VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GRAFICKÝ NÁSTROJ PRO GENEROVÁNÍ IPV6 PAKETŮ

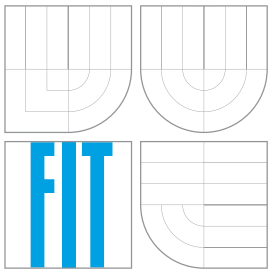
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB JOCHEC

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GRAFICKÝ NÁSTROJ PRO GENEROVÁNÍ IPV6 PAKETŮ

GRAPHICAL TOOL FOR IPV6 PACKET GENERATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAKUB JOCHEC
AUTHOR

VEDOUCÍ PRÁCE — Ing. VIKTOR PUŠ
SUPERVISOR

BRNO 2012

Abstrakt

Práce se zabývá vytvořením grafického rozhraní pro generování IPv6 paketů, pomocí kterého bude možné vytvářet korektní pakety s různými nastaveními. Je zde popsána architektura TCP/IP a následně IPv6 včetně jeho rozšiřujících hlaviček. Popsáno je také několik z dostupných nástrojů pro generování IPv6 paketů. Další část je věnována návrhu aplikace a její implementace pomocí jazyka Python a knihoven wxPython a Scapy.

Abstract

This thesis is targeted on creation of graphic interface for generating IPv6 packets which can be used for creation of correct packets with different options. It describes TCP/IP model of network and IPv6 including extension headers. In next part is presented some of existing tools for IPv6 packet generating. Last part includes user interface design and implementation using Python and wxPython and Scapy libraries.

Klíčová slova

TCP/IP, IPv6, grafické rozhraní, Python, wxPython, Scapy

Keywords

TCP/IP, IPv6, graphic interface, Python, wxPython, Scapy

Citace

Jakub Jochec: Grafický nástroj pro generování IPv6 paketů, bakalářská práce, Brno, FIT VUT v Brně, 2012

Grafický nástroj pro generování IPv6 paketů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Viktora Puše

.....
Jakub Jochec
15. května 2012

Poděkování

Děkuji za odborné vedení a věcné připomínky vedoucímu své bakalářské práce Ing. Viktoru Pušovi.

Obsah

Seznam obrázků

Kapitola 1

Úvod

Internet protocol verze 6 (IPv6) postupně nahrazuje současný dominantní protokol Internetu, kterým je Internet protocol verze 4 (IPv4). Internet přenáší data mezi cílovými body pomocí paketů, které jsou směrovány přez síť pomocí routovacích protokolů. Každý uzel musí být schopný tyto pakety přečíst a podle informací v nich se rozhodnout, kam daný paket pošle.

Tím vzniká potřeba pro nástroje, které umožňují generování paketů obsahující IPv6 datagram. Takovéto programy již existují, ale mají několik zásadních problémů. Mezi ně patří například ukončený vývoj, kde daná aplikace sice zvládá paket vytvořit, ale podporuje pouze malou část ze specifikací nebo je implementace zastaralá a neodpovídá aktualizovaným normám.

Dalším problémem je to, že naprostá většina aplikací má pouze ovládání z příkazové řádky. I když toto není nikterak zásadní nevýhoda, je potřeba vytvořit pro takovéto programy grafické rozhraní. Jestliže je grafické prostředí navrženo kvalitně, může uživateli ušetřit čas. Není totiž nutné studovat jak se daný CLI (command line nástroj) ovládá.

Cílem této práce je vytvořit takový grafický nástroj, který bude schopný vytvořit paket obsahující IPv6 datagram. Takovýto datagram by měl obsahovat většinu z rozšiřujících hlaviček a vybrané ICMPv6 zprávy. Aplikace také musí umět uložit vytvořený paket do souboru pcap.

Než je možné přistoupit k vývoji aplikace, je nutné seznámit se s modelem TCP/IP (kapitola 2, na kterém je fungování Internetu postaveno. V kapitole 3 je popsán vývoj a vlastnosti IPv6. Jedná se zejména o formát datagramu, adres a rozšiřující hlavičky. Jak bylo výše uvedeno, již existují nástroje pro vytváření paketů. Některé z nich jsou popsané v kapitole 4. Samotným návrhem a implementací grafického prostředí včetně popisu použitách nástrojů je věnována kapitola 5. Tato kapitola také zobrazuje výslednou podobu vytvořené aplikace a příklady jejího použití. Kapitola 6 shrnuje výslednou aplikaci. Ukazuje příklad vytvořeného paketu pomocí nástroje Wireshark [15]. Dále porovnává vytvořené prostředí s již existující aplikací PackETH [**?**]. Jsou zde také shrnuty možnosti jejího rozšíření.

Kapitola 2

TCP/IP

Transmission Control Protocol/Internet Protocol neboli TCP/IP je označení pro síťovou architekturu. Specifikuje různé vrstvy v komunikaci mezi zařízeními a jak jsou tyto vrstvy spolu svázané.

2.1 Historie

TPC/IP se poprvé objevilo v 60. letech minulého století. Jednalo se o nové protokoly vyvinuté pro síť ARPANET agentury ARPA. Jednotlivé protokoly se vyvíjely na univerzitách v USA a byly financovány z dotací amerického ministerstva obrany. Dnešní podobu tyto protokoly získaly v druhé polovině 70. let. Postupným rozšiřováním ARPANETu o další sítě se z něj stal Internet.

Agentura ARPA se snažila nově vyvinuté protokoly prosadit i mimo svojí vlastní síť. Jednalo se hlavně o univerzitní sítě. Většina pracovišť provozovala BSD Unix pocházející z University of California v Berkley. DARPA (přejmenovaná ARPA) si proto nechala na zakázku implementovat TCP/IP protokoly do operačního systému BSD Unix. Díky tomuto kroku se tak staly standardní součástí různých Unixů. Přispělo se tak k jejich masovému rozšíření a implementaci i v ostatních operačních systémech.

Rodině protokolů TCP/IP se také jinak říká Internet Protocol Suite, což naznačuje její přímé spojení s dnešním Internetem.

2.2 Vrstvy TCP/IP

Na rozdíl od ISO/OSI, které definuje sedm vrstev komunikace, využívá TCP/IP pouze čtyři.

- Aplikační vrstva (Application layer)
- Transportní vrstva (Transportation layer)
- Síťová vrstva (Network layer)
- Vrstva síťového rozhraní (Network intarface layer)

2.2.1 Vrstva síťového rozhraní

Nejnižší vrstvou TCP/IP modelu je vrstva síťového rozhraní často označována též jako linková vrstva. Má na starost zpracování a vysílání/příjímání paketů po fyzickém spoji mezi dvěma koncovými zařízeními. Nejčastěji se jedná o zpracování Ethernetové hlavičky k paketu, která obsahuje adresy síťových prvků komunikujících zařízení (MAC adresa). To ale neznamená, že by byla komunikace omezena pouze na sítě typu Ethernet. V různých typech sítí se může připojovaná hlavička lišit. Jedná se například o Token ring, X.25, Frame relay apod.

2.2.2 Síťová vrstva

Síťová vrstva (označována také jako Internetová či IP vrstva) je realizována pomocí IP protokolu. Stará se o doručení paketů od odesílatele k adresátovi. Tato vrstva zaobaluje informace protokolů vyšší vrstvy, které jsou jednoznačně identifikovány svým číslem. Jedná se například o ICMP, TCP nebo UDP. Komunikující stanice jsou identifikovány IP adresou. V Případě IP verze 4 se jedná o čtveřici 8-bitových čísel. Při použití IP verze 6 se jedná o osm skupin 16-bitových skupin.

2.2.3 Transportní vrstva

Transportní vrstva bývá také označována jako TCP vrstva. Jejím hlavním úkolem je zajistit přenos dat mezi dvěma komunikujícími účastníky. V případě TCP/IP se jedná přímo o koncové aplikace (prohlížeč - webový server). Podle požadavků komunikujících aplikací je možné regulovat tok dat oběma směry, zajišťovat spolehlivost přenosu, či provozovat spojovaný (TCP) nebo nespojovaný (UDP) přenos. Na této vrstvě se se používají čísla portů, což je 16-bitové číslo. Každá aplikace má přiřazený port, pomocí kterého je jednoznačně určena. Komunikující stanice tak ví, kterému programu přijatá data předat.

Transportní vrstva nejčastěji obsahuje TCP (Transmission Control Protocol) protokol (proto také TCP vrstva), není to však jediná možnost. Pokud není komunikující aplikací vyžadována spolehlivost přenosu, používá se protokol UDP (User Datagram Protocol). Další možností je protokol ICMP (Internet Control Message Protocol), který v síti slouží k oznamování chyb, předávání diagnostických informaci, nebo pro účely směrování.

2.2.4 Aplikační vrstva

Jedná se o nejvyšší vrstvu modelu. Tuto vrstvu využívají jednotlivé aplikace k výměně vlastních dat. Tyto data mohou být například protokoly FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol) apod. Data se zapouzdří do jednoho z protokolů transportní vrstvy (jako je například TCP nebo UDP).

Oproti ISO/OSI aplikace komunikují přímo s transportní vrstvou. Proto si musí veškeré prezentační a relační služby zajišťovat samy.

Protokoly aplikační vrstvy jsou většinou založeny na struktuře server - klient. Proto mají obvyklé servery přiděleny dané porty, jako je například HTTP 80, Telnet 23, SSH 22, FTP 20/21 atd.

Kapitola 3

IPv6

Internet Protocol verze 6 (IPv6) je následovníkem nejrozšířenějšího protokolu internetu IPv4.

3.1 Motivace a vývoj

Hlavním důvodem k vytvoření nového standardu v adresování zařízení v internetu byl nedostatek použitelných adres IPv4. Počátkem 90. let už bylo jasné, že dostupné adresy dojdou a nová zařízení nebudou moci dostat veřejnou IP adresu. Tehdejší studie ukazovaly, že je čas na řešení tohoto problému přibližně 10 let. IETF (The Internet Engineering Task Force) tedy přistoupilo k vytvoření pracovních skupin, které začaly pracovat na vývoji IPv6, tehdy označovaného za IPng (IP next genaration).

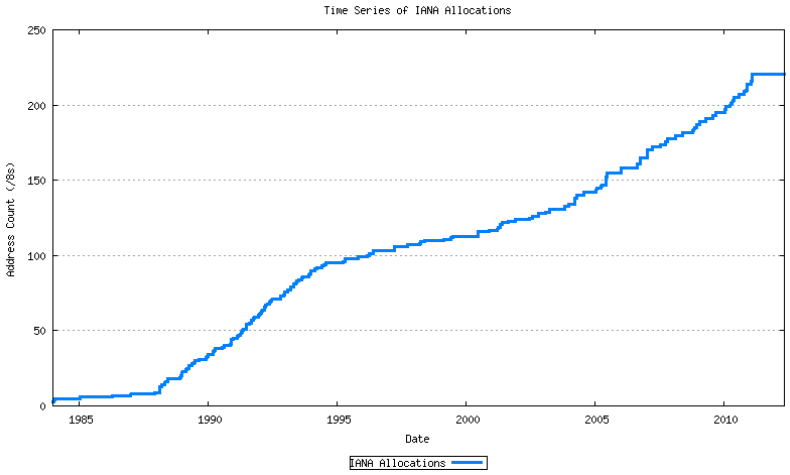
Vzhledem k tomu, že času na vytvoření nového systému bylo relativně dost, IETF se rozhodlo nejen vyřešit nedostatek IP adres, ale i přidat další vlastnosti. Hlavní požadavky byly

- rozsáhlý adresní prostor
- tři druhy adres (unicast, multicast, anycast)
- jednotné adresní schéma pro Internet a vnitřní sítě
- hierarchické směrování v souladu s hierarchickou adresací
- zvýšení bezpečnosti (zavedení šifrování, autentizace a sledovaní trasy paketu)
- podpora služeb se zajištěním kvality
- optimalizace pro vysokorychlostní směrování
- automatická konfigurace
- podpora mobility
- plynulý přechod z IPv4

Vývoje se ujali hlavně Steven Deering a Robert Hinden, kteří v roce 1995 vydali sadu RFC dokumentu definujících základní prvky IPv6. Hlavním bylo RFC 1883: *Internet Protocol, Version 6 (IPv6) Specification* [2].

V tomto RFC byl mimo jiné vyřešen problém, kvůli kterému se s celým vývojem začalo. Délka adresy byla z 32 bitů IPv4 adresy zvětšena na 128 bitů. Tím se vytvořil dostatečně velký prostor, ve kterém je obsaženo 2^{128} adres.

IPv6 tak bylo připraveno k implementaci, ale tu zbrzdilo zavedení beztřídního adresování CIDR (Classless Inter-Domain Routing) a používání NAT (Network address translation), čímž se oddálil problém nedostatku IP adres.



Obrázek 3.1: Přidělené IPv4 bloky (zdroj:ipv4.potaroo.net)

Jak je na obrázku 3.1 vidět, došlo sice ke zpomalení úbytku volných adresových bloků, nicméně kolem roku 2005 se čerpání adresních bloků opět zrychlilo.

Momentální stav je takový, že centrální zásoba bloků IANA byla 3. února 2011 vyčerpána a jednotlivým RIR postupně dochází zásoby. V tabulce 3.1 je vidět aktuální stav registrů a predikce, kdy dojde k vyčerpání zásob jednotlivých registrátorů.

RIR	Předpokládané datum vyčerpání	Zbývající adresy (/8)
APNIC:	19-Apr-2011	1.1647
RIPENCC:	14-Aug-2012	2.2368
ARIN:	20-Jun-2013	4.9043
LACNIC:	29-Jan-2014	3.6393
AFRINIC:	05-Nov-2014	4.3216

Tabulka 3.1: Předpokládané vyčerpání bloků registrátorů (zdroj:ipv4.potaroo.net)

V roce 1998 vyšla revidovaná sada RFC dokumentů (RFC 2460) s definicí základních protokolů a služeb a postupně jsou aktualizovány či doplňovány. V roce 2006 se vytvořila poslední verze adresní architektury, podpora mobility byla zavedena v roce 2004 a revidována v roce 2011. [8]

3.2 Adresy

3.2.1 Typy adres

V RFC 4291 [4] je definován formát adres, jejich typy a další potřebné náležitosti. Každému ze síťových zařízení je přiděleno několik typů adres. Existují 3 typy adres s rozdílným chováním.

Individuální (unicast) jedná se o standardní adresu, která identifikuje právě jedno síťové rozhraní, kterému mají být doručena data.

Skupinové (multicast) slouží pro adresování skupin počítačů. Data odeslaná na tuto adresu musí být doručena všem zařízením.

Výběrové (anycast) označují skupinu zařízení. Data poslaná na tuto adresu se na rozdíl od multicastu doručí pouze jednomu zařízení a to tomu nejbližšímu.

Oproti IPv4 lze vidět, že zmizely všesměrové (broadcast) adresy. Jejich funkci převzaly speciální skupiny, které umožňují distribuci zpráv pro všechny na dané lince.

IPv6 přikazuje aby každé rozhraní mělo několik adres na kterých musí být dostupné, ale nijak neomezuje jejich celkový počet. Pro koncový prvek se jedná o následující typy:

- lokální linková adresa
- všechny individuální a výběrové adresy
- lokální smyčka
- skupinové adresy pro všechny uzly
- skupinová adresa pro vyzývaný uzel pro všechny přidělené individuální a výběrové adresy
- všechny skupinové adresy, jejichž je členem

3.2.2 Vzhled a zápis IPv6 adresy

IPv6 adresa má 128bitů. Standardně se zapisuje jako osm skupin po čtyřech hexadecimálních číslicích. Každá skupina vyjadřuje 16 bitů adresy. Skupiny se od sebe oddělují dvojtečkou. Příkladem takové adresy je

2001:67c:1220:c1c1:7185:ac13:9a0b:b3b4

Pokud adresa obsahuje nulové skupiny, můžou se zkrátit z *0000* na *0*. Počáteční nuly se z jednotlivých bloků mohou vynechat. Jestliže se objeví několik nulových bloků za sebou, lze je nahradit pomocí *::*. Z toho vyplývá, že adresu

0abc:0000:0000:0000:0def:ac13:9a0b:b3b4

lze zapsat jako

abc:0:0:0:0def:ac13:9a0b:b3b4

nebo jako

0abc::0def:ac13:9a0b:b3b4

Extrémním případem je nedefinovaná adresa

0000:0000:0000:0000:0000:0000:0000:0000

kterou lze zkrátit na

::

Zkrácení nulových bloků na *::* lze ovšem v adrese použít pouze jednou. To znamená, že adresa

0abc:0000:0000:0000:0def:0000:0000:0000

lze zkrátit na jednu z následujících možností

abc::0def:0000:0000:0000 nebo 0abc:0000:0000:0000:0def::

Tolik možností zápisu jedné adresy vedlo k vydání RFC 5952: *A Recommendation for IPv6 Address Text Representation* [5], které upravuje jak se adresy zobrazují uživateli. Výpis adres by se měl řídit následujícími pravidly:

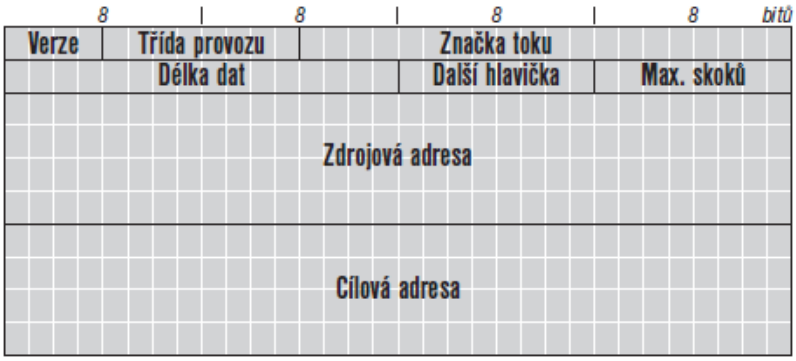
- hexadecimální číslice se píší vždy malými písmeny
- vynechávání počátečních nul je povinné
- *::* musí nahradit nejdelší možný řetězec nul a musí je pohltit všechny. Pokud se vyskytne více nulových bloků stejné délky (např. tři a tři), musí se použít místo prvních bloků.
- pokud s nulovým blokem sousedí nenulové bloky, nesmí se zkrátit na *::*

3.2.3 Prefixy

Prefixy slouží ke stejnému účelu jako v IPv4. Určují příslušnost k dané síti. Všechny přiřazené adresy v dané síti mají stejný prefix. Typ zápisu je

IPv6 adresa/délka prefixu

Délka prefixu označuje, kolik bitů na začátku adresy mají všechny adresy v síti stejné. Pokud bude délka prefixu 64, znamená to, že všechny adresy podsítě mají prvních 64 bitů adresy společné.



Obrázek 3.2: Základní IPv6 hlavička [8]

3.3 Formát datagramu

Formáty datagramu jsou specifikovány v RFC 2460: Internet Protocol, Version 6 (IPv6) Specification [3], které zastaralo RFC 1883. Datagram má standardní formát, kde je na začátku hlavička, kterou následují přenášená data. Oproti IPv4 však došlo k zásadním změnám ve formátu hlavičky. Dříve byla hlavička proměnlivé délky. Jednotlivá zařízení po cestě paketu mohla měnit její velikost přidáváním nebo odebíráním položek, což mělo za následek zatěžování zařízením neustálým přepočítáváním kontrolních součtů.

IPv6 naopak zavedlo pevnou velikost hlavičky a veškerá rozšíření přenesla do rozšiřujících hlaviček, které mohou, ale nemusí být v datagramu obsaženy. Tyto rozšiřující hlavičky jsou popsány v další části práce 3.4.

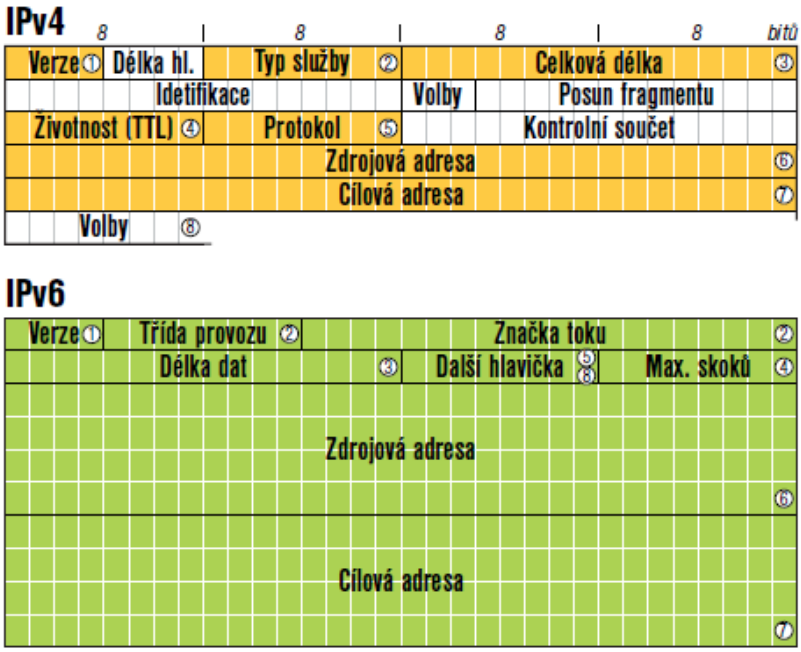
Na obrázku 3.2 je zobrazena IPv6 hlavička datagramu ve standardní podobě. Její velikost je 40B (oproti 20B IPv4 hlavičky).

Verze (Version) Tato položka identifikuje typ IP datagramu a je neměnná. Vždy musí byt 6.

Třída provozu (Traffic Class) Třída provozu je osmibitová položka obsahující prioritu datagramu nebo jeho zařazení do dané přepravní třídy. Vytváří se tím prostor pro poskytování služeb se zaručenou kvalitou. Implicitní hodnota tohoto pole je 0. Samostatný IP protokol nedokáže nijak zaručit přepravní parametry. Poskytuje však tzv. diferencované služby, jejichž pomocí je možné pakety přednostně odbavovat na zařízeních.

Značka toku (Flow label) Jedná se o 20-bitové pole. Stejně jako třída provozu ještě stále není přesně definováno, jak toto pole využívat. Zamýšleným využitím je identifikace toku dat mezi zařízeními (identifikátor paketů se společnými parametry). Pokud stanice nebo směrovače tuto položku nepodporují jsou povinny ji nastavit na 0 při vytváření paketu, při směrování hodnotu neměnit a ignorovat při přijmutí.

Délka dat (Payload length) 16-bitová položka obsahující délku datagramu. Do délky se počítají veškeré informace následující základní IPv6 hlavičku (včetně rozšiřujících hlaviček). Délka obsahu se udává v oktetech, z čehož vyplývá maximální délka paketu 64KB. Toto omezení lze obejít pomocí tzv. Jumbo paketů.



Obrázek 3.3: Porovnání IPv4 a IPv6 hlaviček [8]

Další hlavička (Next header) Next header je 8-bitový selektor, který identifikuje typ hlavičky následující za aktuální hlavičkou. Používá stejné identifikátory jako IPv4.

Dosah (Hop limit) Maximální počet skoků nahrazuje TTL známou z IPv4 datagramu. Zpracování datagramu na každém zařízení se považuje za jeden skok. Při každém průchodu se položka sníží o jedničku. Pokud tak nabude hodnoty nula, paket se zahodí a odesílateli se pošle ICMP zpráva o dosažení maximálního počtu skoků. Tím je síť chráněna před cyklením paketů.

Adresy Zbytek hlavičky zabírají IP adresy koncových uzlů.

3.4 Rozšiřující hlavičky

IPv6 pro nastavení různých možností používá místo jednotlivých polí v základní hlavičce položku Next Header, která obsahuje typ zřetězené hlavičky. Jak je na obrázku 3.3 vidět, došlo tím ke zjednodušení a zpřehlednění celé hlavičky.

Například fragmentace z IPv4 hlavičky byla přenesena do rozšiřující hlavičky. V IPv6 se totiž předpokládá minimum fragmentace. Minimální požadované MTU cesty je totiž 1280B, což by většině aplikací mělo stačit.

Další položkou vyjmutou z hlavičky je kontrolní součet, který by byl vzhledem k počítání součtu v nižších vrstvách (např. Ethernet) redundantní a zbytečně by zatěžoval zařízení jeho neustálým přepočítáváním.

Rozšiřujících hlaviček je několik typů a každá má vlastní formát. Typ dané hlavičky je označen identifikátorem v položce Next header předcházející hlavičky. Lze tak zřetězit libovolný počet hlaviček za sebe. Poslední z hlaviček obsahuje v tomto poli označení typu dat která nese. Některé z hodnot jsou uvedeny v tabulce 3.2 a 3.3.

Typy hlaviček	
0	volby pro všechny (Hop-by-Hop options)
43	Směrování (Routing)
44	Fragmentace (Fragment)
50	Šifrování obsahu (Encapsulating Security Payload)
51	Autentizace (Authentication Header)
59	Poslední hlavička (No next header)
60	Volby pro cíl (Destination option)
135	Mobilita (Mobility)

Tabulka 3.2: Rozšiřující hlavičky

Protokoly vyšší vrstvy	
6	TCP
9	IGP
17	UDP
27	RDP
46	RSVP
47	GRE
58	ICMPv6
88	EIGRP

Tabulka 3.3: Některé z protokolů vyšší vrstvy

I když pořadí zřetězení hlaviček není nijak pevně dáno, bylo vydáno doporučení které definuje následující pořadí

- základní IPv6 hlavička
- volby pro všechny (Hop-by-Hop options)
- volby pro cíl (Destination options)
- směrování (Routing)
- fragmentace (Routing)
- autentizace (authentication)
- šifrování obsahu (encapsulation security payload)
- volby pro cíl (Destination options)
- mobilita (Mobility)

Jak je vidět, *volby pro cíl* jsou obsaženy dvakrát. První udává volby pro první a další cílové adresy datagramu uvedené v hlavičce směrování. Druhou si čte pouze konečný příjemce datagramu.

Obrázek 3.4: Formát hlavičky volby pro všechny a volby pro cíl [8]

Typ	Název
0	Pad1
1	PadN
5	Upozornění směrovače (Router alert)
38	Rychlý start (Quick start)
194	Jumbo obsah

Tabulka 3.4: Definované volby

3.4.1 Volby (Options)

Různé volby lze specifikovat ve dvou typech rozšiřujících hlaviček. Jedná se o volby pro všechny (Hop-by-Hop options s označením 0) a volby pro cíl (Destination options typu 0). Oba typy hlaviček mají stejnou strukturu zobrazenou na obrázku 3.4.

Pole další hlavička označuje typ další hlavičky v pořadí. Délka dat obsahuje informaci o velikosti připojených voleb.

Volby V této položce jsou uloženy volby. Definice IPv6 specifikuje pouze dvě volby: Pad1 a PadN. Jejich využití je vkládání nulových dat sloužících k zarovnání ostatních prvků. V dalších dokumentech byly definovány další rozšiřující možnosti (tabulka 3.4).

Formát voleb Všechny volby mají jednotný tvar. První byte definuje typ volby, druhý délku dat a zbytek obsahuje již samotná data. Struktura těchto dat musí být popsána v dokumentu definující danou volbu. První tři bity mají pevně stanovený význam. První dva popisují, co má zařízení s datagramem udělat, pokud danou volbu nezná.

- 00 - volba se přeskočí
- 01 - datagram se zahodí
- 10 - datagram se zahodí a odesílateli se pošle ICMP zpráva
- 11 - datagram se zahodí a odesílateli se pošle ICMP zpráva, pokud cílová adresa nebyla skupinová

Třetí bit určuje, zda se položka může během své cesty změnit. Pokud je nastavena na 0 měnit se nemůže a naopak.

Pad1

Tato volba vyplňuje 1 byte. Tento byte obsahuje hodnotu 0, kterým definuje svůj typ.

PadN

Pomocí PadN je možné vynechat libovolný počet bytů. První byte označuje typ volby. Následuje ho jeden byte definující délku volby. Samotná data jsou nulové byty.

Upozornění směrovače (Router alert)

Upozornění směrovače bylo definováno v RFC 2711: *IPv6 Router Alert Option* [6]. Jejím účelem je upozornit všechny směrovače po cestě na zajímavý obsah paketu. Jedná se například o zprávy RSVP. Ty rezervují přenosovou kapacitu na směrovačích po cestě paketu. Bez existence tohoto rozšíření by musel směrovač procházet celý paket, aby zjistil jaký protokol v sobě nese a ztrácel by tak čas. Byly definovány 3 základní typy

- 0 - obsahuje MLD zprávu
- 1 - obsahuje RSVP zprávu
- 2 - obsahuje zprávu aktivní sítě

Jumbogramy

Velikost dat se v IPv6 udává pomocí 16-bitového čísla. Pokud je potřeba vytvořit datagram větší, použijí se takzvané jumbogramy. Ten se vytvoří pomocí použití volby Jumbo obsah (Jumbo payload) v hlavičce Volby pro všechny.

Při použití jumbogramu se délka dat v hlavičce IPv6 vynuluje a do hlavičky volby pro všechny se vloží volba Jumbo obsah. Volba se značí jako typ 194, její délka jsou 4 byty. Do těchto 4 bytů se uloží 32-bitové číslo, které značí délku jumbo dat. Tím lze dosáhnout velikosti datagramu až 4 294 967 295 bytů.

3.4.2 Směrování (routing)

Pomocí rozšiřující hlavičky směrování lze ovlivnit cestu paketu v síti. Ve specifikaci je ponecháno prostoru pro zavedení jejich různých typů. V současné době jsou definovány dva přesně popsané a dva volné typy.

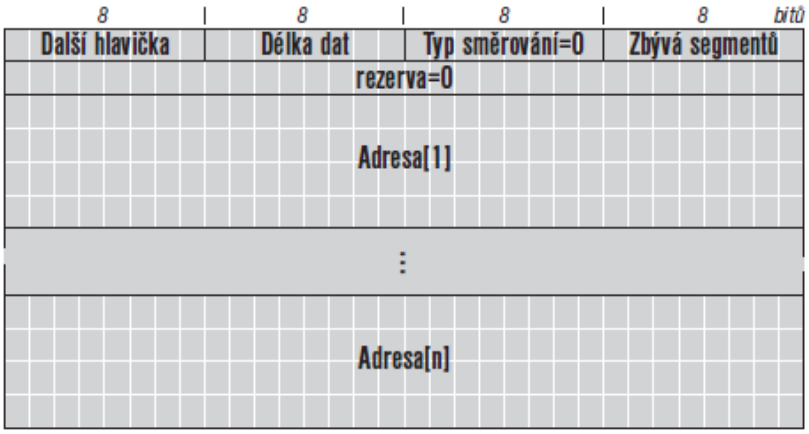
Pokud zařízení přijme paket a neumí pracovat s daným typem směrování, musí se zachovat jedním z následujících způsobů

- Pokud je počet zbývajících segmentů nulový, zařízení musí hlavičku ignorovat a pokračovat na další zřetězenou hlavičku.
- Pokud je počet zbývajících segmentů nenulový, zařízení paket musí zahodit a poslat ICMP Parameter problem s kódem nula, které označuje nerozpoznaný typ směrování

Typ 0

Typ 0 byl definován v RFC 2460 [3] jako jedna ze základních vlastností. Umožňuje vložit do datagramu seznam jedné nebo více adres uzlů, kterými má paket na své cestě projít. Podobnou možnost umožňuje i IPv4. Identifikační číslo tohoto typu hlavičky je 43. Její formát je zobrazen na obrázku 3.5.

Odesílatel při vytváření paketu vloží adresu prvního bodu jako cílovou adresu. Ostatní body, kterými má paket projít posléze vloží do rozšiřující hlavičky a nastaví počet zbývajících uzlů.



Obrázek 3.5: Formát rozšiřující hlavičky směrování [8]

Po příchodu paketu do cílové destinace (první zastávka na cestě), směrovač zjistí počet zbývajících segmentů a spočítá si kolik adres je v hlavičce uložených. Vyjme první nepoužitou adresu (od počtu adres odečte počet zbývajících segmentů) a prohodí ji s aktuální cílovou adresou (adresa routeru). To se opakuje až do stavu, kdy počet zbývajících segmentů je roven nule. V té chvíli ví, že paket dorazil do své skutečné destinace.

Pomocí směrování typu nula lze testovat zda mezi jednotlivými body existuje funkční spojení. Dále je možné směrování použít k odeslání paketu na neveřejnou adresu, kde se jako původní destinace nastaví adresa brány, která posléze prohodí adresy a paket pošle příjemci.

Tento typ směrování může být ovšem použit i k útokům na počítačové sítě. Podstata tkví v tom, že se jako destinace nastaví několik routerů a paket se nechá kolovat. Tím se dá vytvořit tak velký objem přenášených dat, že síť může zkolabovat. Z tohoto důvodu bylo vydáno RFC 5095: *Deprecation of Type 0 Routing Headers in IPv6* [1], které používání směrování typu nula upravují.

Pokud zařízení přijme paket se směrovací hlavičkou, je povinen se zachovat jedním z následujících způsobů:

- pokud je počet zbývajících segmentů nulový, uzel musí ignorovat směrovací hlavičku a pokračovat zpracováním následující hlavičky
- pokud je počet zbývajících segmentů nenulový, uzel musí paket zahodit a odeslat ICMP Parameter problém s kódem nula označující nerozpoznaný typ směrování

Dále se doporučuje, aby zařízení plošně neblokovala pakety se směrovací hlavičkou, jelikož mobilita vyžaduje směrování typu 2.

Typ 2

Směrování typu 2 bylo vyvinuto speciálně pro mobilitu. Jedná se o formát podobný směrování typu nula s tím rozdílem, že do směrovací hlavičky se smí vložit pouze jedna adresa. Pokud je mobilní zařízení přenášeno mezi různými podsítěmi, mění se jeho dočasná adresa. Mimo ní má taky svojí domácí adresu. Aby aplikace neměly narušenou komunikaci, používají domácí adresu. Komunikující uzel tak zašle paket na dočasnou adresu a do směrovací

Obrázek 3.6: Formát rozšiřující fragmentace [8]

hlavičky zapíše adresu domácí. Zařízení přijme paket na dočasné adrese, tu změní na domácí adresu udanou v hlavičce. Aplikace tak získá paket doručený na domácí adresu.

3.4.3 Fragmentace

Fragmentace slouží k přenosu datagramů větších než je MTU (Maximum Transmission Unit) přenosového média. Fragmentaci podporuje i IPv4, ovšem rozdíl je v tom, že datagramy IPv4 může fragmentovat kterýkoliv z uzlů účastnící se přenosu. V IPv6 fragmentuje datagramy výlučně odesílatel.

Pokud paket při své cestě narazí na linku s MTU menším než je jeho velikost, je zahozen a odesílateli je poslána informace o maximální velikosti podporované daným médiem.

Formát hlavičky

Formát hlavičky je zobrazen na obrázku 3.6. Obsahuje položku další hlavička, která odkazuje na typ další připojené hlavičky. Hodnota identifikující fragmentační hlavičku je 44.

Posun fragmentů (Fragment offset) Posun fragmentu je 13-bitové pole obsahující posun dat relativně k fragmentovatelné části originálního paketu.

M (More fragmets) Pole More fragments identigikuje zda se jedná o poslední část fragmentovaného paketu (1 - následují další fragmenty; 0 - jedná se o poslední fragment)

Identifikace (Identification) Identifikace slouží k rozlišení fragmentů různých zpráv. Toto 32-bitové číslo se generuje při vytváření paketu. Pokud se fragmentuje mezi stejným odesílatelem a příjemcem, je nutné, aby tato čísla byla dostatečně rozdílná.

Každý paket se dá rozdělit na fragmentovatelnou a nefragmentovatelnou část. Nefragmentovatelná část je vše co předchází fragmentovací hlavičce. Zbytek datagramu je považován za fragmentovatelnou část.

Při fragmentaci se rozdělí fragmentovatelná část na násobek osmi bytů tak, aby velikost paketu nepřekročila MTU linky. Každý nově vytvořený paket se skládá ze základní hlavičky (včetně rozšiřujících hlaviček až po směrování), ke které se přidá fragmentační hlavička a část fragmentovaných dat. Do identifikace se vloží vygenerované číslo. Údaj pro další hlavičku je použit z poslední hlavičky nefragmentovatelné části datagramu. Posun je vyjádřen jako osmice bytů o které je daný fragment posunut (první fragmentovaná část nese hodnotu nula). Pokud se jedná o poslední fragment, nastaví se příznak M na nulu.

Kapitola 4

Dostupné nástroje pro generování IPv6 paketů

V této kapitole je popsáno několik z dostupných nástrojů a knihoven pro vytváření IPv6 paketů.

SendIP

Prvním z popisovaných nástrojů je *SendIP* [14]. Jedná se o multiplatformní nástroj použitelný z příkazové řádky. Jeho úkolem je vytvářet libovolné pakety, které je následně možné vyslat po síti. Mezi podporované protokoly patří:

- IPv4
  - TCP
    - BGP
  - ICMP
  - UDP
    - RIP
    - NTP
- IPv6
  - ICMPv6
  - TCP
  - UDP
    - RIPng
    - NTP

Jak je vidět, nástroj podporuje IPv6 a základní protokoly vyšší vrstvy. Jeho poslední verze byla vydaná v roce 2003. I když se jedná o schopný nástroj, nepodporuje nativně vytváření rozšiřujících hlaviček. Je tak zapotřebí si paket vytvořit sám.

libdnet

Jedná se o knihovnu zprostředkovávající zjednodušený přístup k několika nízkoúrovňovým možnostem práce se sítí. Jedná se například o manipulaci se síťovou adresou, filtrování provozu, manipulaci se síťovým rozhraním, IP tunelování či přenos IP paketů a Ethernetových rámců.

Podporuje několik programovacích jazyků a širokou škálu operačních systémů. Tato knihovna je využívaná mnohými programy pro práci se sítí, jako je například dhcp-agent, firewalk, ip6sic, nmap, scapy a další.

Další informace a zdrojové kódy je možné nalézt na webových stránkách projektu [10].

dpkt

dpkt [9] je knihovnou implementovanou v jazyce Python. Podporuje množství protokolů, které je možné jednoduchým způsobem nastavovat a spojovat do paketů. Umožňuje také

zápis vytvořených paketů do pcap souboru.

Tento nástroj podporuje i některé rozšiřující hlavičky. Jedná se o směrování a fragmentaci. Hlavičky pro volby nejsou implementované.

scapy

Scapy [13] je další z knihoven pro Python umožňující manipulaci s pakety. Navíc zvládá odposlouchávání sítě, sledování trasy k cíli, útoky, ukládání a načítání pcap formátu a další.

Pomocí tohoto nástroje je možné vytvářet IPv6 pakety obsahující rozšiřující hlavičky volby pro všechny, volby pro cíl, směrování a fragmentace. Další z možných hlaviček čekají na implementaci. Navíc implementuje širokou škálu ICMPv6 zpráv. Jednotlivé části paketu lze jednoduše spojovat pomocí operátoru `/`.

PackETH

Jedná se o jediný popisovaný grafický nástroj [11]. Jeho pomocí se dají vytvářet pakety obsahující protokoly Ethernet II, ARP, IPv4, IPv6, UDP, TCP, ICMP, IGMP, RTP a další. Jeho rozhraní obsahuje 4 okna:

Builder slouží k vytváření paketů

Gen-b umožňuje odesílání vytvořeného paketu do sítě

Gen-s obsahuje generátor pro vysílání až 10 vytvořených paketů

Pcap umožňuje načíst pcap soubor a zobrazit jednotlivé zachycené pakety

Okno pro vytváření paketů je uspořádáno podle jednotlivých hlaviček jak jdou za sebou. V horní části je možné nastavit hlavičku linkové vrstvy. Pod ní se nachází nastavení IPv6 hlavičky a v poslední části je specifikován protokol vyšší vrstvy.

Je umožněno nastavení základní hlavičky IPv6 protokolu, nicméně žádná z rozšiřujících hlaviček není podporována. Protokol ICMPv6 není podporován vůbec.

Kapitola 5

Návrh a implementace

5.1 Využité nástroje

Aplikaci jsem se rozhodl implementovat pomocí jazyka Python [12]. Jedná se o jazyk relativně snadný a přitom velmi efektivní. Podporuje objektově orientované programování, což výrazně zjednodušuje vytváření aplikací a jejich znovupoužitelnost a rozšiřitelnost.

Samotné grafické prostředí je vytvořeno pomocí wxPython [16], které staví na wxWidgets [17]. Pro generování grafické části programu existuje několik programů, do kterých se naskládají grafické prvky a aplikace kód vytvoří. I když použití takovýchto nástrojů je velmi lákavé, rozhodl jsem se takovýchto nástrojů nevyužít a vše napsat ručně. Je tak mnohem jednoduší poskládat si přesně to rozložení, jaké je požadováno.

Z popsaných dostupných knihoven v kapitole 4 pro generování paketů, jsem nakonec vybral knihovnu Scapy. Její pomocí je totiž snadné vytvořit samotný paket. Podporuje také většinu z potřebných možností definovaných v IPv6.

5.2 wxPython

Jedná se o rozšiřující modul pro Python, který zpřístupňuje wxWidgets. Je dostupný pro většinu operačních systémů. Pomocí tohoto modulu je možné vytvářet plně funkční grafické rozhraní.

5.2.1 Ovládací prvky

Při implementaci bylo využito několika tříd. K vytvoření karet, pro výběr typu hlaviček, je použita třída `Notebook`. Jednotlivým záložkám byly pomocí metody `AddPage` přiřazeny panely obsahující nastavení daných hlaviček.

Hlavní menu je vytvořeno pomocí třídy `MenuBar`. Všechny viditelné položky jsou typu `Menu`. Tyto obsahují další položky, které jsou přiřazené k menu pomocí metody `Append`.

Pro vytvoření jednotlivých částí nastavení jsou použity panely s využitím třídy `Panel`. Takto vytvořené panely se posléze pomocí metod `Show` a `Hide` mohou zobrazovat a schovávat. Následně je potřeba rodičovský panel překreslit pomocí `Layout`.

Pokud je potřeba zadat textovou nebo číselnou hodnotu, je využito dvojice `StaticText` a `TextCtrl`. První ze zmiňovaných vytvoří needitovatelný text, kterému je nastaveno co se od uživatele vyžaduje. Vedle nebo pod něj je umístěno editovatelné pole, do kterého se zadává požadovaná hodnota. Toto pole je ve výchozím nastavení jednořádkové, ovšem pomocí nastavení parametru `TE_MULTILINE` je z něj možné udělat víceřádkový.

Výběr z několika možných nastavení je implementován pomocí `RadioBox`u nebo `Choice`. RadioBox je pole výběru, kde lze vybrat právě jednu z nabídnutých možností. Jednotlivé možnosti jsou seznamem, který je při vytváření prvku vložen jako parametr. Choice vytvoří rolovací nabídku, která po rozkliknutí zobrazí všechny možné výběry. Stejně jako u RadioBoxu jsou možnosti seznamem předaným prvku při jeho vytváření.

Posledním z ovládacích prvků je `CheckBox`. Ten je využit v případě, že daná volba má pouze dva stavy. Jeho použitím se vytvoří zatrhávací pole označující, zda byla daná položka nastavena. Tento prvek je využit při nastavování flagů, výběru vložených rozšiřujících hlaviček a povolení k přiložení síťové adresy.

5.2.2 Pozicování prvků

Pro rozložení prvků se využívá `Sizer`ů. Pomocí nich lze vkládat jednotlivé prvky na požadované místo. Aplikace využívá dvou typů: `BoxSizer` a `GridBagSizer`. Pro přidání jednotlivých prvků je využito metody `Add`.

`BoxSizer` umožňuje vkládání prvků buď horizontálně nebo vertikálně. Toho je dosaženo zadáním parametru `HORIZONTAL` nebo `VERTICAL` při jeho vytváření.

`GridBagSizer` umožňuje oproti BoxSizeru vkládání prvků na specifická místa. Toho je dosaženo použitím třídy `GBPosition` jako parametru při vkládání prvku. Jednotlivé prvky mohou také zabírat více než jednu buňku sizeru čehož je dosaženo pomocí zadání `GBSpan`.

Jednotlivé sizery lze do sebe zanořovat. Tak je možné pomocí jednoduchého vnoření několika BoxSizerů dosáhnout chování GridBagSizeru. V práci je využito obou přístupů.

5.3 Struktura kódu

Uživatelské rozhraní je rozděleno do osmi souborů obsahujících vytvořené třídy pro jednotlivé části uživatelského rozhraní. Devátý soubor obsahuje třídu, která vytváří paket pomocí knihovny Scapy a ukládá ho do souboru.

Hlavní částí, pomocí které se program spouští je `packet_creator.py`. Obsahuje vytvoření základu grafického rozhraní, do kterého jsou posléze vkládané jednotlivé části. Jedná se o `Notebook`, `MenuBar` a `StatusBar`.Dále obsahuje metody `OnSave`, `OnSaveAs` a `Settings`.

Metoda `Settings` slouží ke zjištění všech nastavení a jejich uložení do instance třídy `pck_builder.IPv6Packet`.

`OnSave` se vyvolá při vybrání položky *Save* z hlavního menu. Slouží k odeslání získaných nastavení metodě `pck_builder.BuildPacket`. Druhou možností je vybrání *Save As ...* volající `OnSaveAs`. Pomocí této metody se vyvolá dialogové okno umožňující vybrání kam daný paket uložit. Následně se zpětně zavolá metoda `OnSave`.

V souboru `ether_header.py` je umístěna jediná třída s názvem `EtherHead`. V ní je definováno rozložení prvků pro nastavení ethernetové hlavičky. Podobně je řešená i hlavička IPv6, jejíž nastavení je uloženo v souboru `ipv6_header.py` pomocí třídy `IPv6Header`.

Soubor `extension_header.py` obsahuje veškeré implementované nastavení rozšiřujících hlaviček. Hlavní třídou je `ExtensionHead`, která v sobě obsahuje zaškrtávací políčka pro výběr přidaných hlaviček. Dále pak rolovací menu s výběrem hlavičky k nastavení. Jednotlivá nastavení hlaviček jsou implementována pomocí vlastních tříd.

- Fragment
- Routing

- PadN
- Jumbo
- RouterAlert
- HomeAddress
- Options
- DestOpt

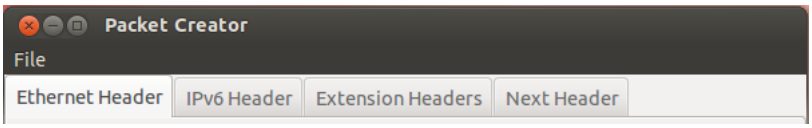
Jak je vidět, v seznamu chybí možnost nastavení `Pad1`. To je zapříčiněno tím, že pro danou volbu není zapotřebí žádného nastavení (viz popis IPv6 voleb na straně 14).

`next_header.py` obsahuje pouze jednu třídu. Jejím úkolem je provádět výběr a správné zobrazení nastavení protokolů vyšší vrstvy. Jedná se o *TCP*, *UDP* a *ICMP*. Panely vytvořené pro každou z těchto tříd jsou uloženy ve zvláštních souborech `nh_tcp.py`, `nh_udp.py` a `nh_icmp.py`. Soubory pro UDP a TCP obsahují pouze jednu třídu vytvářející rozhraní pro jejich nastavení. Oproti tomu `nh_icmp.py` obsahuje několik vytvořených tříd. Hlavní třídou je `NHICMP` sloužící pro zobrazování jednotlivých nastavení po jejich výběru. Každá implementovaná ICMP zpráva má vlastní třídu.

5.4 Vzhled a uspořádání grafického rozhraní

Grafické prostředí vychází z rozložení vytvářeného paketu. Uspořádání jednotlivých karet tak logicky odpovídá struktuře paketu. Každý paket se skládá ze čtyř částí, kterými jsou

- Hlavička Ethernetu (za předpokladu, že se paket vysílá po Ethernetu)
- IPv6 hlavička
- Rozšiřující hlavičky
- Obsah paketu



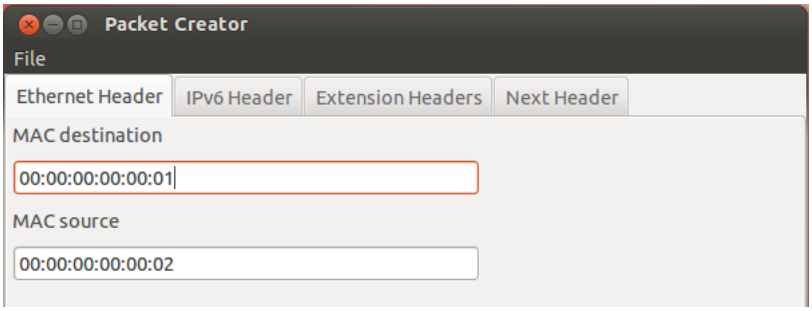
Obrázek 5.1: Rozložení a formát karet pro výběr vrstvy paketu

Každé vrstvě paketu je tak věnována vlastní karta, které jsou pojmenovány Ethernet header, IPv6 header, Extension header a Next header. Na každé z karet je nastavení dané vrstvy. Jednotlivé karty a jejich rozložení je vidět na obrázku 5.1.

5.4.1 Hlavička Ethernetu

Na kartě *Ethernet Header* je možné nastavit cílovou a výchozí MAC adresu pro adresování v síti. Vzhled této karty je vidět na obrázku 5.2.

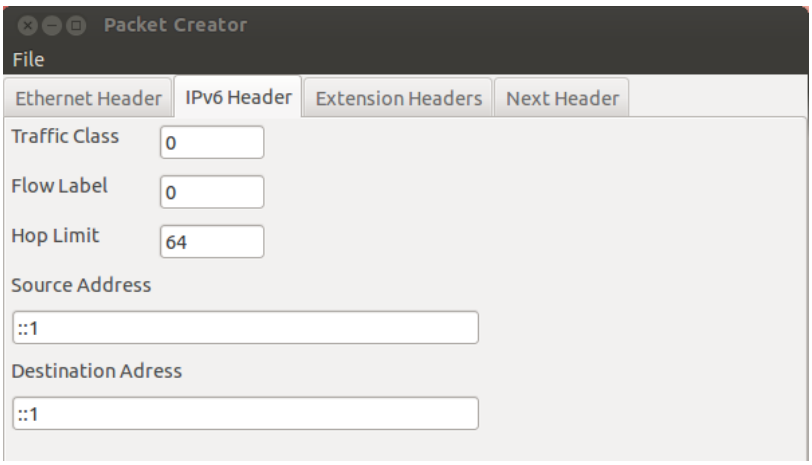
Ethernet byl vybrán proto, že je to nejčastější volba pro budování sítí.



Obrázek 5.2: Nastavení hlavičky Ethernetu

5.4.2 IPv6 hlavička

Základní hlavičce IPv6 datagramu je věnována karta s názvem *IPv6 Header*. Je zde možné nastavit třídu provozu, značku toku, maximální počet skoků, zdrojovou a cílovou adresu. Jedná se o všechny položky definované pro tuto hlavičku. Rozložení vstupních prvků je zobrazeno na obrázku 5.3.



Obrázek 5.3: Nastavení IPv6 hlavičky

5.4.3 Rozšiřující hlavičky

Při prvním otevření záložky *Extension headers* se zobrazí výčet podporovaných hlaviček jako zaškrtávací políčka. Jejich pomocí je možné vybrat která z daných hlaviček se do výsledného paketu uloží.

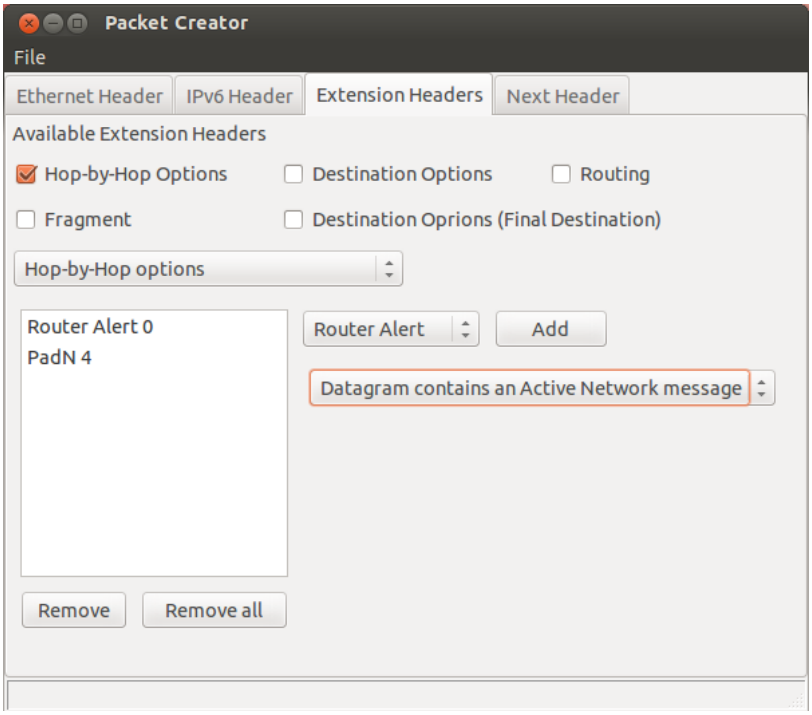
Pod výběrem hlaviček je vytvořeno rolovací menu, ze kterého lze vybrat jednotlivé hlavičky. Podporovaná nastavení daného rozšíření se zobrazí pod rolovacím menu.

Každá z hlaviček má vytvořenou vlastní konfigurační stránku, které obsahuje veškerá podporovaná nastavení v knihovně Scapy.

Volby pro všechny

Volby pro všechny a volby pro cíl sdílejí stejné rozložení. V levé části karty je zobrazen seznam přidaných voleb s možností odebrat vybranou položku nebo všechny přidané volby.

Pravá část je rozdělena na dvě podskupiny. Jedna obsahuje rolovací menu s výběrem jednotlivých menu a tlačítko přidat pro vložení vybrané volby. Druhá část obsahuje jednotlivá nastavení pro danou volbu. Na obrázku 5.4 je zobrazen příklad, kde je vybráno přidání hlavičky voleb pro všechny, která má již přidáno volby *Router Alert: Datagram contains a MLD message* a *PadN* s délkou dat 4. V části pro přidání dalších voleb je vybráno Upozornění směrovače poukazující na zprávu aktivní sítě v datagramu.



Obrázek 5.4: Nastavení Voleb pro všechny

Směrováni

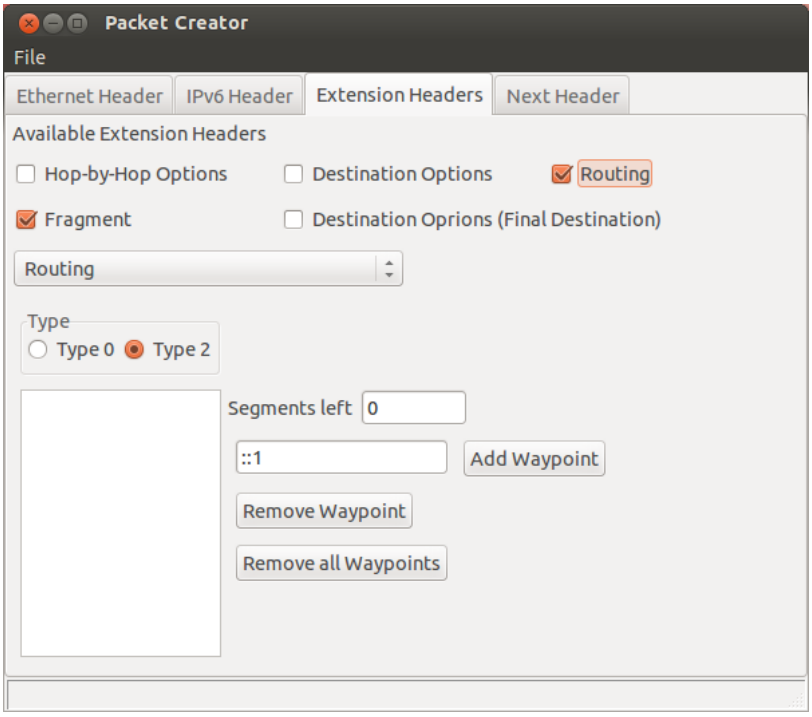
Další z možných rozšiřujících hlaviček je směrování. Prvky pro nastavení mají podobné rozložení jako Volby pro všechny. V levé části se nachází seznam již přidaných uzlů, kterými má daný paket projít. Nad ním se nachází výběr typu směrování. Jelikož byl typ 0 v RFC 5095 [1] odmítnut, je výchozím nastavením typ 2.

Na obrázku 5.5 je zobrazen příklad, kdy je vybrán typ 2 a počet zbývajících segmentů je nastaven na nula.

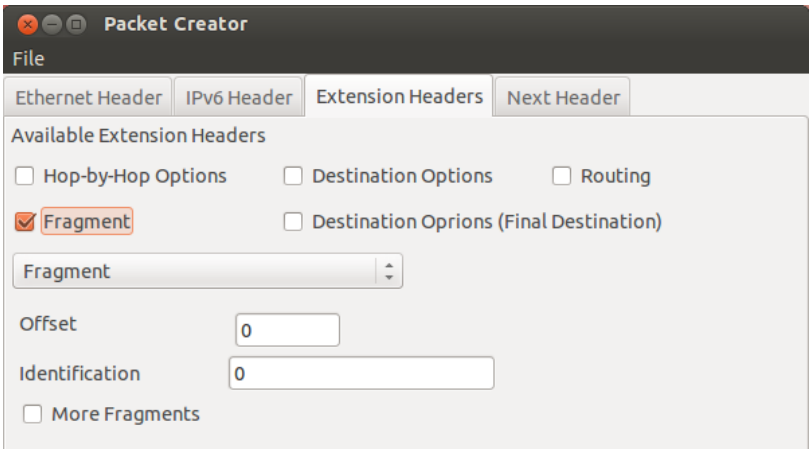
Jelikož typ 2 zakazuje přidání více než jedné adresy, je při přepnutí na tento typ uživatel v případě několika vybraných adres upozorněn, že musí některé z daných adres odebrat. Pokud je počet zbývajících segmentů nastaven na více než je počet přidaných adres, nastaví se při ukládání paketu místo zadaného čísla skutečný počet adres.

Fragmentace

Pokud je uživatelem vyžadována fragmentace, je jí možno nastavit vybráním položky *Fragment* z rolovacího menu. Jejím vybráním se zobrazí možnosti, které obsahují posun fragmentů, identifikaci paketu v rámci fragmentovaného datagramu a zaškrtávací políčko označující zda se jedná o poslední fragment. Vzhled nastavení je možné vidět na obrázku 5.6.



Obrázek 5.5: Nastavení směrování



Obrázek 5.6: Nastavení fragmentace

5.4.4 Protokol vyšší vrstvy

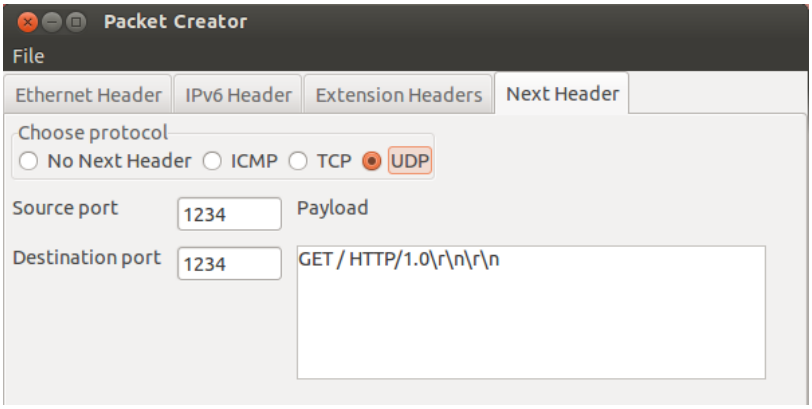
Protokol vyšší vrstvy lze specifikovat na kartě *Next Header*. V horní části se nachází radio box, který slouží k výběru jednoho z protokolů vyšší vrstvy. Vzhledem k tomu, že v práci byl kladen důraz na podporu IPv6. byly implementovány pouze tři protokoly a to ICMP, TCP a UDP. Pokud není vyžadováno vložení žádného z těchto protokolů, je možné vybrat možnost *No next header*, díky které nebude do paketu vložen žádný z výše zmiňovaných.

UDP

Pro nastavení *UDP* protokolu slouží tři pole

- Zdrojový port (Source port)
- Cílový port (Destination port)
- Přenášená data (Payload)

Rozložení nastavení je zobrazeno na obrázku 5.7



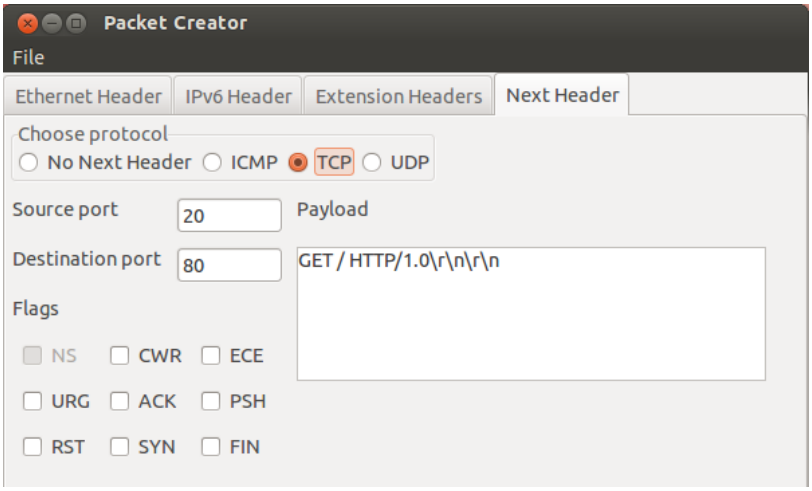
Obrázek 5.7: Nastavení UDP

TCP

Pokud je vybrána možnost *TCP*, je zobrazená podobná karta jako pro UDP. Navíc má vloženo pole pro výběr jednotlivých bitových nastavení TCP zprávy (viz obrázek 5.8).

Význam jednotlivých nastavení (Flags) je následující [7]

- URG (Urgent Data): indikuje přenášení urgentních dat
- ACK (Acknowledge): indikuje platné číslo potvrzení
- PSH (Push data): indikuje požadavek co nejrychlejšího doručení dat aplikační vrstvě
- RST (Reset): indikuje požadavek na reset virtuálního spojení, aplikace mají spojení ukončit, nepředstavuje standardní prostředek ukončení komunikace
- SYN (Synchronize): požadavek na vytvoření spojení
- FIN (Finish): požadavek na ukončení spojení



Obrázek 5.8: Nastavení TCP

ICMP

Důležitým doplňkem protokolu IPv6 jsou ICMP zprávy sloužící k oznamování různých zpráv a k jejich vyžádání. Podobně jako protokoly zmíněné výše má i ICMP vlastní nastavení. Implementováno je několik zpráv souvisejících s IPv6 (tabulka 5.1).

Typ	Název
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
128	Echo Request
129	Echo Reply
130	Multicast Listener Query
131	Multicast Listener Report
132	Multicast Listener Done
133	Router Solicitation
134	Router Advertisement
135	Neighbor Soliditation
136	Neighbor Advertisement
137	Redirect

Tabulka 5.1: Implementované ICMP zprávy

I tato část dodržuje jednotnou strukturu, kde obecnější prvky jsou v horní části okna aplikace a specifičtější ve spodní. Jednotlivé typy zpráv se vybírají pomocí rolovacího menu. Při výběru se v okně objeví sada nastavovacích prvků dané zprávy.

Destination Unreachable V této zprávě lze nastavit jednu z pěti podporovaných zpráv. Jedná se o

- No route to destination
- Communication with destination administratively prohibited
- Beyond scope of source address
- Address unreachable
- Port unreachable

Ke každé z těchto položek se dá pomocí textového pole přiřadit zpráva.

Packet Too Big Nastavení pro tuto položku obsahuje MTU a přiloženou zprávu.

Time Exceeded Zde lze vybrat jeden z dvou typů: Překročení maximálního počtu skoků při přenosu paketu nebo překročení maximálního času složení fragmentovaného paketu.

Parametr Problem Tato zpráva může být jednoho ze tří typů

- Erroneous header field encountered
- Unrecognized Next Header type encountered
- Unrecognized IPv6 option encountered

Důležitým polem je zadání ukazatele na chybné pole (jedná se počet bytů od začátku datagramu).

Echo Request a Echo Reply Jedná se o dvě informační zprávy testující dostupnost koncového zařízení. Zde lze zadat pouze tělo zprávy.

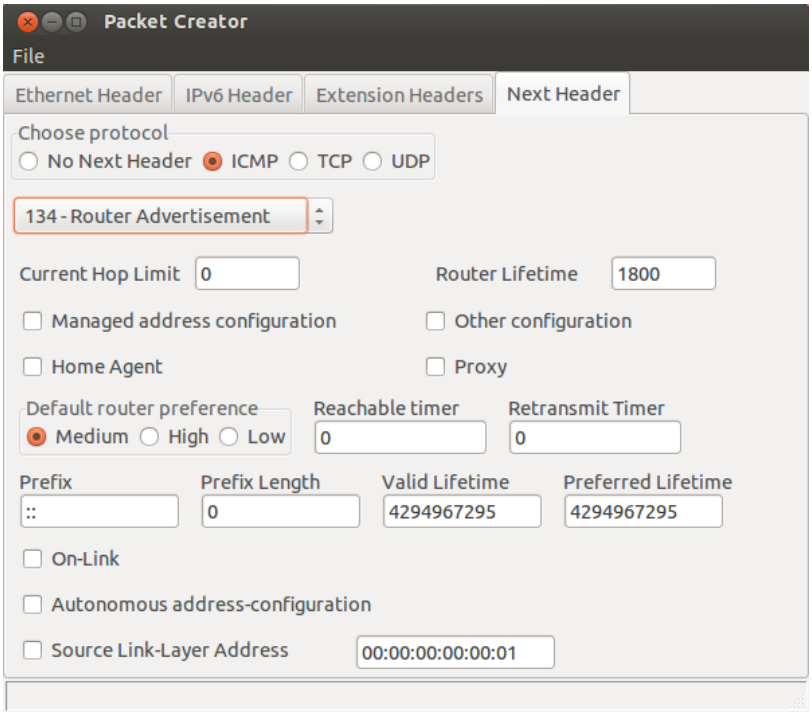
Multicast Listener Discovery V MLD jsou zahrnuty tři zprávy: Query(130), Report(131) a Done(132). Všechny sdílejí stejné nastavení a liší se pouze v typu zprávy. Možná nastavení zahrnují dvě položky:

- Maximum response delay (Maximální zpoždění zprávy)
- Multicast listener address (Skupinová adresa)

Router Solicitation a Router Advertisement Zpráva Router solicitation (Výzva směrovači k ohlášení) nemá implementované žádné volby. Slouží k urychlení vyslání Router Advertisement. Směrovač rozesílá Router Advertisement (Ohlášení směrovače) periodicky nebo na žádost koncových stanic. Tato zpráva má množství voleb, které je potřeba logicky rozložit. Jejich rozložení se podobá postavení v datagramu.

- Current Hop Limit
- Router Lifetime
- Managed address configuration
- Other configuration

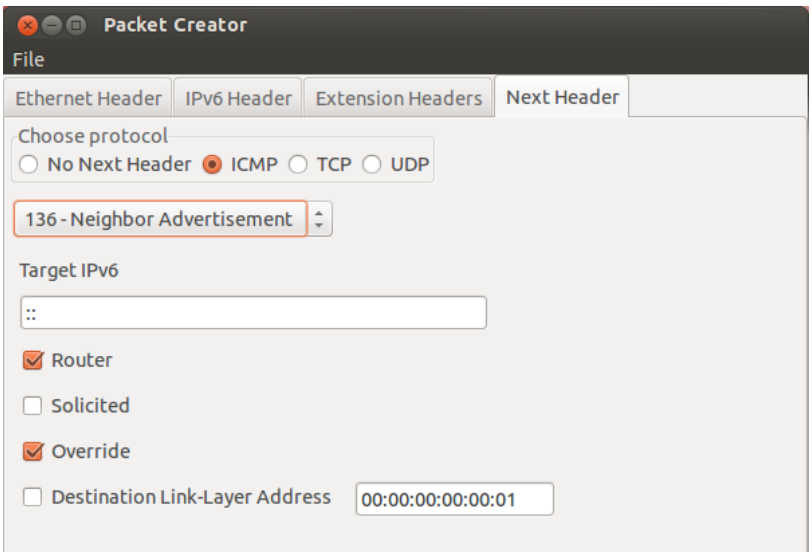
- Home agent
- Proxy
- Default router preference
- Reachable timer
- Retransmit timer
- Prefix
- Prefix length
- Valid lifetime
- Preferred Lifetime
- On-Link
- Autonomous addresss-configuration
- Source Link-Layer Address



Obrázek 5.9: Nastavení oznámení směrovače

Na obrázku 5.9 lze názorně vidět rozložení jednotlivých voleb v okně. Tímto rozložením je dobře využita celá dostupná plocha.

Neighbor solicitation a Neighbor Advertisement Tyto zprávy slouží k objevování a ohlašování sousedů v síti. Výzva sousedovi obsahuje pouze dvě položky: Adresu cíle a zdrojovou MAC adresu, která není povinná. Proto je u této volby zaškrtávací box, které určuje zda se adresa přidá či ne. Ohlašování souseda má již více položek. První z nich je adresa na kterou je oznámení vysláno. Další tři jsou volby realizované pomocí zaškrtávacích boxu. Pomocí nich lze označit ohlašovaný uzel za směrovač, označit zprávu jako vyžádanou. Poslední z trojice slouží k vynucení aktualizace adresy linkové vrstvy příjemcem. Poslední položkou je síťová adresa příjemce, která nemusí být připojena. Proto je opět přítomen zaškrtávací box pro povolení volby.



Obrázek 5.10: Nastavení ohlášení souseda

Redirect Poslední implementovanou možností je možnost přesměrování. Tato nastavení obsahuje pouze dvě pole. Zdrojovou a cílovou adresu.

5.4.5 Uložení paketu do pcap souboru

Vytvořený paket je možné uložit do pcap souboru dvěma způsoby a je realizován pomocí knihovny Scapy. První slouží jako rychlé uložení. Je realizován pomocí menu *File* → *Save*. Tím se provede načtení všech provedených nastavení. Pomocí těchto nastavení je pak knihovnou Scapy vygenerován výsledný IPv6 paket a provede se jeho uložení do *./packet.pcap*. Pokud však uživatel již použil funkci *Save As ...*, uloží se výsledný paket do vybraného souboru.

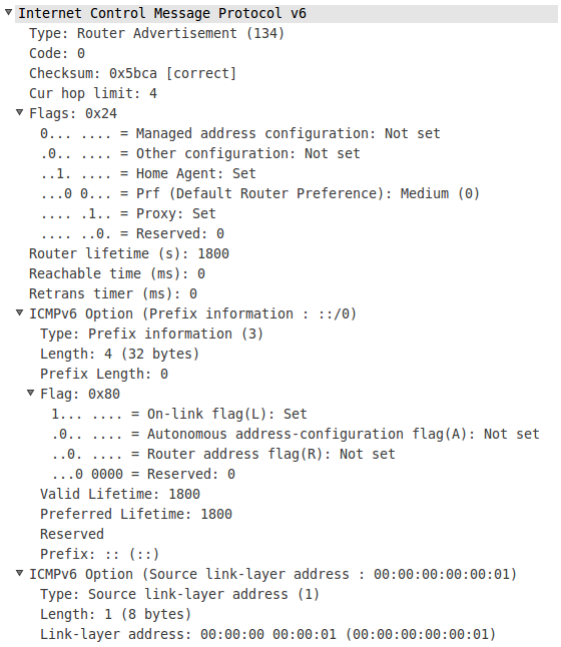
Druhý způsob je využít funkce *File* → *Save As ...*, která vyvolá dialog pro vytvoření souboru. Zadaná cesta a název souboru se předají funkci pro uložení (viz předchozí odstavec) a paket se uloží.

Kapitola 6

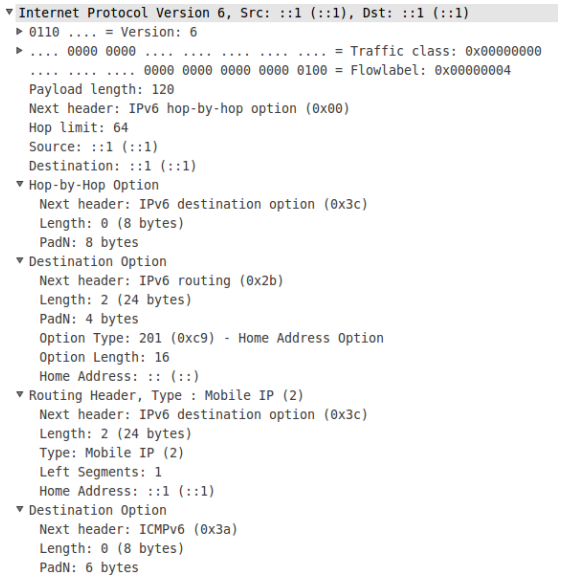
Dosažené výsledky

Obsahem této kapitoly je zhodnocení výsledné aplikace. Jedná se o její výslednou funkčnost a použitelnost.

Základním požadavkem práce bylo vytvoření uživatelského rozhraní pro některou z veřejně dostupných knihoven. Pokryty byly všechny rozšiřující hlavičky, které je program Scapy schopen generovat. Je možné jejich plné nastavování a úprava. Vygenerované pakety byly otestovány pomocí programu Wireshark [15]. Příklad takovéhoto paketu je vidět na obrázcích 6.1 a 6.2.



Obrázek 6.1: ICMPv6 Router Advertisement



Obrázek 6.2: IPv6 s rozšiřujícími hlavičkami

Jak je vidět, nástroj vytváří korektní pakety. Navíc se při řazení rozšiřujících hlaviček drží doporučení z RFC 2460 [3].

Jak bylo v kapitole 5 Návrh a implementace ukázáno, prostředí aplikace je logicky řazeno ze shora dolů, kde v horních částech jsou obecné volby a ve spodních specifické. Díky tomuto rozložení je orientace v aplikaci rychlá a snadná.

6.1 Porovnání možností oproti dalším nástrojům

Ve srovnání s výše popisovanou aplikací PackETH působí vytvořený nástroj poněkud minimalisticky. Neobsahuje možnost posílání paketů do sítě ani jejich načtení a prohlížení. Nicméně na druhou stranu obsahuje daleko více možností pro vytvoření IPv6 datagramu.

Grafické rozhraní vytvořené aplikace používá podobný přístup ze shora dolů. Tím je možné logicky rozdělit volby, které závisí na nastavení ovládacích prvků umístěných výše v aplikaci. Program PackETH ovšem využívá pouze jednoho okna pro nastavení hlaviček všech vrstev, oproti vytvořenému nástroji, který má tyto volby rozdělené do několika karet.

Jedná se o možnosti připojení rozšiřujících hlaviček volby pro všechny, volby pro cíl, routing a fragmentace, které aplikace PackETH nepodporuje vůbec. Dalším podporovaným prvkem je možnost připojit některé z ICMPv6 zpráv.

Kapitola 7

Závěr

Cílem práce bylo vytvořit grafický nástroj pro generování IPv6 paketů. Základním požadavkem pro úspěšné vytvoření takovéto aplikace je seznámení se se vzhledem IPv6 datagramu a jeho návazností na TCP/IP model. Model TCP/IP je popsán v kapitole 2. Ta obsahuje vývoj daného modelu a jeho jednotlivé vrstvy.

Protokolu IPv6 se věnuje kapitola 3. Je zde popsán vývoj vlastního protokolu, vzhled a formality zápisu adres, podoba základní hlavičky a formát rozšiřujících hlaviček včetně jejich doporučeného řazení.

Dalším krokem při vytváření aplikace bylo nalezení již dostupných nástrojů a jejich následné zhodnocení. Tomuto se věnuje kapitola 4. Jsou zde popsány čtyři knihovny a jeden grafický nástroj.

Na základě získaných poznatků byl pro realizaci práce s paketem vybrán Python modul *scapy* [13]. Z popsaných nástrojů je totiž nejlépe integrovaný do jazyka Python, ve kterém je celá práce vytvořená. Navíc má taky nejširší podporu IPv6. Pro tvorbu vlastního rozhraní byla použita knihovna wxPython [16]. Tvorba grafického rozhraní je popsána v kapitole 5.

V poslední části práce (kapitola 6) je diskutována výsledná funkčnost vytvořeného programu. Zároveň je zde obsaženo i porovnání funkčnosti a rozložení s nástrojem PackETH [11].

Výsledkem celého projektu je vytvoření funkčního grafického nástroje, pomocí kterého je možné sestavit paket obsahující IPv6 datagram včetně několika rozšiřujících hlaviček. Podporováno je také několik typů ICMPv6 zpráv, jak společných s ICMP pro IPv4 tak i specifických pro použití společně s IPv6.

Možnosti rozšíření

Díky použití objektů pro vytváření jednotlivých panelů nastavení, je rozšíření stávající aplikace relativně jednoduché. Stačí naprogramovat třídu typu `Panel` a vložit do ní jednotlivé prvky. Následně upravit metodu `Settings` v souboru `packet_creator.py` tak, aby byla nové možnosti schopná načíst a správně je zpracovat. Posledním krokem je upravit nadřazenou třídu tak, aby umožňovala zobrazit nově vytvořený panel. Pokud se ve Scapy objeví nové možnosti, je jejich implementace velmi jednoduchá.

Další možností rozšíření je schopnost vytvářet sady paketů. V momentálním návrhu se počítá s generováním pouze jednoho paketu. Takovéto rozšíření by vyžadovalo vytvoření dalšího okna, ve kterém by se specifikovaly parametry generovaných paketů.

Literatura

[1] Abley; Savola; Neville-Neil: Deprecation of Type 0 Routing Headers in IPv6. http://tools.ietf.org/html/rfc5095.

[2] Deering; Hinden: Internet Protocol, Version 6 (IPv6) Specification. http://tools.ietf.org/html/rfc1883.

[3] Deering; Hinden: Internet Protocol, Version 6 (IPv6) Specification. http://tools.ietf.org/html/rfc2460.

[4] Deering; Hinden: IP Version 6 Addressing Architecture. http://tools.ietf.org/html/rfc4291.

[5] Kawamura; Kawashima: A Recommendation for IPv6 Address Text Representation. http://tools.ietf.org/html/rfc5952.

[6] Petridge; Jackson: IPv6 Router Alert Option. http://tools.ietf.org/html/rfc2711.

[7] Ráb; Ryšavý: IPK Přednáška 4: Spolehlivý přenos dat. https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IPK-IT/lectures/ipk2011_p4-transportni_vrstva_cast2.pdf.

[8] Satrapa, P.: *IPv6*. CZ.NIC, 2011, iSBN 978-80-904248-4-5.

[9] WWW stránky: dpkt - pytrho packet creation / parsing library. http://code.google.com/p/dpkt/.

[10] WWW stránky: libdnet. http://libdnet.sourceforge.net/.

[11] WWW stránky: PackETH. http://packeth.sourceforge.net/.

[12] WWW stránky: Python Programming Language. http://www.python.org/.

[13] WWW stránky: Scapy. http://code.google.com/p/dpkt/.

[14] WWW stránky: SendIP 2.5. http://www.earth.li/projectpurple/progs/sendip.html.

[15] WWW stránky: Wireshark ·Go deep. http://www.wireshark.org/.

[16] WWW stránky: wxPython. http://wxpython.org/.

[17] WWW stránky: wxWidgets, Cross-Platform GUI Library. http://www.wxwidgets.org/.

Příloha A

Obsah CD

- Zdrojové kódy aplikace
- Text práce v PDF a TeX
- Obázky použité při tvorbě textu

Příloha B

Instalace prostředí

Pro úspěšné spuštění aplikace je nutné mít nainstalovaný Python (python2.7), modul wxPython (python-wxgtk2.8) a Scapy (python-scapy). Aplikace byla vyvíjena na Ubuntu 12.04 64bit. Všechny výše zmíněné balíčky jsou dostupné pomocí `apt`.