

Vysoké učení technické v Brně
Fakulta informačních technologií

POČÍTAČOVÉ KOMUNIKACE A SÍTĚ
2020/2021

Projekt IPK 2
Zadání DELTA – Scanner síťové dostupnosti

Tomáš Milostný (xmilos02)

Obsah

Obsah	2
Úvod	3
Popis implementace	4
ArgumentParser	4
SubnetParser	4
Subnet	5
NetworkScanner	5
IcmpPacket	5
Testování	6
Příklad s IPv4	6
Příklad s IPv6	8
Seznam literatury	9

Úvod

Projekt je implementovaný v jazyce **C# 9.0** nad platformou **.NET 5.0**. Využívá knihovny *System.CommandLine* (viz *ArgumentParser*) a *SharpPcap* (viz *NetworkScanner*).

Program skenuje zadané rozsahy IP adres pomocí protokolů ICMP/ICMPv6 a ARP. Oproti původnímu zadání nebyla implementována práce s protokolem NDP pro IPv6 subnety.

Popis implementace

Jádrem projektu je hlavní program v souboru **Program.cs** psaný ve stylu *Top-level statements* představeném v C# 9.0. Program začíná načítáním argumentů příkazové řádky pomocí třídy **ArgumentParser** a její statické metody **ParseArguments**, která vrací trojici zpracovaných argumentů:

- **NetworkInterface @interface** s informacemi o zadaném síťovém rozhraní.
- **int timeout** označující časový limit v milisekundách pro jednotlivé sockety vytvářené později.
- **Subnet[] subnets** – pole s objekty typu **Subnet** vytvořených ze všech sítí zadaných argumenty **--subnet/-s**.

ArgumentParser může vyhazovat výjimky, které jsou tedy ošetřeny v try-catch bloku, více o těchto výjimkách v sekci **ArgumentParser**. Výjimka pro chybějící parametr **--interface/-i** je ošetřena voláním funkce **PrintAllInterfaces**, která vypíše všechna dostupná rozhraní a ukončí program.

Pokud jsou všechny argumenty příkazové řádky v pořádku, pokračuje se voláním funkce **ListScanningRanges**, která vypíše všechny načtené subnety a počet IP adres v jejich rozsahu. Následuje funkce **ScanNetwork**, kde se pro jednotlivé subnety vytváří instance třídy **NetworkScanner** a zavolá se její metoda **ScanAsync**, která projde rozsah IP adres a zjistí dostupnost pomocí protokolů ICMP/ICMPv6 a ARP (více v sekci **NetworkScanner**).

ArgumentParser

Třída se statickou metodou **ParseArguments**, která pole **string[] args** pomocí knihovny **System.CommandLine [1]** převede na trojici **NetworkInterface**, **int**, **Subnet[]**. Zpracovává argumenty:

- **--interface/-i** – Název rozhraní parsovaný na objekt typu **NetworkInterface**. Chybějící nebo argument zadaný špatným jménem rozhraní vyhodí výjimku **InvalidOperationException**, kterou generuje Linq metoda **First** vyhledávající v seznamu dostupných síťových rozhraní
- **--wait/-w** – Celé číslo reprezentující časový limit v milisekundách pro sockety (výchozí hodnota 5000ms). Pokud je zadaná hodnota záporná, je vyhozena výjimka **IndexOutOfRangeException**.
- **--subnet/-s** – IPv4 nebo IPv6 adresa s maskou parsovaná na objekt typu **Subnet** třídou **SubnetParser**, může jich být zadáno více než 1
- **--help/-h/-?** – výpis nápovědy programu
- **--version** – přidáno knihovnou **System.CommandLine**, vypíše verzi programu

SubnetParser

Třída se statickou metodou **ParseSubnets**, která podle řetězců v poli **string[] subnets** vytvoří pole objektů typu **Subnet**. Tato metoda je navržena asynchronně, což jí umožňuje pracovat parametry v poli paralelně privátní statickou metodou **ParseSubnet** a vliv počtu parsovaných subnetů je tedy minimální. Rovněž během vytváření objektu kontroluje validitu zadané IP adresy a velikosti masky sítě.

Subnet

Třída držící informace o skenované síti. Udrží IP adresu zadanou parametrem a masku podsítě, která je privátní metodou **CreateMask** vytvořena jako pole bytů na základě zadaného číselného parametru délky masky a verzi protokolu IP, tedy IPv4 nebo IPv6. Tato maska je následně privátní metodou **ApplyMask** aplikována na zadanou IP adresu bitovým maskováním, které vynuluje případné přebytečné bity nepatřící do adresy sítě. Umožňuje rovněž inkrementaci IP adresy operátorem **++** a veřejnou metodu **IsAtMaxIpAddress** pro zjištění, zda je aktuální adresa na poslední možné v rozsahu daném maskou.

NetworkScanner

Třída síťového skeneru. Konstruktor vyžaduje síťové rozhraní, časový limit pro sockety a skenovanou síť reprezentovanou objektem typu **Subnet**. Dle verze IP skenované sítě se z rozhraní uloží IP adres skenujícího zařízení a verze protokolu ICMP.

Samotné skenování je spuštěno voláním metody **ScanAsync**, která je navržena asynchronně, takže umožňuje daný rozsah IP adres oskenovat paralelně. Pro každou IP adresu z rozsahu subnetu je vytvořen task **EchoAsync**, který vrací čtveřici určující úspěšnost ICMP požadavku a ARP požadavku, cílovou IP adresu pro zpětnou identifikaci ve výpisu výsledků a MAC adresu získanou z požadavku ARP.

EchoAsync začíná vytvořením raw ICMP socketu, napojením zdrojové (**Bind**) a cílové (**ConnectAsync**) IP adresy, a odesláním (**SendAsync**) packetu vytvořeným strukturou **IcmpPacket**. Poté čeká na odpověď (**ReceiveAsync**), které je rovněž předán parametr **CancellationToken** s nastaveným časovým limitem.

Pokud nedošlo ke zrušení operace, je délka přijatých dat uložena do proměnné **icmpLength**, a pokud je větší než 0, je toto volání označeno za úspěch. Pro IPv4 je následně s pomocí knihovny **SharpPcap** vytvořen v metodě **GetArp** vytvořen a zachycen ARP [2] dotaz na MAC adresu zařízení, která je metodou vrácena, pokud nedošlo k chybě nebo vypršení časového limitu. V případě úspěšného získání MAC adresy, je ARP požadavek označen za úspěch a **EchoAsync** končí s výsledky, které jsou zobrazeny na standardní výstup.

IcmpPacket

Struktura ICMP packetu [3]. Obsahuje statickou metodu **EchoRequestAsBytes**, která tuto strukturu převede na pole bytů odeslané raw socketem. V této metodě je vytvořena struktura packetu, kde jsou vyplněny potřebné atributy.

TypeCode je prvních 16 bitů značící typ echo požadavku lišící se podle verze protokolu ICMP [4] nebo ICMPv6 [5]. **Checksum** [6] je spočítána jako negovaný součet **TypeCode**, **Id**, **Sequence**.

Poté je pomocí metod **System.Runtime.InteropServices.Marshal** pro práci s automaticky nespravovanou pamětí převedena struktura packetu na binární reprezentaci odeslanou v raw socketu.

Testování

Příklad s IPv4

Spuštění programu: `dotnet run -- --interface "Wi-Fi" -s 192.168.1.224/24`
Možno vysledovat ICMP požadavky i odpovědi v aplikaci Wireshark.

The image displays two screenshots of the Wireshark network protocol analyzer, showing a capture of ICMP traffic on a Wi-Fi interface. The top screenshot shows a list of packets, with packet 221 selected. The bottom screenshot shows packet 230 selected. Both screenshots show the packet details and hex data.

Top Screenshot (Packet 221):

No.	Time	Source	Destination	Protocol	Length	Info
9	1.630232	192.168.1.203	192.168.1.1	ICMP	42	Echo (ping) request id=0x0100, seq=256/1, ttl=128 (reply in 10)
10	1.632779	192.168.1.1	192.168.1.203	ICMP	42	Echo (ping) reply id=0x0100, seq=256/1, ttl=64 (request in 9)
11	1.637507	192.168.1.203	192.168.1.54	ICMP	42	Echo (ping) request id=0x3600, seq=13824/54, ttl=128 (reply in 19)
12	1.638700	192.168.1.203	192.168.1.64	ICMP	42	Echo (ping) request id=0x4000, seq=16384/64, ttl=128 (no response found!)
13	1.645887	192.168.1.203	192.168.1.126	ICMP	42	Echo (ping) request id=0x7e00, seq=32256/126, ttl=128 (reply in 215)
14	1.648076	192.168.1.203	192.168.1.147	ICMP	42	Echo (ping) request id=0x9300, seq=37632/147, ttl=128 (reply in 20)
15	1.648206	192.168.1.203	192.168.1.148	ICMP	42	Echo (ping) request id=0x9400, seq=37888/148, ttl=128 (no response found!)
16	1.651970	192.168.1.203	192.168.1.185	ICMP	42	Echo (ping) request id=0xb900, seq=47360/185, ttl=128 (no response found!)
17	1.653578	192.168.1.203	192.168.1.199	ICMP	42	Echo (ping) request id=0xc700, seq=50944/199, ttl=128 (reply in 18)
18	1.656767	192.168.1.199	192.168.1.203	ICMP	56	Echo (ping) reply id=0xc700, seq=50944/199, ttl=64 (request in 17)
19	1.690871	192.168.1.54	192.168.1.203	ICMP	56	Echo (ping) reply id=0x3600, seq=13824/54, ttl=255 (request in 11)
20	1.724325	192.168.1.147	192.168.1.203	ICMP	56	Echo (ping) reply id=0x9300, seq=37632/147, ttl=255 (request in 14)
215	1.919023	192.168.1.126	192.168.1.203	ICMP	56	Echo (ping) reply id=0x7e00, seq=32256/126, ttl=64 (request in 13)
221	4.133541	192.168.1.203	192.168.1.224	ICMP	42	Echo (ping) request id=0xe000, seq=57344/224, ttl=128 (reply in 230)
222	4.134443	192.168.1.203	192.168.1.230	ICMP	42	Echo (ping) request id=0xe600, seq=58880/230, ttl=128 (reply in 223)
223	4.137173	192.168.1.230	192.168.1.203	ICMP	42	Echo (ping) reply id=0xe600, seq=58880/230, ttl=128 (request in 222)
230	4.239390	192.168.1.224	192.168.1.203	ICMP	56	Echo (ping) reply id=0xe000, seq=57344/224, ttl=64 (request in 221)

Frame 221: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{02C4894D-EBC0-4D5D-9AD7-94DA7218CC88}, id 0
> Ethernet II, Src: LiteonTe_af:7d:51 (70:c9:4e:af:7d:51), Dst: Raspberr_a0:c0:77 (b8:27:eb:a0:c0:77)
> Internet Protocol Version 4, Src: 192.168.1.203, Dst: 192.168.1.224
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x37fe [correct]
[Checksum Status: Good]
Identifier (BE): 57344 (0xe000)
Identifier (LE): 224 (0x00e0)
Sequence Number (BE): 57344 (0xe000)
Sequence Number (LE): 224 (0x00e0)
[\[Response frame: 230\]](#)

0000 b8 27 eb a0 c0 77 70 c9 4e af 7d 51 08 00 45 00 ...mp N:]Q:~E
0010 00 1c bd fb 00 00 00 01 f7 e9 c8 a8 01 cb c0 a8@.....

Wi-Fi: <live capture in progress> Packets: 421 · Displayed: 17 (4.0%) Profile: Default

Bottom Screenshot (Packet 230):

No.	Time	Source	Destination	Protocol	Length	Info
9	1.630232	192.168.1.203	192.168.1.1	ICMP	42	Echo (ping) request id=0x0100, seq=256/1, ttl=128 (reply in 10)
10	1.632779	192.168.1.1	192.168.1.203	ICMP	42	Echo (ping) reply id=0x0100, seq=256/1, ttl=64 (request in 9)
11	1.637507	192.168.1.203	192.168.1.54	ICMP	42	Echo (ping) request id=0x3600, seq=13824/54, ttl=128 (reply in 19)
12	1.638700	192.168.1.203	192.168.1.64	ICMP	42	Echo (ping) request id=0x4000, seq=16384/64, ttl=128 (no response found!)
13	1.645887	192.168.1.203	192.168.1.126	ICMP	42	Echo (ping) request id=0x7e00, seq=32256/126, ttl=128 (reply in 215)
14	1.648076	192.168.1.203	192.168.1.147	ICMP	42	Echo (ping) request id=0x9300, seq=37632/147, ttl=128 (reply in 20)
15	1.648206	192.168.1.203	192.168.1.148	ICMP	42	Echo (ping) request id=0x9400, seq=37888/148, ttl=128 (no response found!)
16	1.651970	192.168.1.203	192.168.1.185	ICMP	42	Echo (ping) request id=0xb900, seq=47360/185, ttl=128 (no response found!)
17	1.653578	192.168.1.203	192.168.1.199	ICMP	42	Echo (ping) request id=0xc700, seq=50944/199, ttl=128 (reply in 18)
18	1.656767	192.168.1.199	192.168.1.203	ICMP	56	Echo (ping) reply id=0xc700, seq=50944/199, ttl=64 (request in 17)
19	1.690871	192.168.1.54	192.168.1.203	ICMP	56	Echo (ping) reply id=0x3600, seq=13824/54, ttl=255 (request in 11)
20	1.724325	192.168.1.147	192.168.1.203	ICMP	56	Echo (ping) reply id=0x9300, seq=37632/147, ttl=255 (request in 14)
215	1.919023	192.168.1.126	192.168.1.203	ICMP	56	Echo (ping) reply id=0x7e00, seq=32256/126, ttl=64 (request in 13)
221	4.133541	192.168.1.203	192.168.1.224	ICMP	42	Echo (ping) request id=0xe000, seq=57344/224, ttl=128 (reply in 230)
222	4.134443	192.168.1.203	192.168.1.230	ICMP	42	Echo (ping) request id=0xe600, seq=58880/230, ttl=128 (reply in 223)
223	4.137173	192.168.1.230	192.168.1.203	ICMP	42	Echo (ping) reply id=0xe600, seq=58880/230, ttl=128 (request in 222)
230	4.239390	192.168.1.224	192.168.1.203	ICMP	56	Echo (ping) reply id=0xe000, seq=57344/224, ttl=64 (request in 221)

Frame 230: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{02C4894D-EBC0-4D5D-9AD7-94DA7218CC88}, id 0
> Ethernet II, Src: Raspberr_a0:c0:77 (b8:27:eb:a0:c0:77), Dst: LiteonTe_af:7d:51 (70:c9:4e:af:7d:51)
> Internet Protocol Version 4, Src: 192.168.1.224, Dst: 192.168.1.203
Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0x3ffe [correct]
[Checksum Status: Good]
Identifier (BE): 57344 (0xe000)
Identifier (LE): 224 (0x00e0)
Sequence Number (BE): 57344 (0xe000)
Sequence Number (LE): 224 (0x00e0)
[\[Request frame: 221\]](#)
[Response time: 105,849 ms]

0000 70 c9 4e af 7d 51 b8 27 eb a0 c0 77 08 00 45 00 p-N]Q:~E
0010 00 1c d5 63 00 00 40 01 20 82 c0 a8 01 e0 c0 a8 ...c-@.....

Wi-Fi: <live capture in progress> Packets: 543 · Displayed: 17 (3.1%) Profile: Default

```

Administrator: PowerShell
PS C:\Users\tommi\IPK\IPK_2021\ipk-l2l3-scan> dotnet run -- --interface "Wi-Fi" -s 192.168.1.224/24
Scanning ranges:
192.168.1.0/24 (254 hosts)

192.168.1.1:    arp OK (4CEDF818DFBC), icmpv4 OK
192.168.1.2:    arp FAIL, icmpv4 FAIL
192.168.1.3:    arp FAIL, icmpv4 FAIL
192.168.1.4:    arp FAIL, icmpv4 FAIL
192.168.1.5:    arp FAIL, icmpv4 FAIL
192.168.1.6:    arp FAIL, icmpv4 FAIL
192.168.1.7:    arp FAIL, icmpv4 FAIL
192.168.1.8:    arp FAIL, icmpv4 FAIL
192.168.1.9:    arp FAIL, icmpv4 FAIL
192.168.1.10:   arp FAIL, icmpv4 FAIL
192.168.1.11:   arp FAIL, icmpv4 FAIL
192.168.1.12:   arp FAIL, icmpv4 FAIL
192.168.1.13:   arp FAIL, icmpv4 FAIL
192.168.1.14:   arp FAIL, icmpv4 FAIL
192.168.1.15:   arp FAIL, icmpv4 FAIL
192.168.1.16:   arp FAIL, icmpv4 FAIL
192.168.1.17:   arp FAIL, icmpv4 FAIL
192.168.1.18:   arp FAIL, icmpv4 FAIL
192.168.1.19:   arp FAIL, icmpv4 FAIL
192.168.1.20:   arp FAIL, icmpv4 FAIL
192.168.1.21:   arp FAIL, icmpv4 FAIL
192.168.1.22:   arp FAIL, icmpv4 FAIL
192.168.1.23:   arp FAIL, icmpv4 FAIL
192.168.1.24:   arp FAIL, icmpv4 FAIL
192.168.1.25:   arp FAIL, icmpv4 FAIL
192.168.1.26:   arp FAIL, icmpv4 FAIL
192.168.1.27:   arp FAIL, icmpv4 FAIL
192.168.1.28:   arp FAIL, icmpv4 FAIL
192.168.1.29:   arp FAIL, icmpv4 FAIL
192.168.1.30:   arp FAIL, icmpv4 FAIL
192.168.1.31:   arp FAIL, icmpv4 FAIL
192.168.1.32:   arp FAIL, icmpv4 FAIL
192.168.1.33:   arp FAIL, icmpv4 FAIL
192.168.1.34:   arp FAIL, icmpv4 FAIL
192.168.1.35:   arp FAIL, icmpv4 FAIL
192.168.1.36:   arp FAIL, icmpv4 FAIL
192.168.1.37:   arp FAIL, icmpv4 FAIL
192.168.1.38:   arp FAIL, icmpv4 FAIL

```

```

Administrator: PowerShell
192.168.1.195:  arp FAIL, icmpv4 FAIL
192.168.1.196:  arp FAIL, icmpv4 FAIL
192.168.1.197:  arp FAIL, icmpv4 FAIL
192.168.1.198:  arp FAIL, icmpv4 FAIL
192.168.1.199:  arp OK (641CAEF2833C), icmpv4 OK
192.168.1.200:  arp FAIL, icmpv4 FAIL
192.168.1.201:  arp FAIL, icmpv4 FAIL
192.168.1.202:  arp FAIL, icmpv4 FAIL
192.168.1.203:  arp OK (70C94EAF7D51), icmpv4 OK
192.168.1.204:  arp FAIL, icmpv4 FAIL
192.168.1.205:  arp FAIL, icmpv4 FAIL
192.168.1.206:  arp FAIL, icmpv4 FAIL
192.168.1.207:  arp FAIL, icmpv4 FAIL
192.168.1.208:  arp FAIL, icmpv4 FAIL
192.168.1.209:  arp FAIL, icmpv4 FAIL
192.168.1.210:  arp FAIL, icmpv4 FAIL
192.168.1.211:  arp FAIL, icmpv4 FAIL
192.168.1.212:  arp FAIL, icmpv4 FAIL
192.168.1.213:  arp FAIL, icmpv4 FAIL
192.168.1.214:  arp FAIL, icmpv4 FAIL
192.168.1.215:  arp FAIL, icmpv4 FAIL
192.168.1.216:  arp FAIL, icmpv4 FAIL
192.168.1.217:  arp FAIL, icmpv4 FAIL
192.168.1.218:  arp FAIL, icmpv4 FAIL
192.168.1.219:  arp FAIL, icmpv4 FAIL
192.168.1.220:  arp FAIL, icmpv4 FAIL
192.168.1.221:  arp FAIL, icmpv4 FAIL
192.168.1.222:  arp FAIL, icmpv4 FAIL
192.168.1.223:  arp FAIL, icmpv4 FAIL
192.168.1.224:  arp OK (B827EBA0C077), icmpv4 OK
192.168.1.225:  arp FAIL, icmpv4 FAIL
192.168.1.226:  arp FAIL, icmpv4 FAIL
192.168.1.227:  arp FAIL, icmpv4 FAIL
192.168.1.228:  arp FAIL, icmpv4 FAIL
192.168.1.229:  arp FAIL, icmpv4 FAIL
192.168.1.230:  arp OK (28C63FB182C3), icmpv4 OK
192.168.1.231:  arp FAIL, icmpv4 FAIL
192.168.1.232:  arp FAIL, icmpv4 FAIL
192.168.1.233:  arp FAIL, icmpv4 FAIL
192.168.1.234:  arp FAIL, icmpv4 FAIL
192.168.1.235:  arp FAIL, icmpv4 FAIL
192.168.1.236:  arp FAIL, icmpv4 FAIL

```


Příklad s IPv6

```
Administrator: PowerShell

PS C:\Users\tommi\IPK\IPK_2021\ipk-l2l3-scan> dotnet run -- --interface "Wi-Fi" -s fe80::1743:9e75:745a:438e/123
Scanning ranges:
fe80::1743:9e75:745a:4380/123 (30 hosts)

fe80::1743:9e75:745a:4381: icmpv6 FAIL
fe80::1743:9e75:745a:4382: icmpv6 FAIL
fe80::1743:9e75:745a:4383: icmpv6 FAIL
fe80::1743:9e75:745a:4384: icmpv6 FAIL
fe80::1743:9e75:745a:4385: icmpv6 FAIL
fe80::1743:9e75:745a:4386: icmpv6 FAIL
fe80::1743:9e75:745a:4387: icmpv6 FAIL
fe80::1743:9e75:745a:4388: icmpv6 FAIL
fe80::1743:9e75:745a:4389: icmpv6 FAIL
fe80::1743:9e75:745a:438a: icmpv6 FAIL
fe80::1743:9e75:745a:438b: icmpv6 FAIL
fe80::1743:9e75:745a:438c: icmpv6 FAIL
fe80::1743:9e75:745a:438d: icmpv6 FAIL
fe80::1743:9e75:745a:438e: icmpv6 OK
fe80::1743:9e75:745a:438f: icmpv6 FAIL
fe80::1743:9e75:745a:4390: icmpv6 FAIL
fe80::1743:9e75:745a:4391: icmpv6 FAIL
fe80::1743:9e75:745a:4392: icmpv6 FAIL
fe80::1743:9e75:745a:4393: icmpv6 FAIL
fe80::1743:9e75:745a:4394: icmpv6 FAIL
fe80::1743:9e75:745a:4395: icmpv6 FAIL
fe80::1743:9e75:745a:4396: icmpv6 FAIL
fe80::1743:9e75:745a:4397: icmpv6 FAIL
fe80::1743:9e75:745a:4398: icmpv6 FAIL
fe80::1743:9e75:745a:4399: icmpv6 FAIL
fe80::1743:9e75:745a:439a: icmpv6 FAIL
fe80::1743:9e75:745a:439b: icmpv6 FAIL
fe80::1743:9e75:745a:439c: icmpv6 FAIL
fe80::1743:9e75:745a:439d: icmpv6 FAIL
fe80::1743:9e75:745a:439e: icmpv6 FAIL
```

icmpv6						
No.	Time	Source	Destination	Protocol	Length	Info
13	0.003556	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:438d	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:438d from 70:c9:4e:af:7d:51
14	0.003570	fe80::a56c:ee8d:9499:5c1a	fe80::1743:9e75:745a:438e	ICMPv6	62	Echo (ping) request id=0x0e00, seq=3584, hop limit=128 (reply in 31)
15	0.003903	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:438f	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:438f from 70:c9:4e:af:7d:51
16	0.004038	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4390	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4390 from 70:c9:4e:af:7d:51
17	0.004235	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4391	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4391 from 70:c9:4e:af:7d:51
18	0.004615	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4392	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4392 from 70:c9:4e:af:7d:51
19	0.004675	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4393	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4393 from 70:c9:4e:af:7d:51
20	0.004693	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4394	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4394 from 70:c9:4e:af:7d:51
21	0.004868	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4395	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4395 from 70:c9:4e:af:7d:51
22	0.005006	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4396	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4396 from 70:c9:4e:af:7d:51
23	0.005147	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4397	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4397 from 70:c9:4e:af:7d:51
24	0.005298	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4398	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4398 from 70:c9:4e:af:7d:51
25	0.005421	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4399	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4399 from 70:c9:4e:af:7d:51
26	0.005497	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439a	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439a from 70:c9:4e:af:7d:51
27	0.005644	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439b	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439b from 70:c9:4e:af:7d:51
28	0.005755	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439c	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439c from 70:c9:4e:af:7d:51
29	0.005849	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439d	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439d from 70:c9:4e:af:7d:51
30	0.005940	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439e	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439e from 70:c9:4e:af:7d:51
31	0.103258	fe80::1743:9e75:745a:438e	fe80::a56c:ee8d:9499:5c1a	ICMPv6	62	Echo (ping) reply id=0x0e00, seq=3584, hop limit=64 (request in 14)
32	0.851681	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4381	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4381 from 70:c9:4e:af:7d:51
> Frame 14: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface \Device\NPF_{02C4894D-EB0C-4D5D-9A07-94DA7218CC88}, id 0						
> Ethernet II, Src: LiteonTe_af:7d:51 (70:c9:4e:af:7d:51), Dst: Raspberr_a0:c0:77 (b8:27:eb:a0:c0:77)						
> Internet Protocol Version 6, Src: fe80::a56c:ee8d:9499:5c1a, Dst: fe80::1743:9e75:745a:438e						
Internet Control Message Protocol v6						
Type: Echo (ping) request (128)						
Code: 0						
Checksum: 0x746b [correct]						
[Checksum Status: Good]						
Identifier: 0x0e00						
Sequence: 3584						
[Response In: 31]						

icmpv6						
No.	Time	Source	Destination	Protocol	Length	Info
14	0.003570	fe80::a56c:ee8d:9499:5c1a	fe80::1743:9e75:745a:438e	ICMPv6	62	Echo (ping) request id=0x0e00, seq=3584, hop limit=128 (reply in 31)
15	0.003903	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:438f	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:438f from 70:c9:4e:af:7d:51
16	0.004038	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4390	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4390 from 70:c9:4e:af:7d:51
17	0.004235	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4391	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4391 from 70:c9:4e:af:7d:51
18	0.004615	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4392	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4392 from 70:c9:4e:af:7d:51
19	0.004675	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4393	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4393 from 70:c9:4e:af:7d:51
20	0.004693	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4394	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4394 from 70:c9:4e:af:7d:51
21	0.004868	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4395	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4395 from 70:c9:4e:af:7d:51
22	0.005006	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4396	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4396 from 70:c9:4e:af:7d:51
23	0.005147	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4397	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4397 from 70:c9:4e:af:7d:51
24	0.005298	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4398	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4398 from 70:c9:4e:af:7d:51
25	0.005421	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4399	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4399 from 70:c9:4e:af:7d:51
26	0.005497	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439a	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439a from 70:c9:4e:af:7d:51
27	0.005644	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439b	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439b from 70:c9:4e:af:7d:51
28	0.005755	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439c	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439c from 70:c9:4e:af:7d:51
29	0.005849	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439d	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439d from 70:c9:4e:af:7d:51
30	0.005940	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:439e	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:439e from 70:c9:4e:af:7d:51
31	0.103258	fe80::1743:9e75:745a:438e	fe80::a56c:ee8d:9499:5c1a	ICMPv6	62	Echo (ping) reply id=0x0e00, seq=3584, hop limit=64 (request in 14)
32	0.851681	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4381	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4381 from 70:c9:4e:af:7d:51
33	0.851874	fe80::a56c:ee8d:9499:5c1a	ff02::1:ff5a:4382	ICMPv6	86	Neighbor Solicitation for fe80::1743:9e75:745a:4382 from 70:c9:4e:af:7d:51
> Frame 31: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface \Device\NPF_{02C4894D-EB0C-4D5D-9A07-94DA7218CC88}, id 0						
> Ethernet II, Src: Raspberr_a0:c0:77 (b8:27:eb:a0:c0:77), Dst: LiteonTe_af:7d:51 (70:c9:4e:af:7d:51)						
> Internet Protocol Version 6, Src: fe80::1743:9e75:745a:438e, Dst: fe80::a56c:ee8d:9499:5c1a						
Internet Control Message Protocol v6						
Type: Echo (ping) reply (129)						
Code: 0						
Checksum: 0x736b [correct]						
[Checksum Status: Good]						
Identifier: 0x0e00						
Sequence: 3584						
[Response To: 14]						
[Response Time: 99,688 ms]						

Seznam literatury

[1] Parse the Command Line with System.CommandLine [online]. [cit. 2021-4-25]. Dostupné z: <https://docs.microsoft.com/en-us/archive/msdn-magazine/2019/march/net-parse-the-command-line-with-system-commandline>

[2] SharpPcap.ARP [online]. [cit. 2021-4-25]. Dostupné z: [http://sharppcap.sourceforge.net/html/docs/SharpPcap/ARP.html#M:SharpPcap.ARP.Resolve\(System.Net.IPAddress\)](http://sharppcap.sourceforge.net/html/docs/SharpPcap/ARP.html#M:SharpPcap.ARP.Resolve(System.Net.IPAddress))

[3] Raw Sockets and ICMP [online]. [cit. 2021-4-25]. Dostupné z: https://courses.cs.vt.edu/cs4254/fall04/slides/raw_6.pdf

[4] Internet Control Message Protocol (ICMP) Parameters [online]. [cit. 2021-4-25]. Dostupné z: <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml#icmp-parameters-codes-8>

[5] Internet Control Message Protocol version 6 (ICMPv6) Parameters [online]. [cit. 2021-4-25]. Dostupné z: <https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-parameters-codes-6>

[6] ICMP echo checksum [online]. [cit. 2021-4-25]. Dostupné z: <https://stackoverflow.com/a/20247802>