

# Segment Routing v Birdu

**Matyáš Kroupa**

Sitola, FI MUNI

22. 1. 2026

# Kdo jsem

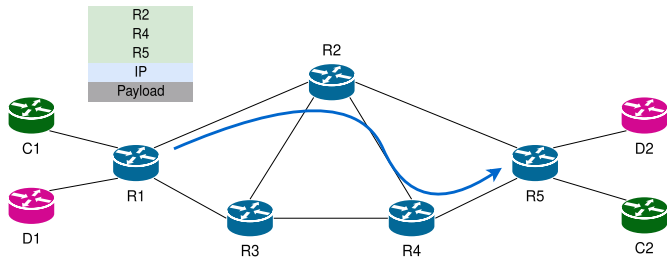
- Doktorand na Fakultě informatiky Masarykovy univerzity
- Laboratoř síťových technologií Sitola

# Motivace

- Směrovací tabulky ve velkých sítích neškálují
- Řešeno pomocí MPLS
- Spousta nových protokolů
- Velmi stavový traffic engineering
- Nefunguje napříč administrativními doménami

# Segment Routing

- Forma source routing
- Vstupní směrovač do paketu zapíše seznam segmentů
- Segmenty jsou brány jako instrukce
- Síť provede daný program

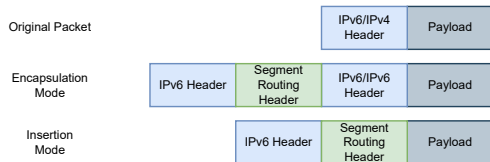


# Segment routing

- Snaha co nejvíce využít link-state
- Buď OSPF, nebo IS-IS jako control plane
- MPLS nebo IPv6 jako data plane

# SRv6

- Používá IPv6 jako data plane
- IPv6 pakety přenášejí payload skrze síť
- Segmenty jsou IPv6 adresy
- Segmenty jsou zakódovány do rozšiřující hlavičky - Segment Routing Header
- Segmenty jednoho uzlu se sdružují do lokátorů - IPv6 prefixy
- Základní typy segmentů:
  - End: Jdi skrze uzel
  - End.X: Jdi skrze linku



# SRv6 do Birdu

- Praktická část mojí diplomové práce
- Data plane: IPv6
- Control plane: OSPFv3

# Postup

- Rozšířená LSA
- SRv6 rozšíření
- Abstrakce pro počítání SPF
- Podklady pro Flex-algo

# Rozšířená LSA

- OSPF má fixní tvar LSA
- Původní myšlenka: nové LSA na nové typy informací
- Nevhodné pro nové atributy linek
- Řešení: rozšířená LSA používající TLV
- Nejsou zpětně kompatibilní

# TLV

- Type-Length-Value
- Value má typicky fixní a proměnnou část
- Sub-tlv lze libovolně zanořovat
- Length pokrývá i variabilní část -> jednoduše se píše validace

```
+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      |
+---+---+---+---+---+---+---+---+---+---+
|                  Fix Value                  |
+---+---+---+---+---+---+---+---+---+---+
|  Fix Value  |  Fix Value  |
+---+---+---+---+---+---+---+---+---+---+
|                  Sub-TLVs                  |
+---+---+---+---+---+---+---+---+---+---+
```

# Endianita

- Sítě jsou big-endian
- Na little-endian je třeba prohodit pořadí oktetů
- Ale jen v rámci 32-bitových slov

```
struct ospf_tlv_intra_area_prefix {  
    u32 metric;  
#ifdef CPU_BIG_ENDIAN  
    u8 prefix_len;  
    u8 prefix_options;  
    u16 padding;  
#else  
    u16 padding;  
    u8 prefix_options;  
    u8 prefix_len;  
#endif  
    u32 data[]; /* Prefix and sub_tlvs */  
};
```

# Bitová pole

- Některé hodnoty v TLV jsou 24-bitové
- Kombinování hodnot v jednom slově
- Zdánlivé řešení: bitová pole
- Nezaručují endiianitu

```
struct ospf_lsa_e_link /* E-Link-LSA */
{
#ifdef CPU_BIG_ENDIAN
    u32 rtr_priority : 8, options : 24;
#else
    u32 options : 24, rtr_priority : 8;
#endif
    u32 tlvs[];
};
```

# Implementace

- Nové datové struktury pro LSA a TLV
- Validace
- Zpracování přijatých LSA
- Generování LSA
  - Generování SIDů implicitně i explicitně
- Možnost konfigurace
- Výpis na CLI

## Příklad konfigurace

```
protocol ospf v3 ospf6 {  
    extended lsas full;  
    srv6;  
    ipv6 {  
        export all;  
        import all;  
    };  
    area 0 {  
        interface -"ens3", -"ens10", "ens*" {  
            type ptp;  
        };  
        interface "ens10";  
        srv6 locator fc00:0:11::/64 {  
            sid ::1:0:2 end.dt6 vrftable 42;  
        };  
    };  
}
```

## Příklad výpisu show

```
2001:db8:11::/64      unicast [...] * I (150/10) [10.0.0.11]
    dev ens10
fc00:0:11::4/128      [...] * srv6local behavior End.X nh6 ens5
fc00:0:11::1/128      [...] * srv6local behavior End
fc00:0:11::2/128      [...] * srv6local behavior End.X nh6 ens4
fc00:0:11::5/128      [...] * srv6local behavior End.DX6 nh6 ens5
fc00:0:11::3/128      [...] * srv6local behavior End.DX6 nh6 ens4
fc00:0:11::1:0:2/128  [...] * srv6local behavior End.DT6 vrftable 42
fc00:0:5::/64         unicast [...] * I (150/30) [10.0.0.5]
    via fe80::e8b:41ff:fe64:1 on ens4
fc00:0:4::/64         unicast [...] * I (150/20) [10.0.0.4]
    via fe80::e8b:41ff:fe64:1 on ens4
fc00:0:1::/64         unicast [...] * I (150/10) [10.0.0.1]
    via fe80::e81:9ff:fed6:2 on ens5
fc00:0:2::/64         unicast [...] * I (150/20) [10.0.0.2]
```

# Testování

- GNS3
- Jednoduchá topologie
- Virtuální stroje s Fedorou
- Kompilace, instalace a konfigurace přes Ansible
- Uzavřená IPv6 síť
- Natovaná IPv4 do Internetu

# Refactoring SPF

- Flexible algorithm bude potřebovat více instancí SPF
- SPF z pohledu jiného směrovače
- Napříč celým kódem

```
-ospf_rt_spf(struct ospf_proto *p)  
+ospf_rt_spf(struct spf_computation *spf)
```

## Pár čísel na závěr

- Cca 3200 řádků nového kódu
- Třetina kódu napsána v autobuse

# Poděkování

Marie Matějce a Ondrovi Zajíčkovi za uvítání do projektu a veškerou pomoc