

Rámec Ethernet II:

Zadání:

Níže jsou uvedeny bajty dvou rámců typu Ethernet II, tak jak byly zachyceny programem Wireshark. Tento program ve svých výpisech neuvádí návěští rámců ani jejich kontrolní součet, takže tato pole v zadání nehlédte.

1. rámec:

```
01 00 5e 00 00 12 00 00 5e 00 01 01 08 00 45 c0
00 28 00 00 00 00 ff 70 17 f0 c0 a8 01 fb e0 00
00 12 21 01 c8 01 00 01 54 55 c0 a8 01 fe 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

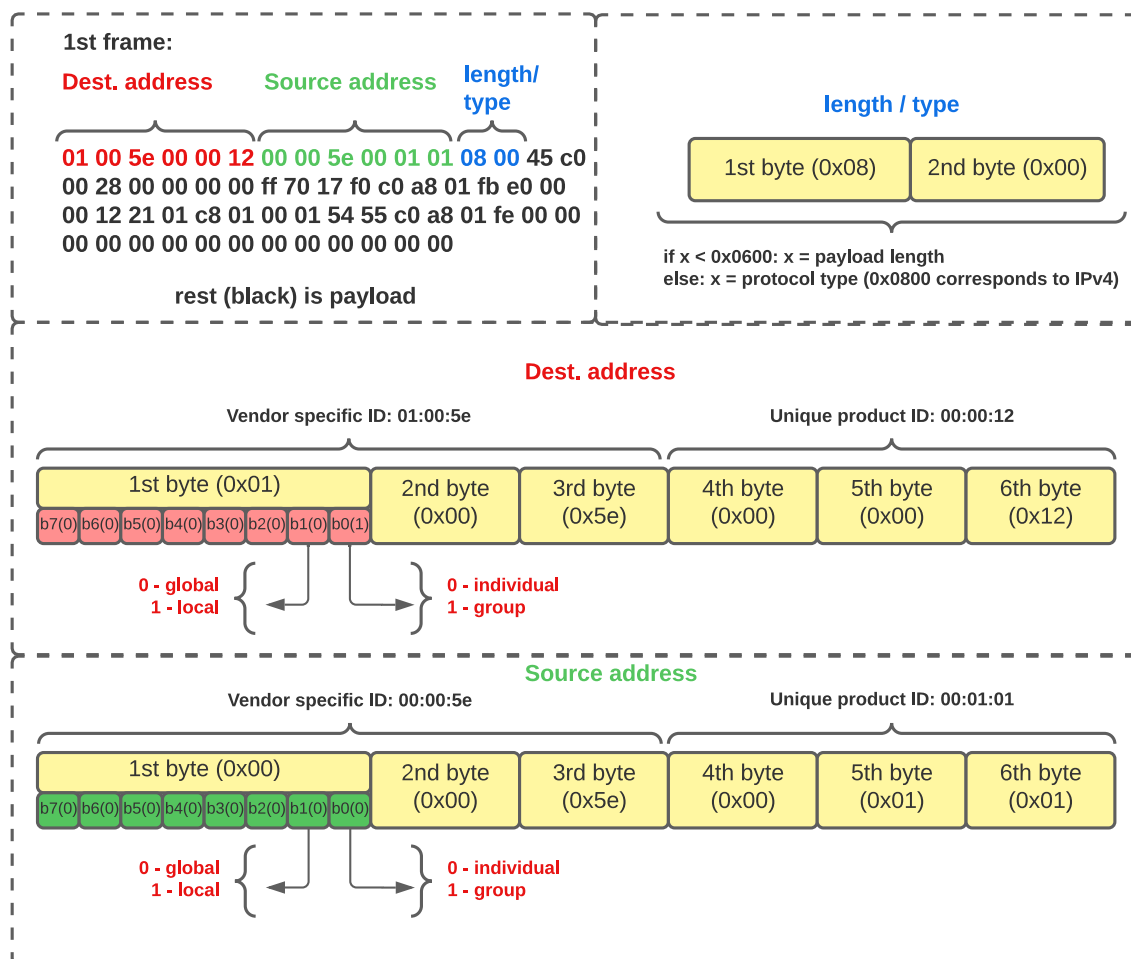
2. rámec:

```
ff ff ff ff ff f0 f3 36 af f4 54 08 06 00 01
08 00 06 04 00 01 00 07 0d af f4 54 18 a6 ac 01
00 00 00 00 00 00 18 a6 ad 9f 06 01 04 00 00 00
00 02 01 00 03 02 00 00 05 01 03 01
```

Z uvedených údajů zjistíte pro oba rámce cílovou a zdrojovou adresu a zjištěné adresy charakterizujte (globální - skupinová - individuální, globální správa adresy - lokální správa adresy, případně uveďte výrobce karty). Dále určete protokol, jehož zpráva je v těle daného rámce přenášena. Potřebné informace lze získat například na následujících stránkách: iana.org a adminsub.net.

Vypracování:

Na následujícím obrázku je znázorněn postup "dekódování 1.rámce". Dekódování obou rámců bylo následně provedeno pomocí python scriptu, který nalzenete v příloze na konci dokumentu. Souhrnné výsledky pro první i druhý rámec a popis jednotlivých adres je na následující straně.



Popis jednotlivých typů adres:

- **Globální (univerzální) vs lokální MAC adresa:**

Globální MAC adresy jsou celosvětově unikátními identifikátory, které zařízením přidělují přímo jejich výrobci. Unikátnost není sice garantována, nicméně kolize dvou globálních MAC adres jsou velice nepravděpodobné. Lokální adresy jsou většinou přidělovány softwarem síťové karty a umožňují tak přepsat globální adresu. Zda je adresa lokální nebo globální lze rozlišit pomocí tzv. U/L bitu (viz obrázek na předchozí stránce).

- **Individuální (Unicast) vs skupinová (multicast) MAC adresa:**

Individuální adresy mohou být jak zdrojové tak i cílové. Individuální adresa je určena pouze pro jedno zařízení s danou MAC adresou. Pokud cílové zařízení obdrží zprávu s individuální MAC adresou, která mu nenáleží, tak zprávu (frame) zignoruje.

Skupinové adresy jsou pouze cílové. Zprávy, které mají skupinovou MAC adresu, jsou určené více zařízením zároveň. Speciálním typem je pak skupinová adresa s hodnotou FF:FF:FF:FF:FF:FF (Broadcast). Na zprávu s touto MAC adresou "slyší" všechny síťové karty. Broadcast se používá například při ARP.

Dekódování prvního rámce:

Destination address:

first byte: 0x01 → 0b00000001 → group and global address.

MAC: 01:00:5E:00:00:12 → Skupinová adresa pravděpodobně od výrobce ICANN, IANA Department

Source address:

first byte: 0x01 → 0b00000001 → individual and global address.

MAC: 00:00:5E:00:01:01 → Vendor specific part: 00:00:5E ICANN, IANA Department

length/type:

0x0800 → DEC: Value is bigger than DEC: 1536 → Ethertype is: IPv4

Dekódování druhého rámce:

Destination address:

first byte: 0xFF → 0b11111111 → Broadcast.

MAC: FF:FF:FF:FF:FF:FF → Broadcast

Source address:

first byte: 0xF0 → 0b11110000 → individual and global address.

MAC: F0:F3:36:AF:F4:54 → Vendor specific part: F0:F3:36 TP-LINK TECHNOLOGIES CO.,LTD.

length/type:

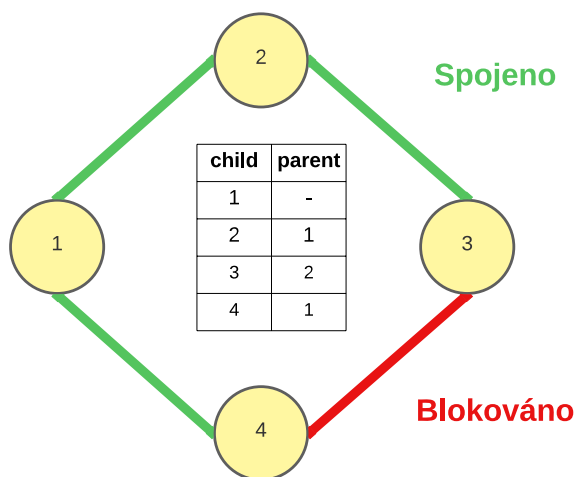
0x0806 → DEC: Value is bigger than DEC: 1536 → Ethertype is: ARP

Protokol STP:

Zadání:

Je dána ethernetová síť sestávající ze čtyř přepínačů s identifikátory 1, 2, 3 a 4. Tyto přepínače jsou propojeny podle níže uvedeného obrázku. Pro zadanou síť určete pomocí algoritmu STP její kostru. Přenesené zprávy запиšte do přehledné tabulky a zjištěnou kostru zakreslete s vyznačením, které porty jsou zapnuty a které blokovány.

Vypracování:



1. zpráva (R,L,ID)					
ID:	1	2	3	4	Nová
1	1,0,1	2,0,2		4,0,4	1,0,1
2	1,0,1	2,0,2	3,0,3		1,1,2
3		2,0,2	3,0,3	4,0,4	2,1,3
4	1,0,1		3,0,3	4,0,4	1,1,4

2. zpráva (R,L,ID)					
ID:	1	2	3	4	Nová
1	1,0,1	1,1,2		1,1,4	1,0,1
2	1,0,1	1,1,2	2,1,3		1,1,2
3		1,1,2	2,1,3	1,1,4	1,2,3
4	1,0,1		2,1,3	1,1,4	1,1,4

Přílohy:

EthernetFrame.py

```
1 from frames import first_frame, second_frame
2 frames = [first_frame, second_frame]
3
4 def parseMAC(MAC_frame):
5     MAC_str = ""
6     for byte,i in zip(MAC_frame,range(len(MAC_frame))):
7         if i == 0:
8             individual_group = byte & 1
9             global_local = (byte & 0b10) >> 1
10            if individual_group == 1:
11                individual_group = "group"
12            elif individual_group == 0:
13                individual_group = "individual"
14            if global_local == 0:
15                global_local = "global"
16            elif global_local == 1:
17                global_local = "local"
18            print("first byte: 0x{byte:02X} -> 0b{bin(byte)[2:].zfill(8)} ->{
individual_group} and {global_local} adress.\nMAC: ",end = "")
19
20            MAC_str += f"{byte:02X}:"
21        print(f"{MAC_str[:-1]} -> Vendor specific part: {MAC_str[0:8]}")
22
23 for fr, i in zip(frames, range(len(frames))):
24
25     print(f"\n\nFRAME {i+1}:")
26     dest_addr = fr[0:6]
27     src_addr = fr[6:6+6]
28     length_type = fr[12: 12+2]
29     payload = fr[14:]
30
31     print("Destination address:")
32     parseMAC(dest_addr)
33
34     print("\nSource address:")
35     parseMAC(src_addr)
36
37     print(f"\nlength/type: 0x{length_type[0]:02X}{length_type[1]:02X} -> DEC: ", end = '')
38     len_type_value = int(f"{length_type[0]:02X}{length_type[1]:02X}",16)
39     if len_type_value < 1536:
40         print(f"Value is smaller than DEC: 1536 -> length of payload is: {len_type_value}
bytes")
41     else:
42         #Common eth_type values:
43         if len_type_value == 0x800:
44             eth_type = "IPv4"
45         elif len_type_value == 0x86DD:
46             eth_type = "IPv6"
47         elif len_type_value == 0x806:
48             eth_type = "ARP"
49         elif len_type_value == 0x811:
50             eth_type = "VLAN"
51
52     print(f"Value is bigger than DEC: 1536 -> Ethertype is: {eth_type}")
```