

Síťové aplikace a správa sítí

# Programování síťové služby

Mapování adresového prostoru IPv6 pomocí OUI

3. listopadu 2012

Autor: Radek Ševčík, [xsevci44@stud.fit.vutbr.cz](mailto:xsevci44@stud.fit.vutbr.cz)  
Fakulta Informačních Technologií  
Vysoké Učení Technické v Brně

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Analýza problému</b>	<b>1</b>
2.1	Identifikátor rozhraní . . . . .	1
2.2	Adresování v IPv6 . . . . .	1
2.3	Podsítě . . . . .	2
2.4	Seznam OUI . . . . .	3
<b>3</b>	<b>Implementace</b>	<b>3</b>
<b>4</b>	<b>Testování</b>	<b>4</b>
<b>5</b>	<b>Závěr</b>	<b>4</b>

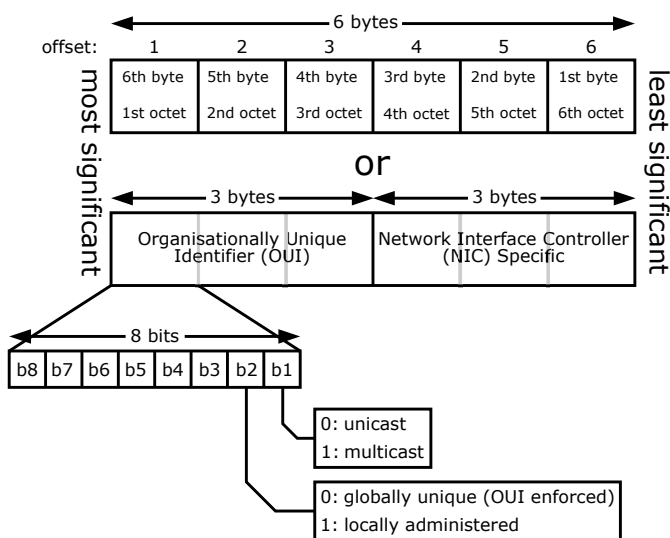
# 1 Úvod

Cílem projektu je vyhledat v adresovém prostoru IPv6 síťové rozhraní s přiřazenými modifikovanými EUI-64 adresami. Tyto EUI-64 adresy budou tvořeny ze seznamu OUI, které si program načte ze souboru. Vyhledávání bude probíhat za použití zpráv Echo Request protokolu ICMPv6.

## 2 Analýza problému

### 2.1 Identifikátor rozhraní

Každé síťové rozhraní musí mít jedinečnou unicast adresu v rámci podsítě. Tato adresa je 64bitová a nazývá se identifikátor rozhraní. Identifikátor rozhraní používá IPv6 modifikovaný EUI-64 formát adresy. Ten se vytvoří inverzí bitu U/L adresy IEEE EUI-64. Tradiční rozhraní používá adresu založenou na IEEE 802, jinak známou jako MAC-48 (obr. 1). EUI-64 má stejný formát jako MAC-48, rozdíl je pouze v části NIC, která je 40bitová (obr. 2). Adresa MAC-48 se namapuje na EUI-64 vložení hodnoty 0xFFFE mezi OUI a NIC. Celý proces je na obr. 3.



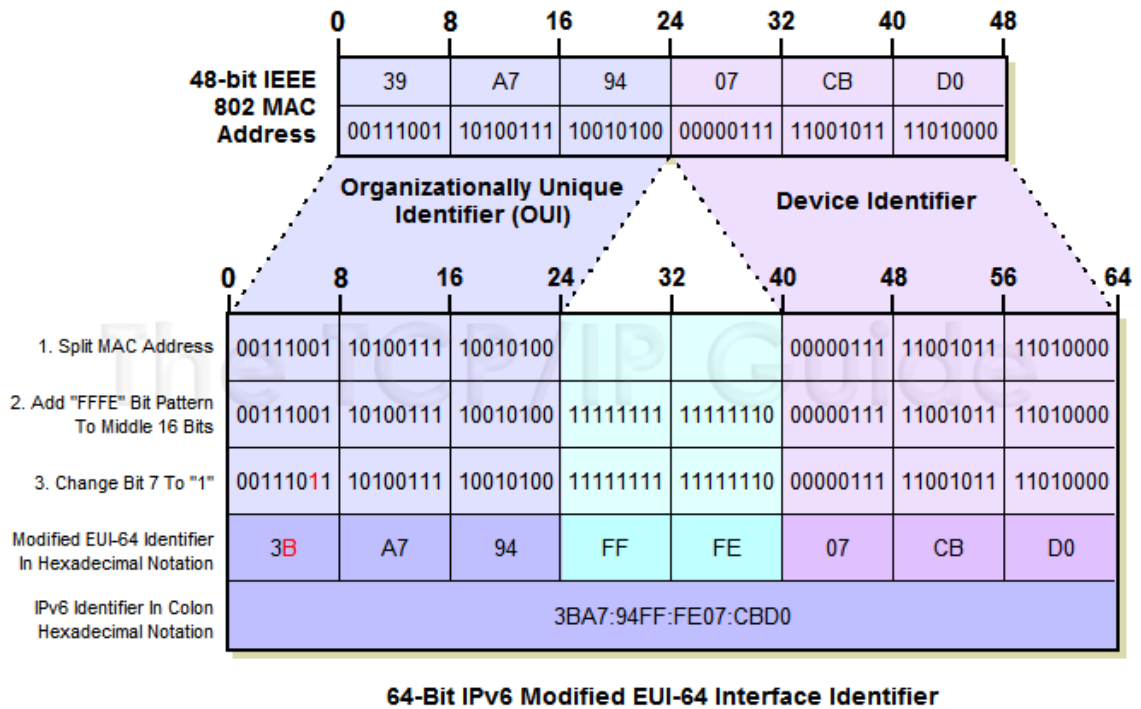
Obrázek 1: MAC-48, převzato z [7].



Obrázek 2: EUI-64, převzato z [8].

### 2.2 Adresování v IPv6

Podle [4] je vyžadováno, aby každá globální unicast IPv6 adresa, mimo těch s prefixem 000b, obsahovala 64bitový identifikátor rozhraní tvořený v modifikovaném EUI-64 formátu. Mimo to se EUI-64 používá i v privátních sítích a linkových adresách.



Obrázek 3: Konverze MAC-48 na IPv6 modifikovaný EUI-64, převzato z [9].

48 (or more)	16 (or fewer)	64
routing prefix	subnet id	interface id

Tabulka 1: Formát globální unicast IPv6 adresy.

7	1	40	16	64
fc00	L	global id	subnet id	interface id

Tabulka 2: Formát lokální unicast IPv6 adresy.

10	54	64
fe80	0	interface id

Tabulka 3: Formát link-local unicast IPv6 adresy.

## 2.3 Podsítě

Jedním z požadavků na program je, aby uměl proskenovat všechny podsítě ve zvoleném adresovém prostoru. Adresový prostor je specifikován ve formátu prefix/prefixlen. Po oštrění této adresy můžeme použít následující pseudokód.

```
(* prefix je unicast adresa *)
(* prefixlen je číslo do 64 *)
procedure process(prefix: in6_addr; prefixlen: integer);
var
  subnet: in6_addr;
  subnet_max, net_mask: uint64;
```

```

    i: integer;
begin
    subnet_max := (1 shl (64 - prefixlen)) - 1;
    net_mask := not subnet_max;

    (* ošetříme adresu sítě *)
    (* dle RFC3587 je posledních 64bitů interface_id *)
    with prefix do
    begin
        s6_addr32[0] := s6_addr32[0] and htobe32(net_mask shr 32); (* MSB *)
        s6_addr32[1] := s6_addr32[1] and htobe32(net_mask and UINT32_MAX); (* LSB *)
        s6_addr32[2] := 0;
        s6_addr32[3] := 0;
    end;

    (* vygenerujeme jednotlivé podsítě v rozsahu <0..subnet_max> *)
    for i := 0 to subnet_max do
    begin
        with subnet do
        begin
            s6_addr32[0] := prefix.s6_addr32[0] or htobe32(i shr 32); (* MSB *)
            s6_addr32[1] := prefix.s6_addr32[1] or htobe32(i and UINT32_MAX); (* LSB *)
            s6_addr32[2] := 0;
            s6_addr32[3] := 0;
        end;

        ...

    end;
end;

```

## 2.4 Seznam OUI

Dle zadání může seznam OUI obsahovat také textový řetězec, popisující název výrobce síťového rozhraní. Tento seznam můžeme popsat regulárním výrazem v PCRE, kde \1 je OUI a \2 je název výrobce.

```
^([[:xdigit:]]{2}:([[:xdigit:]]{2}:([[:xdigit:]]{2}))[:space:]]+(.*)$
```

## 3 Implementace

Databáze OUI je naimplementována jako lineární seznam.

Procedury pro generování podsítí a identifikátorů rozhraní pro zachování své jednoduchosti používají callback systém.

Je vytvořen jediný soket obsluhující vstup i výstup. Čekání na události zajišťuje funkce poll. Pro zápis do soketu je upřednostněna funkce `sendmsg` oproti `sendto`, protože je více low-level. Jelikož pracujeme s ICMPv6, je možné nastavit filtr pro přijetí zpráv pouze typu

ICMP6\_ECHO\_REPLY. Přesto však musíme kontrolovat nejen adresu odesílatele, ale i `icmp6_id`, který je doporučen nastavit na `pid` procesu. Systém totiž přeposílá veškeré ICMP6\_ECHO\_REPLY zprávy.

Aplikace vytváří samostatné vlákno pro obsluhu soketu. Hlavní vlákno generuje adresy a ukládá je do fronty. Jakmile je fronta zaplněná, vlákno se uspí. Druhé vlákno simultánně z fronty adresy odebírá a odesílá ICMP zprávy. Když se fronta vyprázdní, probudí se hlavní vlákno. Pro synchronizaci jsou využity `mutexes` a `condition variables` knihovny `pthread`.

## 4 Testování

Během vývoje došlo k několika optimalizacím. Testy probíhaly ve VirtualBoxu se systémem FreeBSD. Byla prohledávána jedna podsít'. Příklad spuštění je uveden v souboru `Readme`.

V následujícím měření měl virtuální stroj přiřazené pouze 1 jádro fyzického CPU a aplikace využívala 100% času procesoru.

```
real    160m45.658s
user      0m12.711s
sys       69m21.798s
```

```
real    125m16.801s
user      0m 4.722s
sys      51m26.185s
```

```
real    109m49.439s
user      0m 4.027s
sys      46m25.887s
```

Při poskytnutí většího výkonu byly nameřeny tyto hodnoty.

```
real    51m 6.630s
user      0m36.667s
sys      51m38.404s
```

## 5 Závěr

Tento projekt mi přinesl užitečné znalosti o praktické realizaci IPv6 aplikací a způsobu adresování. Mimo jiné také to, že jedna nejmenovaná společnost ve výchozím nastavení svého OS nepoužívá jako link-local adresu modifikovaný EUI-64 formát jako většina ostatních, nýbrž z důvodů bezpečnosti používá adresu dle [6] RFC 4941.

## Reference

- [1] Deering, S. and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, December 1998. <<http://www.ietf.org/rfc/rfc2460.txt>>
- [2] Hinden, R. and S. Deering, *IP Version 6 Addressing Architecture*, RFC 4291, February 2006. <<http://www.ietf.org/rfc/rfc4291.txt>>
- [3] Hinden, R. and B. Haberman, *Unique Local IPv6 Unicast Addresses*, RFC 4193, October 2005. <<http://www.ietf.org/rfc/rfc4193.txt>>
- [4] Hinden, R., Deering, S., and E. Nordmark, *IPv6 Global Unicast Address Format*, RFC 3587, August 2003. <<http://www.ietf.org/rfc/rfc3587.txt>>
- [5] Conta, A., Deering, S., and M. Gupta, Ed., *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 4443, March 2006. <<http://www.ietf.org/rfc/rfc4443.txt>>
- [6] Narten, T., Draves, R., and S. Krishnan, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, RFC 4941, September 2007. <<http://www.ietf.org/rfc/rfc4941.txt>>
- [7] Wikipedia contributors, *MAC address*, Wikipedia, The Free Encyclopedia, Retrieved 1 Nov. 2012. <<http://en.wikipedia.org/wiki/MAC-48>>
- [8] Microsoft, *IPv6 interface identifiers*, Microsoft TechNet, 21 Jan. 2005. <<http://technet.microsoft.com/en-us/library/cc736439.aspx>>
- [9] Charles M. Kozierok, *IPv6 Interface Identifiers and Physical Address Mapping*, The TCP/IP Guide, Retrieved 1. Nov. 2012. <[http://www.tcpipguide.com/free/t\\_IPv6InterfaceIdentifiersandPhysicalAddressMapping.htm](http://www.tcpipguide.com/free/t_IPv6InterfaceIdentifiersandPhysicalAddressMapping.htm)>
- [10] Donald E. Knuth, *Umění programování 2., Seminumerické algoritmy*, Brno Computer Press, 2010. Vyd. 1. xi, 763 s., ISBN 978-80-251-2898-5