010 00 00 00 58 00 01 fe 80 00 00 00 00 00 02 30 ... X... ..
> 0000 0000 = Traffic Class: 0x00 (DSCP: CS0)	0020 48 ff fe 2b d2 98 ff 02 00 00 00 00 00 00 00 H+ ...
.... 0000 0000 0000 0000 0000 0000 = Flow Label: 0x00000	0030 00 01 ff 02 70 00 3a 07 05 02 00 00 01 38 7c 7c
Payload Length: 88	0040 68 69 64 64 65 6e 20 6d 65 73 73 61 67 65 3a 20 hidden = message;
Next Header: IPv6 Hop-by-Hop Option (0)	0050 77 72 69 74 65 20 69 6e 20 67 6f 6f 67 6c 65 20 write in google
Hop Limit: 1	0060 73 65 61 72 63 68 20 22 6d 65 74 65 6f 72 69 74 search " meteorit
Source Address: fe80::230:48ff:fe2b:d298	0070 65 22 7c 7c 00 00 82 00 60 19 27 10 00 00 ff 02 e" ";
Destination Address: ff02::1:ff02:7000	0080 00 00 00 00 00 00 00 00 00 01 ff 02 70 00
[Source SLAAC MAC: SuperMicroCo_2b:d2:98 (00:30:48:2b:d2:98)]	
> IPv6 Hop-by-Hop Option	
Next Header: ICMPv6 (58)	
Length: 7	
[Length: 64 bytes]	
> Router Alert	
> PadN	
> Type: PadN (0x01)	
Length: 56	
PadN: 7c7c68696464656e206d6573736167653a20777269746520696e20676f65	
> Internet Control Message Protocol v6	

Obr. 4.25: Skrytí informace v rozšiřujícím IPv6 záhlaví

Jako další následují dva DNS dotazy a jim příslušející odpovědi. V první z těchto odpovědí je opět skrytá informace, a to konkrétně v podpisu *RRSIG* jak ilustruje obrázek 4.26.

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.00000...	192.168.1.104	192.168.1.1	DNS	70	Standard query 0x0002 ANY novinky.cz
2	0.00170...	192.168.1.1	192.168.1.104	DNS	234	Standard query response 0x0002 ANY novinky.cz SOA ans.seznam.cz RRSIG
3	0.00311...	192.168.1.104	192.168.1.1	DNS	70	Standard query 0x0004 AAAA novinky.cz
4	0.00452...	192.168.1.1	192.168.1.104	DNS	126	Standard query response 0x0004 AAAA novinky.cz AAAA 2a02:598:a::78:151 AAAA 2a02:598:2::151

> Frame 2: 234 bytes on wire (1872 bits), 234 bytes captured (1872 bits) on	0000	50 e5 49 1a 12 d6 3c 7c 3f 1f bb a4 08 00 45 00	P.I...< ?.....E.
> Ethernet II, Src: ASUSTekCOMPU_1f:bb:a4 (3c:7c:3f:1f:bb:a4), Dst: Gigabyt	0010	00 dc 04 d2 00 00 3f 11 f2 85 c0 a8 01 01 c0 a8	?.....
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.104	0020	01 68 00 35 f6 df 00 c8 0e 64 00 02 81 80 00 01	.h.5.....d.....
> User Datagram Protocol, Src Port: 53, Dst Port: 63199	0030	00 02 00 00 00 00 07 6e 6f 76 69 6e 6b 79 02 63	n ovinky:c
> Domain Name System (response)	0040	7a 00 00 ff 00 01 c0 0c 00 06 00 01 00 00 00 70	p
Transaction ID: 0x0002	0050	00 2e 03 61 6e 73 06 73 65 7a 6e 61 6d c0 14 0a	ans:s eznam...
> Flags: 0x8180 Standard query response, No error	0060	68 6f 73 74 6d 61 73 74 65 72 c0 2c 65 dd c3 80	hostmast er,e...
Questions: 1	0070	00 00 03 84 00 00 01 2c 00 09 3a 80 00 00 01 2c	,.....
Answer RRs: 2	0080	c0 0c 00 2e 00 01 00 00 00 70 00 5e 00 06 0d 02	,p.....
Authority RRs: 0	0090	00 00 01 2c 65 f0 38 80 65 dd ae 68 5d 99 07 6e	...,e.8...e..hl..n
Additional RRs: 0	00a0	6f 76 69 6e 6b 79 02 63 7a 00 d2 1c df 4c 2b a8	ovinky:c z.....l...
> Queries	00b0	8d 97 fb c3 04 7c 7c 68 69 64 64 65 6e 20 6d 65	 h idden me
> Answers	00c0	73 73 61 67 65 3a 20 33 2e 31 35 38 30 4e 2c 20	ssage: 3 .1580N,
> novinky.cz: type SOA, class IN, mname ans.seznam.cz	00d0	31 30 31 2e 37 31 31 38 45 20 7c 7c ca e2 22 16	101.7118 E ...".
> novinky.cz: type RRSIG, class IN	00e0	51 47 e8 27 fe 53 5d 19 ef 00	QG.'s]. ..

Signature: d21cdf4c2ba88d97fbc3047c7c68696464656e206d657373616765

Obr. 4.26: Skrytí informace v DNSSEC RRSIG

Poté jsou v zadání projektu vloženy fragmentované IPv6 pakety. Simulované pakety představují ping komunikaci mezi dvěma koncovými zařízeními. Celkem je odesláno pět *Echo request* zpráv a na ně pět odpovídajících zpráv *Echo reply*. Vždy ve druhém paketu dané Echo request/reply zprávy je součástí obsáhlé datové části jedna pětina skryté informace. Části skryté informace si studenti musejí poskládat, aby získali výsledný řetězec reprezentující odkaz na vyplnění formuláře.

Obrázek 4.27 znázorňuje zachycenou komunikaci a obrázek 4.28 informaci skrytou v datové části.

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.00000...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	IPv6	15...	IPv6 fragment (off=0 more=y ident=0x6de56723 nxt=58)
2	0.00118...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	IPv6	15...	IPv6 fragment (off=1448 more=y ident=0x6de56723 nxt=58)
3	0.00326...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	ICMPv6	174	Echo (ping) request id=0x0001, seq=116, hop limit=128 (reply in 6)
4	0.01332...	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d8...	IPv6	15...	IPv6 fragment (off=0 more=y ident=0xb7f789cf nxt=58)
5	0.01443...	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d8...	IPv6	15...	IPv6 fragment (off=1448 more=y ident=0xb7f789cf nxt=58)
6	0.01657...	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d8...	ICMPv6	174	Echo (ping) reply id=0x0001, seq=116, hop limit=128 (request in 3)
7	1.01663...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	IPv6	15...	IPv6 fragment (off=0 more=y ident=0x6de56725 nxt=58)
8	1.01774...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	IPv6	15...	IPv6 fragment (off=1448 more=y ident=0x6de56725 nxt=58)
9	1.01979...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	ICMPv6	174	Echo (ping) request id=0x0001, seq=117, hop limit=128 (reply in 12)
10	1.02984...	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d8...	IPv6	15...	IPv6 fragment (off=0 more=y ident=0x90c220c5 nxt=58)
11	1.03090...	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d8...	IPv6	15...	IPv6 fragment (off=1448 more=y ident=0x90c220c5 nxt=58)
12	1.03294...	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d8...	ICMPv6	174	Echo (ping) reply id=0x0001, seq=117, hop limit=58 (request in 9)
13	2.04270...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	IPv6	15...	IPv6 fragment (off=0 more=y ident=0x6de56727 nxt=58)

Obr. 4.27: Fragmentované IPv6 pakety

V poslední části zadání projektu je simulována mDNS komunikace ve které klient žádá o překlad lokálních jmen. Na obrázku 4.29 je znázorněna odpověď, kterou klient dostává. V údaji AAAA je skrytá informace.

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.00000...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	IPv6	15...	IPv6 fragment (off=0 more=y ident=0x6de56723 nxt=58)
2	0.00118...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	IPv6	15...	IPv6 fragment (off=1448 more=y ident=0x6de56723 nxt=58)
3	0.00326...	2001:67c:1220:9892:5dc5:d809:e77:b...	2001:67c:1220:9892:e990:4b...	ICMPv6	174	Echo (ping) request id=0x0001, seq=116, hop limit=128
4	0.01332...	2001:67c:1220:9892:e990:4b75:cf99:...	2001:67c:1220:9892:5dc5:d8...	IPv6	15...	IPv6 fragment (off=0 more=y ident=0xb7f789cf nxt=58)

> Frame 2: 1510 bytes on wire (12080 bits), 1510 bytes captured	0460	6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64	l m n o p q r s t u v w a b c d
> Ethernet II, Src: GigaByteTech_1a:12:d6 (50:e5:49:1a:12:d6),	0470	65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74	e f g h i j k l m n o p q r s t
> Internet Protocol Version 6, Src: 2001:67c:1220:9892:5dc5:d809:e77:b...	0480	75 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d	u v w a b c d e f g h i j k l m
▼ Data (1448 bytes)	0490	6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 65 66	n o p q r s t u v a b c d e f
Data [truncated]: 6f70204c4f4f4b20434c4f53454c592c20494e20	04a0	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	g h i j k l m n o p q r s t u v
[Length: 1448]	04b0	77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f	w a b c d e f g h i j k l m n o
	04c0	70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68	p q r s t u v w a b c d e f g h
	04d0	69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61	i j k l m n o p q r s t u v w a
	04e0	62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71	b c d e f g h i j k l m n o p q
	04f0	72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a	r s t u v w a b c d e f g h i j
	0500	6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63	k l m n o p q r s t u v w a b c
	0510	64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73	d e f g h i j k l m n o p q r s
	0520	74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c	t u v w a b c d e f g h i j k l
	0530	6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 65	m n o p q r s t u v a b c d e f
	0540	66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75	f g h i j k l m n o p q r s t u
	0550	61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70	a b c d e f g h i j k l m n o p
	0560	71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69	q r s t u v a b c d e f g h i j
	0570	6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62	j k l m n o p q r s t u v w a b
	0580	63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72	c d e f g h i j k l m n o p q r
	0590	73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b	s t u v w a b c d e f g h i j k
	05a0	6c 6d 20 50 61 72 74 20 31 2f 35 3a 20 68 74 74	l m Part 1/5: htt
	05b0	70 73 3a 2f 2f 66 6f 72 6d 73 2e 6f 66 66 69 63	ps://for ms.offic
	05c0	65 2e 63 6f 6d 2f 50 61 67 65 73 2f 44 65 73 69	e.com/Pa ges/Desi
	05d0	67 6e 50 61 67 65 56 32 2e 61 73 70 78 3f 73 75	gnPageV2 .aspx?su
	05e0	62 70 61 67 65 3d	bpage=

Obr. 4.28: Skrytí informace v datové části IPv6 paketu

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.00000...	192.168.1.104	224.0.0.251	MDNS	92	Standard query 0x0006 ANY DESKTOP-USERJ07.local, "QM" question OPT
2	0.00152...	192.168.1.10	224.0.0.251	MDNS	147	Standard query response 0x0006 AAAA 7564:616c:6f73:7420:312e:312e:3139:3933 AAAA fe80::
3	0.05058...	192.168.1.104	224.0.0.251	MDNS	108	Standard query 0x003a A LAPTOP-81HAGIR1.local, "QU" question AAAA LAPTOP-81HAGIR1.local
4	0.06208...	192.168.1.156	192.168.1.104	MDNS	147	Standard query response 0x003a A 147.229.147.98 AAAA 2001:67c:1220:9892:354a:1431:8626:

> Frame 2: 147 bytes on wire (1176 bits), 147 bytes captured (1176	0000	01 00 5e 00 00 fb dc 68 0c 3f 4f 9f 08 00 45 00	...h...?0...E...
> Ethernet II, Src: HewlettPacka_3f:4f:9f (dc:68:0c:3f:4f:9f), Dst:	0010	00 85 04 d2 00 00 ff 11 13 e8 c0 a8 01 0a e0 00	
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 224.0.0.251	0020	00 fb 14 e9 14 e9 00 71 7a 17 00 06 84 00 00 00	q z.....
> User Datagram Protocol, Src Port: 5353, Dst Port: 5353	0030	00 03 00 00 00 0f 44 45 53 4b 54 4f 50 2d 55	D ESKTOP-U
▼ Multicast Domain Name System (response)	0040	53 45 52 4a 30 37 05 6c 6f 63 61 6c 00 00 1c 00	SERJ07:l ocal....
Transaction ID: 0x0006	0050	01 00 00 00 3c 00 10 75 64 61 6c 6f 73 74 20 31	u dalost 1
> Flags: 0x8400 Standard query response, No error	0060	2e 31 2e 31 39 39 33 c0 0c 00 1c 00 01 00 00 00	...1.1993.....
Questions: 0	0070	3c 00 10 fe 80 00 00 00 00 00 00 b6 3b f5 5c b7	<.....;.\...
Answer RRs: 3	0080	62 47 12 c0 0c 00 01 00 01 00 00 00 3c 00 04 93	b6.....<...
Authority RRs: 0	0090	e5 92 9c	...
Additional RRs: 0			
▼ Answers			
▼ DESKTOP-USERJ07.local: type AAAA, class IN, addr 7564:616c:			
Name: DESKTOP-USERJ07.local			
Type: AAAA (28) (IPv6 Address)			
.000 0000 0000 0001 = Class: IN (0x0001)			
0... = Cache flush: False			
Time to live: 60 (1 minute)			
Data length: 16			
AAAA Address: 7564:616c:6f73:7420:312e:312e:3139:3933			

Obr. 4.29: Skrytí informace v mDNS odpovědi

4.9 Shrnutí

Při tvorbě a generování nestandardních paketů byly zjištěny nejen limitace velikostí vkládaných dat a počtu polí, přes která lze data vkládat, ale také několik málo limitací samotného nástroje Ostinato.

Již dříve bylo zmíněno, že hlavní výhodou Ostinato je možnost volit protokoly na jednotlivých vrstvách modelu TCP/IP dle preference. Jak ukazuje obrázek 4.30, u každého protokolu je možné vyplnit hodnoty či zvolit typy polí, která mají být pro tvorbu paketu použita. Benefitem této funkce je, že pokud není zaškrtnuta možnost "Override", jsou vybrané hodnoty, jako je délka paketu (*Length*) či kontrolní součet (*Checksum*), automaticky přizpůsobovány provedeným změnám.

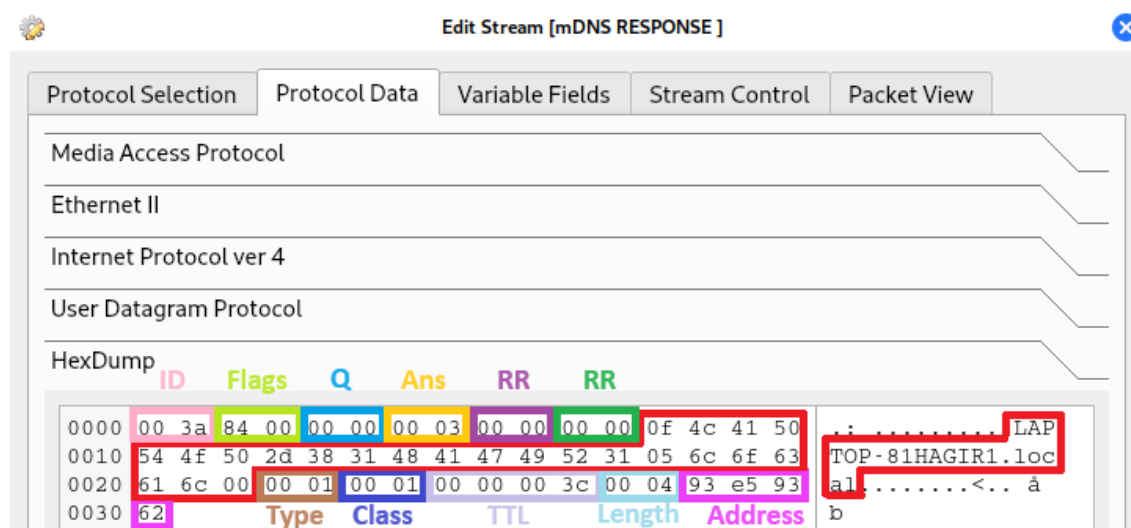
The screenshot shows the 'Edit Stream [mDNS RESPONSE]' window. The 'Protocol Data' tab is selected. The protocol stack is as follows:

- Media Access Protocol
- Ethernet II
- Internet Protocol ver 4
 - ☐ Override Version: 4
 - ☐ Override Header Length (x4): 5
 - DSCP: cs0, Not-ECT
 - ☐ Override Length: 133
 - Identification: 04 D2
 - Fragment Offset (x8): 0
 - ☐ Don't Fragment
 - ☐ More Fragments
 - Time To Live (TTL): 255
 - ☐ Override Protocol: 11
 - ☐ Override Checksum: 32 41
 - Source: 192.168.1.156, Mode: Fixed, Count: 16, Mask: 255.255.255.0
 - Destination: 192.168.1.104, Mode: Fixed, Count: 16, Mask: 255.255.255.0
 - Options: (empty)
- User Datagram Protocol
- HexDump

Obr. 4.30: Volba protokolů na jednotlivých vrstvách TCP/IP (Ostinato)

Naopak jedním ze záporů Ostinato je, že na aplikační vrstvě si protokoly volit nelze, tzn. reprezentaci aplikačních protokolů je nutné vytvořit od základu s pomocí HexDump polí. Například protokol mDNS či ICMPv6 zprávu typu 143 - *MLDv2 Report* Ostinato nenabízí. Je proto nutné vložit na místo aplikační vrstvy pole HexDump a v něm hexadecimálními hodnotami reprezentovat typ pole, jeho délku, porty a další potřebné údaje. Hexadecimální zápis je méně přehledný jak je vidět

na obrázku 4.31, který barevně znázorňuje jednotlivé hodnoty polí mDNS odpovědi.



Obr. 4.31: Hexadecimální reprezentace protokolu (Ostinato)

Pokud má být některá z těchto hodnot změněna, je třeba najít odpovídající hexadecimální znak v poli HexDump a ten přepsat. Například údaj *Checksum* se mění s každým jedním změněným bajtem dat, a tak je nutné jeho hodnotu při každé změně ručně přepsat. Obdobně je nutné měnit i délku paketu a další variabilní hodnoty.

Je také nutné myslet na to, že při úpravě polí paketů tak, aby do nich bylo možné vložit skrytý text, nestačí upravit pouze konkrétní pole, které bude informaci obsahovat, ale je nutné upravovat také předcházející či další související pole. V případě, kdy je reprezentace protokolu tvořena pouze pomocí polí HexDump a jednotlivé údaje jsou zapsány pomocí hexadecimálních znaků, se může často stát, že je některá hodnota přehlédnutá a její změna opomenutá. Je proto vhodné si znaky v polích HexDump rozdělit tak, že vždy jedno pole HexDump bude obsahovat pouze znaky reprezentující jedno z polí paketu, například délku.

Další z nevýhod či technických nedostatků Ostinato je nemožnost vrácení úprav. Neexistuje zde žádná funkce pro návrat do předchozího kroku či možnost průběžného ukládání. Ostinato je jednoduchý nástroj, který však pro účely generování nestandardních datových toků posloužil velmi dobře.

Jak je zřejmé ze srovnávací tabulky 4.2, do různých typů polí lze vkládat různé velké objemy dat. Pokud je potřeba skrýt menší množství dat v rozmezí 1 až 50 B, je vhodné použít například varianty IPv6 *Fragment Header*, TCP *Option*, či mDNS záznam A/AAAA. Všechny uvedené varianty mají však své další limity. Například velikost dat vkládaných do TCP *Option* je ovlivňována dalšími položkami TCP

záhlaví, které nesmí přesáhnout velikost 60 B přičemž pro volitelné položky je vyhrazeno maximálně 40 B.

Pro skrytí obsáhlejších dat až do rozsahu 255 B lze zvolit variantu DNS *Additional records* či BGP *Optional Parameters*. Je též možné využít IPv6 *Routing Header* či MLDv2 *Report*. V rámci těchto protokolů se data mohou vkládat do IPv6 adres po šestnácti bajtech, nicméně je umožněno vložit těchto 16 B dlouhých záznamů několik za sebou.

Ostatní varianty, které zahrnují další typy rozšiřujících IPv6 záhlaví, DNSSEC podpis *RRSIG*, zprávu *Handshake* protokolu QUIC, či ICMPv6 *ping* jsou vhodné pro vkládání dat o velikostech od 500 B až po limit MTU.

U některých variant je nutné vkládat data pouze po určitých násobcích bajtů, čímž je vyžadováno neustálé přepočítávání celkové délky paketu a to může být do jisté míry omezující. Snadno použitelnou variantou jež tuto kontrolu nevyžaduje je například *TXT* pole, které může být použito v rámci DNS odpovědi, nebo pole *PadN* v IPv6 Destination Options. Taková pole mohou pojmout až 255 B, a navíc může být těchto polí vkládáno více za sebou.

5 Testování paketů na reálné síti

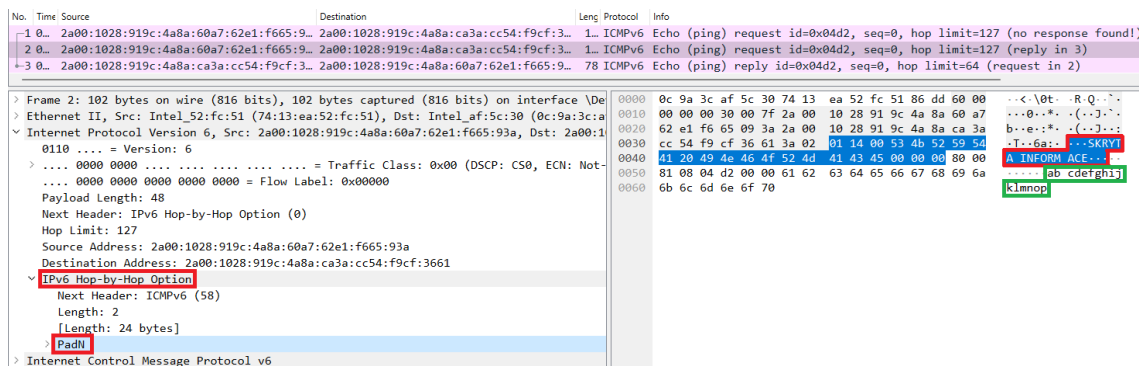
Hlavním cílem této diplomové práce bylo vytvořit několik variant nestandardních datových toků. Pro tento účel bylo využito virtualizační prostředí VMware Workstation. Pomocí nástroje Ostinato byly generovány celé datové toky, v nichž určité pakety obsahovaly skrytou informaci a správnost simulace těchto datových toků byla ověřována nástrojem Wireshark. Vzhledem k tomu, že vygenerované pakety obsahují i skrytá vložená data, naskytla se otázka, zda takto upravené pakety budou správně fungovat i na reálné síti. Testování vygenerovaných nestandardních paketů na reálné bezdrátové LAN síti mezi dvěma počítači se věnuje právě tato kapitola.

Pro testování byly využity dva laptopy připojené k domácí Wi-Fi síti. Z nástroje Ostinato byly generovány pakety a jejich zachycení a sledování odezvy na ně bylo provedeno nástrojem Wireshark. Následující kapitoly 5.1 až 5.6 prezentují některé varianty nestandardních paketů se skrytou informací, které reálnou sítí mezi dvěma zařízeními prošly, tzn. byla na ně přijata odezva. Testování probíhalo pro jednotlivé varianty na základě stejného principu.

Kapitoly 5.6 a 5.7 stručně popisují proces testování, ve kterém bylo ověřováno, zda Laptop 2 dokáže zachytit paket generovaný v nástroji Ostinato instalovaném na zařízení Laptop 1. Otestované pakety zachycené v nástroji Wireshark jsou poskytnuty jako příloha E.

5.1 ICMPv6 ping - IPv6 Hop-by-Hop Option

Jako jedno z prvních proběhlo testování *Echo request* a *Echo reply* zpráv mezi dvěma zařízeními. Nejprve byla dostupnost zařízení ověřena pomocí příkazového řádku a standardního pingu a následně byla v nástroji Ostinato vygenerována zpráva *Echo request* s informací skrytou v rozšiřujícím IPv6 záhlaví typu *Hop-by-Hop Option*, jak lze vidět na obrázku 5.1.



Obr. 5.1: Zpráva Echo request se skrytou informací

Toto záhlaví se běžně v paketu tohoto typu nenachází, nicméně zde bylo hlavním účelem pouze prokázat, že vygenerovaný paket s takto nestandardně skrytou informací nebude při odeslání reálným síťovým rozhraním odfiltrován z hlediska odeslání a obdobně pak na druhé straně nebude odfiltrován při přijetí reálným zařízením a operačním systémem a že na ping dané zařízení zareaguje. Odpověď *Echo reply* v tomto případě nebyla uměle generována, ale odeslal ji sám operační systém. Na podobném principu probíhalo i testování dalších variant. Obrázek 5.2 ukazuje, že v odpovědi chybí IPv6 rozšiřující záhlaví, které se ale běžně nepřeposílá a že datová část odpovědi je shodná s datovou částí žádosti.

No.	Time	Source	Destination	Length	Protocol	Info
1	0.	2a00:1028:919c:4a8a:60a7:62e1:f665:9...	2a00:1028:919c:4a8a:ca3a:cc54:f9cf:3...	1.	ICMPv6	Echo (ping) request id=0x04d2, seq=0, hop limit=127 (no response found!)
2	0.	2a00:1028:919c:4a8a:60a7:62e1:f665:9...	2a00:1028:919c:4a8a:ca3a:cc54:f9cf:3...	1.	ICMPv6	Echo (ping) request id=0x04d2, seq=0, hop limit=127 (reply in 3)
3	0.	2a00:1028:919c:4a8a:ca3a:cc54:f9cf:3...	2a00:1028:919c:4a8a:60a7:62e1:f665:9...	78	ICMPv6	Echo (ping) reply id=0x04d2, seq=0, hop limit=64 (request in 2)

> Frame 3: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \DeviceNPF{...} > Ethernet II, Src: Intel_af:5c:30 (0c:9a:3c:af:5c:30), Dst: Intel_52:fc:51 (74:13:ea:52:fc:51) > Internet Protocol Version 6, Src: 2a00:1028:919c:4a8a:ca3a:cc54:f9cf:3661, Dst: 2a00:1028:919c:4a8a:60a7:62e1:f665:93a > Internet Control Message Protocol v6	0000 74 13 ea 52 fc 51 0c 9a 3c af 5c 30 86 dd 60 00 t:R Q...<\0... 0010 00 00 00 18 3a 40 2a 00 10 28 91 9c 4a 8a ca 3a@*...(-J... 0020 cc 54 f9 cf 36 61 2a 00 10 28 91 9c 4a 8a 60 a7 -T-6a*...(-J... 0030 62 e1 f6 65 09 3a 81 00 80 08 04 d2 00 00 61 62 b...:.....ab 0040 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 cdefghij klmnop
--	--

Obr. 5.2: Odpověď Echo reply

5.2 DNS - IPv6 Fragment Header

Dále byl v nástroji Ostinato generován DNS dotaz na překlad názvu zařízení na jeho IP adresu. Z důvodu testování byl tento DNS dotaz s informací skrytou v *IPv6 Fragment Header*, ilustrovaný na obrázku 5.3, zaslán přímo na dotazované zařízení, které v daném okamžiku pro jednoduchost představuje zároveň i DNS server. Odpověď obsahující požadovanou informaci byla opět získána. I v tomto případě *IPv6 Fragment Header* v odpovědi nebylo použito.

No.	Time	Source	Destination	Length	Protocol	Info
1	0.	2a00:1028:919c:4a8a:60a7:62e1:f665:93a	2a00:1028:919c:4a8a:828c:97ff:fedb:1b62	100	DNS	Standard query 0x0002 A Laptop-J2399
2	0.	2a00:1028:919c:4a8a:60a7:62e1:f665:93a	2a00:1028:919c:4a8a:828c:97ff:fedb:1b62	100	DNS	Standard query 0x0002 A Laptop-J2399
3	0.	2a00:1028:919c:4a8a:828c:97ff:fedb:1b62	2a00:1028:919c:4a8a:60a7:62e1:f665:93a	108	DNS	Standard query response 0x0002 A Laptop-J2399 A 10.0.1.18

> Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface Intel_52:fc:51 (74:13:ea:52:fc:51), Dst: KaonGroup_db:1b:62 > Ethernet II, Src: Intel_52:fc:51 (74:13:ea:52:fc:51), Dst: KaonGroup_db:1b:62 > Internet Protocol Version 6, Src: 2a00:1028:919c:4a8a:60a7:62e1:f665:93a, Dst: 2a00:1028:919c:4a8a:828c:97ff:fedb:1b62 > User Datagram Protocol, Src Port: 49281, Dst Port: 53 > Domain Name System (query)	0000 80 8c 97 db 1b 62 74 13 ea 52 fc 51 86 dd 60 00 bt..R Q... 0010 00 00 00 2e 2c 40 2a 00 10 28 91 9c 4a 8a 60 a7, @*...(-J... 0020 62 e1 f6 65 09 3a 2a 00 10 28 91 9c 4a 8a 82 8c b...e;*...(-J... 0030 97 ff fe db 1b 62 11 00 00 00 49 4e 46 4f c0 81 t...-INFO 0040 00 35 00 26 10 91 00 02 01 00 00 01 00 00 00 00 -5-&..... 0050 00 00 0c 4c 61 70 74 6f 70 2d 4a 32 33 39 39 00 ...Lapto p-J2399 0060 00 01 00 01
---	---

Obr. 5.3: DNS dotaz se skrytou informací

5.3 DNS - Additional Records

Také bylo testováno, zda lokální síť mezi dvěma zařízeními bez problému projde DNS dotaz s informací skrytou v *TXT* poli v části Additional records. Additional records jsou záznamy, které se standardně vyskytují pouze v DNS odpovědi. Nicméně jak znázorňuje obrázek 5.4, i tento dotaz s nestandardně skrytou informací fungoval a odezva na něj byla též přijata.

No.	Time	Source	Destination	Length	Protocol	Info
1	0...	192.168.1.103	192.168.1.1	98	DNS	Standard query 0x0004 AAAA seznam.cz TXT
2	0...	192.168.1.103	192.168.1.1	98	DNS	Standard query 0x0004 AAAA seznam.cz TXT
3	0...	192.168.1.1	192.168.1.103	125	DNS	Standard query response 0x0004 AAAA seznam.cz AAAA 2a02:598:a::79:222 AAAA 2a02:598:2::1222

> Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784	0000	38 6b 1c 56 77 dc 74 13	ea 52 fc 51 08 00 45 00	8k·Vw·t· ·R·Q··E·
> Ethernet II, Src: Intel_52:fc:51 (74:13:ea:52:fc:51), Dst: M	0010	00 54 3f f6 00 00 80 11	76 ea c0 a8 01 67 c0 a8	·T?·...· v...·g·
> Internet Protocol Version 4, Src: 192.168.1.103, Dst: 192.16	0020	01 01 00 00 00 35 00 40	52 23 00 04 01 00 00 01	·...·5·@ R#·...·
> User Datagram Protocol, Src Port: 0, Dst Port: 53	0030	00 00 00 00 00 01 06 73	65 7a 6e 61 6d 02 63 7a	·...·s eznam·cz
> Domain Name System (query)	0040	00 00 1c 00 01 00 00 10	00 01 00 00 01 00 00 12	·...·
Transaction ID: 0x0004	0050	11 53 4b 52 59 54 41 20	49 4e 46 4f 52 4d 41 43	·SKRYTA·INFORMAC
> Flags: 0x0100 Standard query	0060	45 20		E
Questions: 1				
Answer RRs: 0				
Authority RRs: 0				
Additional RRs: 1				
> Queries				
> Additional records				
> <Root>: type TXT, class IN				
Name: <Root>				
Type: TXT (16) (Text strings)				
Class: IN (0x0001)				
Time to live: 256 (4 minutes, 16 seconds)				
Data length: 18				
TXT length: 17				
TXT: SKRYTA INFORMACE				
[Response In: 3]				

Obr. 5.4: DNS dotaz se záznamem Additional records

5.4 mDNS - IPv6 Destination Options

Obdobně jako u DNS dotazu, i v případě mDNS byl zaslán dotaz na překlad jména *Laptop-J2399* s tou změnou, že daný dotaz byl zaslán na multicastovou adresu, jak lze vidět na obrázku 5.5. V dotazu je v poli *IPv6 Destination Options* taktéž skrytá textová informace. Odpověď na dotaz, zobrazená na obrázku 5.6, byla úspěšně obdržena.

5.5 LLMNR - IPv6 Routing Header

V případě LLMNR byl opět generován požadavek na překlad lokálního jména. Dotaz byl zaslán na multicastovou adresu ff02::1:3 určenou pro LLMNR. Informace skrytá v dotazu v rozšiřujícím záhlaví *IPv6 Routing Header* neměla vliv na zpracování a požadovaná odpověď na tento paket byla stejně jako u mDNS vrácena.

No.	Time	Source	Destination	Length	Protocol	Info
1	0...	fe80::86e8:f548:7e25:575c	ff02::fb	122	MDNS	Standard query 0x0000 A Laptop-J2399.local, "QM" question
2	0...	fe80::86e8:f548:7e25:575c	ff02::fb	122	MDNS	Standard query 0x0000 A Laptop-J2399.local, "QM" question
3	0...	fe80::33a2:22ea:a0db:db1f	ff02::fb	108	MDNS	Standard query response 0x0000 A 10.0.1.18

> Frame 1: 122 bytes on wire (976 bits), 1	0000	33 33 00 00 00 fb 74 13	ea 52 fc 51 86 dd 60 0c	33.....t..R.Q...
> Ethernet II, Src: Intel_52:fc:51 (74:13:	0010	bb 2f 00 44 3c 7f fe 80	00 00 00 00 00 00 86 e8	./D<... ..
> Internet Protocol Version 6, Src: fe80::	0020	f5 48 7e 25 57 5c ff 02	00 00 00 00 00 00 00 00	..H~%W\... ..
0110 = Version: 6	0030	00 00 00 00 00 fb 11 02	01 14 53 4b 52 59 54 41	SKRYTA
> 0000 0000	0040	20 49 4e 46 4f 52 4d 41	43 45 00 00 00 00 14 e9	INFORMA CE....
.... 1100 1011 1011 0010 1111 = Flow L	0050	14 e9 00 2c 40 f4 00 00	00 00 00 01 00 00 00 00	...@... ..
Payload Length: 68	0060	00 00 0c 4c 61 70 74 6f	70 2d 4a 32 33 39 39 05	...Lapto p-J2399
Next Header: Destination Options for I	0070	6c 6f 63 61 6c 00 00 01	00 01	local... ..
Hop Limit: 127				
Source Address: fe80::86e8:f548:7e25:5				
Destination Address: ff02::fb				
> Destination Options for IPv6				
> User Datagram Protocol, Src Port: 5353,				
> Multicast Domain Name System (query)				

Obr. 5.5: mDNS dotaz se skrytou informací

No.	Time	Source	Destination	Length	Protocol	Info
1	0....	fe80::86e8:f548:7e25:575c	ff02::fb	122	MDNS	Standard query 0x0000 A Laptop-J2399.local, "QM" question
2	0....	fe80::86e8:f548:7e25:575c	ff02::fb	122	MDNS	Standard query 0x0000 A Laptop-J2399.local, "QM" question
3	0....	fe80::33a2:22ea:a0db:db1f	ff02::fb	108	MDNS	Standard query response 0x0000 A 10.0.1.18

> Frame 3: 108 bytes on wire (864 bits),	0000	33 33 00 00 00 fb 0c 9a	3c af 5c 30 86 dd 60 01	33.....<\0...
> Ethernet II, Src: Intel_af:5c:30 (0c:9a:	0010	6d 25 00 36 11 01 fe 80	00 00 00 00 00 00 33 a2	m%6... ..3..
> Internet Protocol Version 6, Src: fe80::	0020	22 ea a0 db db 1f ff 02	00 00 00 00 00 00 00 00	"... ..
> User Datagram Protocol, Src Port: 5353,	0030	00 00 00 00 00 fb 14 e9	14 e9 00 36 30 b9 00 00	60...
> Multicast Domain Name System (response)	0040	84 00 00 00 00 01 00 00	00 00 0c 4c 61 70 74 6f	Lapto
Transaction ID: 0x0000	0050	70 2d 4a 32 33 39 39 05	6c 6f 63 61 6c 00 00 01	p-J2399 local...
> Flags: 0x8400 Standard query response	0060	00 01 00 00 00 3c 00 04	0a 00 01 12	<... ..
Questions: 0				
Answer RRs: 1				
Authority RRs: 0				
Additional RRs: 0				
> Answers				
[Unsolicited: True]				

Obr. 5.6: Odpověď na mDNS dotaz

5.6 ICMPv6 ping - IPv6 Hop-by-Hop Option

Tento způsob testování byl proveden za použití dvou zařízení připojených ke stejné Wi-Fi síti. Ze zařízení Laptop 2 s adresou 2a00:1028:919c:4a8a:828c:97ff:fedb:1b62 byl zaslán na Laptop 1 s adresou 2a00:1028:919c:4a8a:60a7:62e1:f665:93a ping z příkazového řádku. Ve srovnání s prvním přístupem prezentovaným v této kapitole, byla v tomto případě odpověď *Echo reply* ze zařízení Laptop 1 vygenerována pomocí nástroje Ostinato. I takto vygenerovaná odpověď poslaná z nástroje Ostinato spuštěném na Laptopu 1 byla na Laptopu 2 úspěšně přijata. Skrytá informace se nacházela v rozšiřujícím IPv6 záhlaví *Hop-by-Hop Option* a také v datové části ICMPv6.

5.7 UDP - IPv6 Authentication Header

Obdobně jako ping bylo vyzkoušeno i zaslání čistého UDP paketu. UDP paket byl vygenerován v nástroji Ostinato na zařízení Laptop 1 a přijat na zařízení Laptop 2. Informace byla skrytá opět v rozšiřujícím IPv6 záhlaví typu *Authentication Header* a v datové části. Obsah odeslaného a přijatého paketu byl identický. V tomto případě na zařízení Laptop 2 neběžel žádný proces, který by na paket odpověděl. Lze si však představit například aplikaci pro DNS dotazy, které jsou běžně přenášeny přes UDP.

5.8 Shrnutí

I přesto, že se rozšiřující IPv6 záhlaví v těchto typech paketů běžně nevyskytují, pakety ke svému příjemci došly a úspěšně byly na vysílajícím zařízení přijaty i příslušné odezvy. Tím byl splněn požadovaný účel, tedy otestování, že i některé pakety s nestandardně skrytou datovou informací reálnou bezdrátovou lokální sítí mohou projít.

Závěr

Teoretická část diplomové práce popisuje strukturu síťového modelu TCP/IP a některé komunikační protokoly použité na jednotlivých vrstvách sítě. Podstatné jsou pro tuto práci protokoly, které jsou zahrnuty v osnovách bakalářského předmětu Komunikační technologie a magisterského předmětu Pokročilé komunikační techniky. Cílem je totiž vytvoření částečného a snadno modifikovatelného zadání semestrálního projektu pro studenty, kteří jeden z těchto předmětů absolvují.

Praktická část je realizována ve virtualizačním prostředí VMware a za využití nástroje Ostinato pro generování paketů a Wireshark pro zachycení a analýzu vytvořených datových toků. Pod spojením *generování nestandardních paketů* lze rozumět generování takových paketů, jež obsahují skrytou informaci. Taková informace je skryta v částech záhlaví, kde by nemusela být očekávána.

Pro generování jednotlivých paketů byl využit program Ostinato, jehož hlavní výhodou je možnost volby téměř libovolného protokolu na každé vrstvě modelu TCP/IP a též možnost přizpůsobení parametrů. Data je možné upravovat či přidávat pomocí polí HexDump. Ta mohou být vkládána libovolně mezi jednotlivé vrstvy a jejich obsah je také volitelný. Jsou tedy vhodnou možností pro skrytí informace.

Prvním krokem v procesu návrhu sestav paketů byla podrobná analýza vybraných komunikačních protokolů v programu Wireshark. Byla prozkoumána jednotlivá pole zachycené komunikace a jejich obsah. Tímto způsobem byly nalezeny potenciální možnosti pro skrytí informace. Poté bylo třeba v programu Ostinato vygenerovat různé scénáře komunikací využívajících pro přenos dat právě tyto vybrané protokoly, přičemž bylo klíčové obsah paketů reprezentovaný hexadecimálními hodnotami do jednotlivých polí rozdělovat tak, aby bylo možné některé konkrétní hodnoty snadno upravovat. Po vytvoření bylo nutné zkontrolovat funkčnost těchto datových toků pomocí nástroje Wireshark. Pokud byly vytvořené pakety programem Wireshark zobrazeny dle požadavků a bez výpisu chybových hlášení, následovala fáze vkládání skryté hodnoty. To bylo docíleno vkládáním a upravováním polí HexDump. Každá změna v obsahu paketu byla opět monitorována pomocí programu Wireshark.

Tato diplomová práce navrhuje několik variant pro skrytí informace v rámci datových toků různých komunikačních protokolů modelu TCP/IP a dva datové toky, které kombinují více vybraných variant, čímž prokazují, že v jednom paketu lze skrýt informaci na několika místech, tzn. na internetové, transportní i aplikační vrstvě zároveň. Díky bezproblémové kombinovatelnosti a modifikovatelnosti poskytnutých řešení mohou být z těchto variant sestrojeny jiné formy paketů. Též byla vytvořena další verze projektu pro studenty předmětu MPC-PKT jako ukázka celé sady paketů různých protokolů. Návrh projektu je tvořen několika datovými toky, přičemž

v každém z nich je skryt jiný typ informace. K vygenerovaným datovým tokům je poskytnut i podrobný popis dílčích variant. V textu této práce jsou představeny jen některé varianty vytvořených paketů. Kompletní sestava datových provozů je poskytnutá jako příloha 5.8.

Za účelem srovnání jednotlivých variant byly též testovány a prakticky ověřovány limitace minimální a maximální velikosti jednotlivých polí využitých pro vkládání skryté informace. Jak shrnují přehledové tabulky 4.1 a 4.2, limitace se u jednotlivých variant značně liší. Mnohdy je data nutné vkládat po určitých násobcích B a maximální délka vkládaných dat může být omezená. Z toho vyplývá, že každá varianta je vyhovující pro odlišný účel a velikost skrývaných dat.

Taktéž byly některé z vytvořených nestandardních datových toků testovány na reálné bezdrátové lokální síti mezi dvěma zařízeními. Cílem testování bylo zjistit, zda pakety s nestandardně skrytou informací, nebudou na reálné síti na straně odesílatele či příjemce odfiltrovány. Testování zvolených variant bylo úspěšné, a proto jsou pakety vygenerované v nástroji *Ostinato* považovány za užitečné.

Literatura

- [1] JEŘÁBEK, J. *Pokročilé komunikační techniky*. Skriptum FEKT, Vysoké učení technické v Brně. 2022.
- [2] OSTERLOH, H. *TCP/IP - Kompletní průvodce*. Praha: SoftPress, 2002.
- [3] KUROSE, James F. a Keith W. ROSS. *Computer networking: a top-down approach*. Seventh edition. Essex: Pearson, 2017. ISBN 1-292-15359-8.
- [4] IPCisco.com. *Ethernet Basics*. Online. 2018. Dostupné z: <https://ipcisco.com/ethernet-basics/>. [cit. 29. 9. 2023].
- [5] MOLENAAR, R. *CCNP route*. Online. Dostupné z: <https://networklessons.com/cisco/ccnp-route/ipv4-packet-header>. [cit. 1. 10. 2023].
- [6] TutorialsPoint.com. *IPv6 - Headers*. Online. 2022. Dostupné z: https://www.tutorialspoint.com/ipv6/ipv6_headers.htm. [cit. 1. 10. 2023].
- [7] HINDEN, R., DEERING, S. *IP Version 6 Addressing Architecture*. Online. 2006. Dostupné z: <https://www.rfc-editor.org/rfc/rfc4291.html#page-2>. [cit. 1. 10. 2023].
- [8] PHAN, V. *Generation of IPv6 and ICMPv6 packets and their impact on the operation of network devices*. Online. 2023. Brno: VUT v Brně. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=252768. [cit. 1. 10. 2023].
- [9] GeeksforGeeks.org. *Internet Control Message Protocol (ICMP)*. Online. 2023. Dostupné z: <https://www.geeksforgeeks.org/internet-control-message-protocol-icmp/>. [cit. 2. 10. 2023].
- [10] NARTEN, T., NORDMARK, E., SIMPSON, W., SOLIMAN, H. *Neighbor Discovery for IP version 6 (IPv6)*. Online. 2007. Dostupné z: <https://www.rfc-editor.org/rfc/rfc4861>. [cit. 10. 10. 2023].
- [11] JEŘÁBEK, J. *Komunikační technologie*. Skriptum FEKT, Vysoké učení technické v Brně. 2022.
- [12] Cisco.com. *IP Multicast Technology Overview*. Online. 2021. Dostupné z: https://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html. [cit. 12. 10. 2023].

- [13] Cisco.com. *IPv6 Multicast Listener Discovery Protocol*. Online. 2012. Dostupné z:
https://www.cisco.com/en/US/docs/ios-xml/ios/ipmulti_lsm/configuration/15-0sy/ipv6-mcast-mld-xe.html. [cit. 12. 10. 2023].
- [14] LIANG, K. *Explaining TCP/IP*. Online. 2018. Dostupné z:
https://kelvin.ink/2018/06/18/ExplainingTCP_IP/. [cit. 2. 10. 2023].
- [15] CHESHIRE, S., KROCHMAL, M. *Multicast DNS*. Online. 2013. Dostupné z:
<https://www.rfc-editor.org/rfc/rfc6762#appendix-D>. [cit. 12. 3. 2024].
- [16] ABOBA, B., THALER, D., ESIBOV, L. *Link-Local Multicast Name Resolution (LLMNR)*. Online. 2007. Dostupné z:
<https://www.rfc-editor.org/rfc/rfc4795#section-1.2>. [cit. 12. 3. 2024].
- [17] CILLI, C., CISA, CISM, CRISC, CGEIT. *Understanding Covert Channels of Communication*. Online. 2017. Dostupné z:
<https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2017/understanding-covert-channels-of-communication>. [cit. 9. 11. 2023].
- [18] WU, Z. *Information Hiding in Speech Signals for Secure Communication*. Online. 2014. Dostupné z:
<https://www.sciencedirect.com/book/9780128013281/information-hiding-in-speech-signals-for-secure-communication>. [cit. 9. 11. 2023].
- [19] LU, Z., GUO, S. *Lossless Information Hiding in Images*. Online. 2017. Dostupné z:
<https://www.sciencedirect.com/science/article/abs/pii/S09780128120064000012>. [cit. 9. 11. 2023].
- [20] GAN, Q., YAU, W., GAN, Y.S., SALAM, I., GUO, S., CHANG, CH., WU, Y., ZHOU, L. *DSteganoM: Deep steganography for motion capture data*. Online. 2023. Dostupné z:
<https://www.sciencedirect.com/science/article/pii/S0957417423024570#sec1>. [cit. 10. 11. 2023].
- [21] GeeksforGeeks.org. *Image Steganography in Cryptography*. Online. 2023. Dostupné z:
<https://www.geeksforgeeks.org/image-steganography-in-cryptography/>. [cit. 10. 11. 2023].

- [22] Scapy.net. *What is Scapy?* Online. Dostupné z:
<https://scapy.net/>. [cit. 14. 10. 2023].
- [23] Kali.org. *Hping3*. Online. Aktualizováno 2023. Dostupné z:
<https://www.kali.org/tools/hping3/>. [cit. 15. 10. 2023].
- [24] *Hping3: Full tutorial from noob to pro*. Online. 2021. Dostupné z:
<https://techyrick.com/hping3-full-tutorial-for-dummies-to-pro/#:~:text=Useful%20commands%20in%20hping3%20tool%20%F0%9F%91%87%201%20-c,mode%20%208%20%E2%80%93scan%3A%20SCAN%20mode.%20Da1%C5%A1%C3%AD%20polo%C5%BEky>. [cit. 15. 10. 2023].
- [25] MARTINGARCIA, L. *Nping Reference Guide*. Online. Dostupné z:
<https://nmap.org/book/nping-man.html#nping-man-echo-mode>. [cit. 15. 10. 2023].
- [26] NAGLE, D. *Packet Sender*. Online. Dostupné z:
<https://packetsender.com/documentation#gui>. [cit. 15. 10. 2023].
- [27] Ostinato.org. *Ostinato. Traffic Generator for Network Engineers*. Online. Dostupné z:
<https://ostinato.org/features>. [cit. 16. 10. 2023].
- [28] Ostinato.org. *Ostinato. Open-source*. Online. Dostupné z:
<https://ostinato.org/open-source#:~:text=Starting%20April%202024%2C%20Ostinato%20is%20no%20longer%20open-source.,older%20versions%20continue%20to%20be%20available%20on%20GitHub..> [cit. 10. 5. 2024].
- [29] ŠÍMA, J. *Nové scénáře s programem Wireshark v komunikačních technologiích*. Online. Brno: VUT v Brně, 2022. Dostupné z:
https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=239335. [cit. 17. 10. 2023].
- [30] RYBKA, Š. *Simulační scénáře v prostředí IT Guru pro předmět Komunikační technologie*. Online. Brno: VUT v Brně. 2013. Dostupné z:
https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=67684. [cit. 17. 10. 2023].
- [31] ŠÍMA, J. *Využití programu Wireshark v předmětu Komunikační technologie*. Online. Brno: VUT v Brně. 2020. Dostupné z:
https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=209702. [cit. 17. 10. 2023].

- [32] SWANN, M., ROSE, J., BENDIAB, G., SHIAELES, S., SAVAGE, N. *Tools for Network Traffic Generation - A Quantitative Comparison*. Online. 2020. Dostupné z:
<https://arxiv.org/ftp/arxiv/papers/2109/2109.02760.pdf>. [cit. 18. 10. 2023].
- [33] *Comparative study of various traffic generator tools*. Online. 2014. Dostupné z:
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6799557>. [cit. 18. 10. 2023].
- [34] MAZURCZYK, W., SZCZYPIORSKI, K. *Evaluation of steganographic methods for oversized IP packets*. Online. 2010. Dostupné z:
<https://link.springer.com/article/10.1007/s11235-010-9362-7>. [cit. 18. 10. 2023].
- [35] VMware.com. *VMware Workstation 17 Player*. Online. Dostupné z:
<https://www.vmware.com/products/workstation-player.html>. [cit. 4. 10. 2023].
- [36] Kali.org. *What is Kali Linux?* Online. Aktualizováno 2023. Dostupné z:
<https://www.kali.org/docs/introduction/what-is-kali-linux/>. [cit. 4. 10. 2023].
- [37] SHARPE, R., WARNICKE, E., LAMPING, U. *Wireshark User's Guide*. Online. Dostupné z:
https://www.wireshark.org/docs/wsug_html_chunked/. [cit. 5. 10. 2023].
- [38] OREBAUGH, A., RAMIREZ, G., BURKE, J., MORRIS, G., PESCE, L., WRIGHT, J. *Wireshark a Ethereal: Kompletní průvodce analýzou a diagnostik sítí*. Computer Press, a.s. 2008. Vydání první. ISBN 978-80-251-2048-4.
- [39] Internet Assigned Numbers Authority. *Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters*. Online. Aktualizováno 2023. Dostupné z:
<https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml>. [cit. 23. 11. 2023].
- [40] DEERING, S., HINDEN, R. *Internet Protocol, Version 6 (IPv6) Specification*. Online. 2017. Dostupné z:
<https://www.rfc-editor.org/rfc/rfc8200.html#section-4.1>. [cit. 13. 3. 2024].

Seznam symbolů a zkratek

ARP	Address Resolution Protocol
AS	autonomní systém
B	bajt
BGP	Border Gateway Protocol
BPC-KOM	Komunikační technologie
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DF	don't fragment
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
FCS	Frame Check Sequence
FHS	Filesystem Hierarchy Standard
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Security
IANA	Internet Assigned Numbers Authority
IARP	Inverse Address Resolution Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IP	Internet Protocol
LAN	Local Area Network
LLMNR	Link-Local Multicast Name Resolution

MAC	Media Access Control
mDNS	Multicast Domain Name System
MF	more fragments
MLD	Multicast Listener Discovery
MPC-PKT	Pokročilé komunikační techniky
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
NA	Neighbor Advertisement
ND	Neighbor Discovery
NS	Neighbor Solicitation
OSPF	Open Shortest Path First
PadN	padding
PDU	Protocol Data Unit
PIM	Protocol Independent Multicast
QoS	Quality of Service
RA	Router Advertisement
RS	Router Solicitation
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TTL	Time to Live
UDP	User Datagram Protocol
USB	Universal Serial Bus

Seznam příloh

A	Datové provozky a vytvořené soubory	85
A.1	<i>bgp.ostm / bgp.pcapng</i>	85
A.2	<i>dhcp.ostm / dhcp.pcapng</i>	85
A.3	<i>dns.ostm / dns.pcapng</i>	85
A.4	<i>ftp.ostm / ftp.pcapng</i>	86
A.5	<i>icmpv6.ostm / icmpv6.pcapng</i>	86
A.6	<i>mdns.ostm / mdns.pcapng</i>	86
A.7	<i>mld.ostm / mld.pcapng</i>	86
A.8	<i>quic.ostm / quic.pcapng</i>	87
A.9	<i>tcp.ostm / tcp.pcapng</i>	87
A.10	<i>udp.ostm / udp.pcapng</i>	87
A.11	Popis souborů: <i>popis.docx</i>	87
B	Limitace velikostí variant pro skrytí dat	88
B.1	<i>auth_header.ostm / auth_header.pcapng</i>	88
B.2	<i>bgp_optional.ostm / bgp_optional.pcapng</i>	88
B.3	<i>dhcp_option.ostm / dhcp_option.pcapng</i>	88
B.4	<i>dns_add_records.ostm / dns_add_records.pcapng</i>	88
B.5	<i>dns_txt.ostm / dns_txt.pcapng</i>	88
B.6	<i>dns_rrsig.ostm / dns_rrsig.pcapng</i>	89
B.7	<i>dns_signature.ostm / dns_signature.pcapng</i>	89
B.8	<i>hop_by_hop_opt.ostm / hop_by_hop_opt.pcapng</i>	89
B.9	<i>mdns.ostm / mdns.pcapng</i>	89
B.10	<i>mld.ostm / mld.pcapng</i>	89
B.11	<i>quic_ids.ostm / quic_ids.pcapng</i>	89
B.12	<i>routing_header.ostm / routing_header.pcapng</i>	89
B.13	<i>tcp_option.ostm / tcp_option.pcapng</i>	90
C	Kombinace variant pro skrytí dat	91
C.1	<i>kombinace_ipv6.ostm / kombinace_ipv6.pcapng</i>	91
C.2	<i>komb_vrstvy.ostm / komb_vrstvy.pcapng</i>	91
C.3	Popis souborů: <i>popis_kombinace.docx</i>	91
D	Další verze projektu pro MPC-PKT	92
D.1	<i>dns.ostm / dns.pcapng</i>	92
D.2	<i>icmpv6_fragmenty.ostm / icmpv6_fragmenty.pcapng</i>	92
D.3	<i>mdns.ostm / mdns.pcapng</i>	92

D.4	<i>mld_v1.ostm / mld_v1.pcapng</i>	92
D.5	<i>mld_v2.ostm / mld_v2.pcapng</i>	93
D.6	<i>projekt_pkt.ostm / projekt_pkt.pcapng</i>	93
D.7	Popis souborů: <i>popis_projekt.docx</i>	93
E	Testování paketů na reálné LAN síti	94
E.1	<i>test_dns.pcapng</i>	94
E.2	<i>test_dns_ar.pcapng</i>	94
E.3	<i>test_echo_reply.pcapng</i>	94
E.4	<i>test_llmnr.pcapng</i>	94
E.5	<i>test_mdns.pcapng</i>	94
E.6	<i>test_ping.pcapng</i>	95
E.7	<i>test_udp.pcapng</i>	95

A Datové provozy a vytvořené soubory

V této části příloh jsou krátce popsány jednotlivé datové toky, které byly generovány za účelem skrytí dat. Ke každé variantě je poskytnut vždy soubor *.ostm* a *.pcapng*. Soubory *.ostm* prezentují datové toky tvořené v nástroji Ostinato, ve kterém je též možné tyto datové toky upravovat a konkrétní bajty paketů modifikovat dle preferencí a požadavků. Soubory *.pcapng* znázorňují zachycení generovaných datových toků v programu Wireshark. Dále je též poskytnut textový soubor přehledně popisující jednotlivé varianty a jejich limitace.

A.1 *bgp.ostm* / *bgp.pcapng*

Datové toky simulují komunikaci mezi dvěma zařízeními s IP adresami 192.168.100.1 a 192.168.100.2. Nejprve dochází k navázání TCP spojení pomocí 3-way handshake, následně si zařízení vymění *BGP Open* zprávy a poté zprávy *Keepalive*.

Tajná informace je skrytá v první *BGP Open* zprávě. Pro skrytí informace je využit protokol aplikační vrstvy BGP, konkrétně pole *Optional Parameters: Authentication*.

A.2 *dhcp.ostm* / *dhcp.pcapng*

Datové toky simulují komunikaci, ve které čtyři zařízení žádají DHCP server o přidělení IPv4 adresy. Souběžně s klasickou DHCP komunikací probíhá i skrytá komunikace, ve které dochází k výpočtu rovnice.

Tajná informace je skrytá v každé zprávě. Pro skrytí informace je využit protokol aplikační vrstvy DHCP, konkrétně pole *Option: (215) Unassigned*.

A.3 *dns.ostm* / *dns.pcapng*

Datové toky simulují překlady doménových jmen na různé typy záznamů včetně A, AAAA, TXT a dalších.

V této variantě se nacházejí čtyři možnosti pro skrytí informace. Ke skrytí informace je vždy využit protokol aplikační vrstvy DNS.

První možnost skrytí informace je ukázána v druhé *DNS Response* zprávě, v sekci Answers – *type TXT*.

Druhá možnost skrytí informace je ukázána ve třetí *DNS Response* zprávě, v sekci Answers – *Additional records – type TXT*.

Třetí možnost skrytí informace je ukázána v páté *DNS Response* zprávě, v sekci *Answers – type RRSIG*, v poli *Signature*.

Čtvrtá možnost skrytí informace je ukázána v šesté *DNS Response* zprávě, v sekci *Answers – type RRSIG*, v polích *Original TTL*, *Signature Expiration*, *Signature Inception*, *Key Tag*.

A.4 *ftp.ostm / ftp.pcapng*

Datové toky simulují připojení zařízení s IPv6 adresou fe80::20c:29ff:fe69:e12a na FTP server.

Tajná informace je skrytá ve zprávě *FTP Request: USER* sloužící k autentizaci uživatele. Pro skrytí informace je využit protokol internetové vrstvy IPv6, konkrétně rozšířené IPv6 záhlaví *Authentication Header* a jeho pole *AH SPI*, *AH Sequence*, *AH ICV*.

A.5 *icmpv6.ostm / icmpv6.pcapng*

Datové toky simulují výměnu Echo request a reply zpráv mezi dvěma zařízeními. Jednotlivé zprávy jsou fragmentovány na tři části. Přenos je ukončen zprávou *ICMP Destination Unreachable*.

Tajná informace je skrytá právě ve zprávě *ICMP Destination Unreachable*. Pro skrytí informace je využit protokol internetové vrstvy IPv6, konkrétně několik polí typu *Address* rozšířeného IPv6 záhlaví *Routing Header*.

První tři fragmenty také obsahují informaci skrytou v rámci rozšířeného IPv6 záhlaví *Fragment Header*. Do něho lze vložit pouze 4 B dat, a proto není velmi vhodné.

A.6 *mdns.ostm / mdns.pcapng*

Datové toky simulují překlady lokálních doménových jmen pomocí protokolů LLMNR a mDNS.

Tajná informace je skrytá ve druhé zprávě *mDNS Response*. Pro skrytí informace je využit protokol aplikační vrstvy mDNS, konkrétně záznam *AAAA* v sekci *Answers*.

A.7 *mld.ostm / mld.pcapng*

Datové toky simulují žádosti stanic o připojení do multicastové skupiny ff02::16.

Tajná informace je skrytá v první *MLDv2 Report* zprávě. Pro skrytí informace je využit protokol ICMPv6, konkrétně pole *Multicast Address Record Exclude*.

A.8 *quic.ostm / quic.pcapng*

Datové toky simulují komunikaci mezi klientem a serverem pomocí protokolu QUIC.

Tajná informace je skrytá v první *Handshake* zprávě. Pro skrytí informace je využit protokol aplikační vrstvy QUIC, konkrétně pole *Destination Connection ID* a *Source Connection ID*.

A.9 *tcp.ostm / tcp.pcapng*

Datové toky simulují komunikaci mezi zařízením s IP adresou 10.1.2.1 a veřejným DNS Cloudflare serverem 1.1.1.1. Klient žádá o překlad doménového jména *komppkt.utko.fekt.vut.cz*.

Tajná informace je skrytá v prvních třech TCP zprávách, které simulují navázání TCP spojení. Pro skrytí informace je využit protokol transportní vrstvy TCP, konkrétně pole *TCP Options - Experimental: Unknown*.

A.10 *udp.ostm / udp.pcapng*

Datové toky simulují výměnu dat mezi dvěma zařízeními. Část tajné informace je skrytá v každé zprávě zasílané zařízením s IP adresou 2001:db8:85a3::8a2e:370:7334. Konečná informace vzniká poskládáním dílčích částí.

Informace je v každé zprávě skrytá na dvou místech. Pro skrytí první části informace je využit protokol internetové vrstvy IPv6, konkrétně rozšířené záhlaví *IPv6 Hop-by-Hop Option* a v něm pole *PadN*. Druhá část informace je skrytá v datové části transportního protokolu UDP.

A.11 **Popis souborů:** *popis.docx*

Tento soubor poskytuje přehledný popis jednotlivých datových toků včetně specifikace limitací velikostí vkládaných dat.

B Limitace velikostí variant pro skrytí dat

Přílohami jsou opět soubory *.ostm* a *.pcapng*, které znázorňují maximální velikost dat, které lze vložit do polí datových toků použitých v rámci této diplomové práce pro skrývání dat.

Soubory představují limitace konkrétních polí a také celého paketu. Pro některé varianty totiž platí, že polí daného typu může být vloženo více za sebou a tím pádem může být skryto větší množství dat.

B.1 *auth_header.ostm / auth_header.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *IPv6 - Authentication Header*. Maximum činí 1 026 B.

B.2 *bgp_optional.ostm / bgp_optional.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *BGP - Optional Parameter: Authentication*. Maximum činí 255 B.

B.3 *dhcp_option.ostm / dhcp_option.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *DHCP - Option: Unassigned*. Maximum činí 255 B.

B.4 *dns_add_records.ostm / dns_add_records.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *DNS - Additional records: TXT*. Maximum činí 255 B.

B.5 *dns_txt.ostm / dns_txt.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *DNS - Answers: TXT*. Maximum činí 255 B.

B.6 *dns_rrsig.ostm / dns_rrsig.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do polí *DNS - Answers - RRSIG: Original TTL, Signature Expiration, Signature Inception, Key Tag*. Maximum činí celkem 14 B.

B.7 *dns_signature.ostm / dns_signature.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *DNS - Answers - RRSIG: Signature*. Maximum je omezeno velikostí MTU, která je v tomto případě 1500 B a spolu s dalšími položkami paketu omezuje velikost podpisu na 1 344 B.

B.8 *hop_by_hop_opt.ostm / hop_by_hop_opt.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *IPv6 - Hop-by-Hop-Option: PadN*. Maximum činí 255 B.

B.9 *mdns.ostm / mdns.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *mDNS - Answers: AAAA Address*. Maximum činí 16 B.

B.10 *mld.ostm / mld.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *ICMPv6 - Multicast Address Record Exclude*. Maximum činí 16 B.

B.11 *quic_ids.ostm / quic_ids.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *QUIC - Destination + Source Connection ID*. Do každého ID lze vložit 255 B. Maximum činí 510 B.

B.12 *routing_header.ostm / routing_header.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *IPv6 - Routing Header: Address*. Maximum činí 16 B.

B.13 *tcp_option.ostm / tcp_option.pcapng*

Soubory prezentují maximální velikost dat, které lze vložit do pole *TCP Option* - *Experimental: Unknown*. Maximum je závislé na dalších položkách TCP Options, v tomto případě činí 37 B.

C Kombinace variant pro skrytí dat

Tyto přílohy prezentují soubory, které znázorňují, jak je možné zkombinovat skrytí informace v rámci jednoho paketu na více místech zároveň. I zde je ke každé variantě poskytnut soubor *.ostm* a *.pcapng* a jejich přehledný popis.

C.1 *kombinace_ipv6.ostm* / *kombinace_ipv6.pcapng*

Na zprávách *Echo request* a *Echo reply* je ukázáno, že v jednom paketu za sebe lze vložit až pět rozšiřujících IPv6 záhlaví. Konkrétně se jedná o typy: 0 - *IPv6 Hop-by-Hop Option*, 60 - *Destination Options*, 43 - *Routing Header*, 44 - *Fragment Header*, 51 - *Authentication Header*.

Pro skrytí informace jsou využita pole *PadN*, *Address*, či *AH ICV*.

C.2 *komb_vrstvy.ostm* / *komb_vrstvy.pcapng*

Tato varianta simuluje DNS dotaz a odpověď na doménu *kom-ppkt.utko.fekt.vut.cz*.

V rámci DNS odpovědi je informace skrytá na čtyřech místech paketu, a to na třech různých vrstvách modelu TCP/IP. Na vrstvě síťové je informace skrytá v *IPv6 Authentication Header* polích *AH Sequence* a *AH ICV*, v rámci transportní vrstvy v *TCP Options - Experimental: Unknown* a na vrstvě aplikační na dvou místech v rámci protokolu *DNS*, konkrétně v poli *RRSIG Signature* a *Additional records: type TXT*.

C.3 Popis souborů: *popis_kombinace.docx*

Tento soubor poskytuje přehledný popis výše zmíněných paketů.

D Další verze projektu pro MPC-PKT

Přílohy souvisí s návrhem zadání projektu pro předmět MPC-PKT. Přílohy obsahují dílčí části zadání projektu stejně jako jeho celou verzi, ve které jsou tyto čtyři dílčí sestavy datových toků poskládány za sebe. Jednotlivé datové toky představují různé komunikační protokoly, na jejichž výuku je předmět MPC-PKT zaměřen. V každém z těchto toků je skrytá tajná informace, kterou studenti mají odhalit.

D.1 *dns.ostm / dns.pcapng*

Datové toky simulují dva DNS dotazy a jim příslušející odpovědi.

Skrytou informací jsou souřadnice **Petronas Twin Towers**, které jsou vloženy v první DNS odpovědi v poli *RRSIG Signature*.

D.2 *icmpv6_fragmenty.ostm / icmpv6_fragmenty.pcapng*

Datové toky simulují výměnu zpráv Echo request a Echo reply mezi dvěma zařízeními. Pakety jsou fragmentované.

Skrytou informací je odkaz na vyplnění formuláře. Odkaz je rozdělen na pět částí a každá část je skrytá vždy v datové části každého druhého fragmentu dané zprávy.

D.3 *mdns.ostm / mdns.pcapng*

Datové toky simulují dva mDNS dotazy a jim příslušející odpovědi.

Skrytou informací je text dotazující se na událost ze dne **1.1.1993** a je vložen v první mDNS odpovědi, v prvním záznamu, v poli *AAAA Address*.

D.4 *mld_v1.ostm / mld_v1.pcapng*

Datové toky simulují komunikaci několika zařízení a zahrnují zprávy *RS*, *RA*, *NS*, *NA*, *MLD Query* a *MLD Report*.

Skrytou informací je text **write in google search: meteorite**, který vybízí studenta k provedení této akce, jež způsobí přelétnutí meteoritu přes okno prohlížeče. Text je vložen v první MLD Query, v části *IPv6 Hop-by-Hop Option*, poli *PadN*.

D.5 *mld_v2.ostm / mld_v2.pcapng*

Tato varianta je obdobou varianty předchozí. Jediným rozdílem je místo vložení informace. Po vzoru starších zadání je text vložený jako Ethernet trailer.

D.6 *projekt_pkt.ostm / projekt_pkt.pcapng*

Tyto soubory prezentují zadání semestrálního projektu a jsou spojením všech dílčích částí představených výše.

D.7 Popis souborů: *popis_projekt.docx*

Tento soubor poskytuje přehledný popis jednotlivých datových toků.

E Testování paketů na reálné LAN síti

Přílohy prezentují nestandardní pakety zachycené nástrojem Wireshark na reálné lokální Wi-Fi síti mezi dvěma zařízeními.

E.1 *test__dns.pcapng*

Soubor zachycuje paket *DNS Standard query* generovaný v nástroji Ostinato a odpověď *DNS Standard query response* zaslanou operačním systémem. V *DNS Standard query* je informace skrytá v *IPv6 Fragment Header*.

E.2 *test__dns__ar.pcapng*

Soubor zachycuje paket *DNS Standard query* generovaný v nástroji Ostinato a odpověď *DNS Standard query response* zaslanou operačním systémem. V *DNS Standard query* je informace skrytá v sekci *Additional Records - TXT*.

E.3 *test__echo__reply.pcapng*

Soubor zachycuje paket *Echo reply* generovaný v nástroji Ostinato na zařízení Laptop 1 a zasílaný na zařízení Laptop 2. Informace je skrytá v *IPv6 Hop-by-Hop Option* a datové části ICMPv6.

E.4 *test__llmnr.pcapng*

Soubor zachycuje paket *LLMNR Standard query* generovaný v nástroji Ostinato a odpověď *LLMNR Standard query response* zaslanou operačním systémem. V *LLMNR Standard query* je informace skrytá v *IPv6 Routing Header*.

E.5 *test__mdns.pcapng*

Soubor zachycuje paket *mDNS Standard query* generovaný v nástroji Ostinato a odpověď *mDNS Standard query response* zaslanou operačním systémem. V *mDNS Standard query* je informace skrytá v *IPv6 Destination Options*.

E.6 *test_ping.pcapng*

Soubor zachycuje paket *Echo request* generovaný v nástroji Ostinato a odpověď *Echo reply* zaslanou operačním systémem. V *Echo request* je informace skrytá v *IPv6 Hop-by-Hop Option*.

E.7 *test_udp.pcapng*

Soubor zachycuje paket *UDP* generovaný v nástroji Ostinato na zařízení Laptop 1 a zasílaný na zařízení Laptop 2. Informace je skrytá v *IPv6 Authentication Header* a datové části UDP.