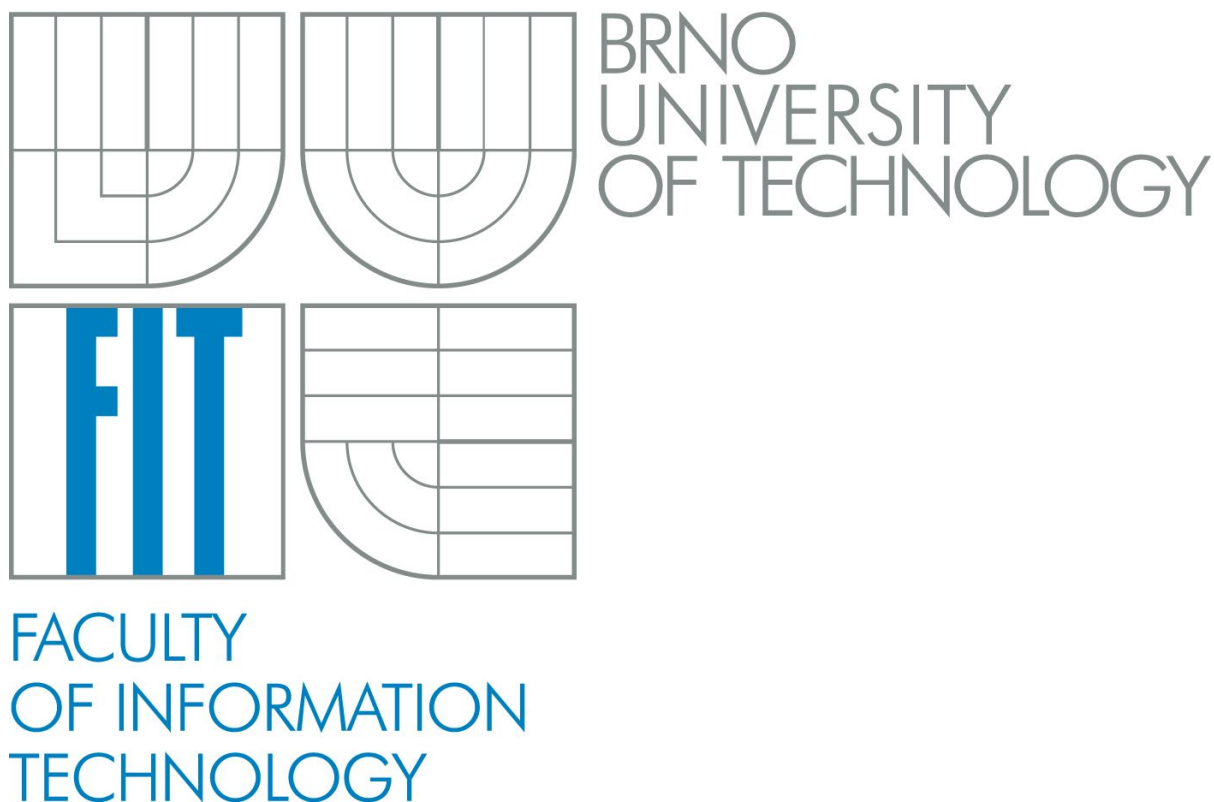


# Projekt 6in4

ISA - Síťové aplikace a správa sítí



## 1. Uvedenie do problematiky

6in4 je mechanizmus na migrovanie z internetoveho protokolu verzie 4 (IPv4) do IPv6. 6in4 používa tunelovanie, aby zapúzdрил IPv6 traffic na predem nakonfigurované IPv4 spojenie. 6in4 traffic je posielaná v IPv4 packetoch, ktoré používajú protokol číslo 41. Tento protokol je vyhradený na zapúzdrowanie IPv6 packetoch. V 6in4 je IPv6 packet ihneď za IPv4 headerom. Znamená to, že na každý IPv6 packet je nadbytočných 20 bajtov (IPv4 header veľkosť). Maximálna veľkosť IPv4 paketu je 1500 bajtov, teda 1480 pri zapuzdrenom IPv6 pakete, aby mohol byť poslaný bez fragmentácie.

## 2. Návrh aplikace

Program je rozdelený na **3** hlavné časti:

Tunnely bežia vo vlastných threads, aby mohol zároveň prijímať a odosielať pakety.

- **lan** interface smeruje IPv6 sieť
- **wan** interface smeruje na IPv4 sieť
- **remote** adresa je opačná adresa tunela

### tunnel\_6in4

V tejto časti aplikácie sa počúva na **lan** interface a po príchode packetu sa prepošle IPv6 packet v IPv4 packete na **remote** adresu.

### tunnel\_4in6

V tejto časti aplikácie sa počúva na **wan** interface a po príchode packetu sa rozbalí IPv6 packet v IPv4 packete na **lan** interface.

### main

Táto časť aplikácie má na starosti spracovať parametry, vytvoriť threads a v nich zavolať funkcie na fungovanie programu, ukončiť threads (pri **SIGINT**) a uklidiť za sebou zdroje.

### 3. Popis implementácie

Rozhodol som sa maximálne využiť **kernel**, aby som nemusel ručne modifikovať **IP** hlavičky.

**tunnel\_6in4** funkcia obsahuje:

- **send4\_init()** - inicializuje socket a nastaví potrebné parametry pre IPv4 headers
- **listen6\_init()** - inicializuje socket a nastaví potrebné parametry pre IPv6 headers
- **listen6\_start()** - zahajuje listening na všetky prijaté IPv6 pakety (IPPROTO\_RAW)
  - **send4\_start()** - od **listen6\_start()** dostane IPv6 packet, ktorý zabalí do IPv4 paketu a prepošle ho druhú stranu tunelu - **remote**
- **send4\_exit()** a **listen6\_start()** - zatvárajú file descriptors, sú volané v prípade signálu **SIGINT**

**tunnel\_4in6** funkcia obsahuje:

- **listen4\_init()** - inicializuje socket a nastaví potrebné parametry pre IPv4 headers
- **send6\_init()** - inicializuje socket a nastaví potrebné parametry pre IPv6 headers
- **listen4\_start()** - zahajuje listening na všetky prijaté IPv6 pakety na interface **lan**
  - **send6\_start()** - od **listen4\_start()** dostane IPv4 packet, ktorý rozbalí a prepošle ho na interface **lan**
- **send6\_exit()** a **listen4\_start()** - zatvárajú file descriptors, sú volané v prípade signálu **SIGINT**

### 4. Návod k použitiu

Podrobné vysvetlenie aj príklad konfigurácie a výstupu je v súbore **README.html** (aby bol kód čitateľný a kopírovateľný)

## 5. Referencie

<https://en.wikipedia.org/wiki/6in4>

<https://podpora.nic.cz/page/661/ipv6-tunelovaci-mechanismy---instalace-a-nastaveni/>