Služby počítačových sítí: HTTP, IPv6

Jan Kasprzak
<kas@fi.muni.cz>
https://www.fi.muni.cz/~kas/

Kapitola 1

Protokol HTTP

Protokol HTTP

- HyperText Transfer Protocol
- Přenos dokumentů po síti
- Vyhrazený port – `80/tcp`

Verze protokolu HTTP

- Původní verze – označovaná též 0.9
- Verze 1.0 – RFC 1945 (využití RFC 822 a MIME)
- Verze 1.1 – RFC 2068 (přesnější definice doby platnosti dokumentu a další rozšíření)
- Verze 2 – RFC 7540, 2015 (multiplexování požadavků)
- Verze 3 – RFC 9114, 2022 (nad QUIC)

Protokol HTTP

- HyperText Transfer Protocol
- Přenos dokumentů po síti
- Vyhrazený port – `80/tcp`

Verze protokolu HTTP

- Původní verze – označovaná též 0.9
- Verze 1.0 – RFC 1945 (využití RFC 822 a MIME)
- Verze 1.1 – RFC 2068 (přesnější definice doby platnosti dokumentu a další rozšíření)
- Verze 2 – RFC 7540, 2015 (multiplexování požadavků)
- Verze 3 – RFC 9114, 2022 (nad QUIC)

Příklad: Komunikace pomocí HTTP

```
$ telnet localhost http
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /
<HTML>
<HEAD>
...
```

Protokol, dnes označovaný jako HTTP/0.9

Příklad: Komunikace pomocí HTTP

```
$ telnet localhost http
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /
<HTML>
<HEAD>
...
```

Protokol, dnes označovaný jako HTTP/0.9

Příklad: Komunikace pomocí HTTP 1.0

```
$ telnet www.fi.muni.cz http
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0
Host: localhost
User-Agent: Telnet/1.0.1 X11

HTTP/1.0 200 OK
Date: Tue, 05 Oct 2010 23:14:11 GMT
Last-Modified: Tue, 05 Oct 2010 22:46:25 GMT
Content-Type: text/html; charset=UTF-8
Content-Language: cs

<HTML> ...
```

Metoda POST

- Přenos dat od klienta k serveru
- Obsah formuláře, upload souborů, atd.
- Data se posílají za hlavičkami
- Odpovědí je dokument

Příklad: Metoda POST

```
$ telnet localhost http
...
POST /cgi-bin/skript.cgi HTTP/1.0
Content-Length: 20

1234567890
abcdefghijklmnop
^D
HTTP/1.0 200 OK
....
Content-type: text/plain

1234567890
abcdefgh
Connection closed by foreign host.
```

Formát HTTP požadavku a odpovědi

- RFC 822 plus MIME
- Příkaz – metoda, stav
- Hlavička
  - řádky tvaru *klíč*: *parametry*
  - pokračovací řádky – začínají bílým znakem
- Konec hlaviček – prázdný řádek
- Zbytek je tělo zprávy

Formát HTTP požadavku a odpovědi

- RFC 822 plus MIME
- Příkaz – metoda, stav
- Hlavička
  - řádky tvaru *klíč*: *parametry*
  - pokračovací řádky – začínají bílým znakem
- Konec hlaviček – prázdný řádek
- Zbytek je tělo zprávy

Formát HTTP požadavku a odpovědi

- RFC 822 plus MIME
- Příkaz – metoda, stav
- Hlavička
  - řádky tvaru *klíč*: *parametry*
  - pokračovací řádky – začínají bílým znakem
- Konec hlaviček – prázdný řádek
- Zbytek je tělo zprávy

Formát HTTP požadavku a odpovědi

- RFC 822 plus MIME
- Příkaz – metoda, stav
- Hlavička
  - řádky tvaru *klíč*: *parametry*
  - pokračovací řádky – začínají bílým znakem
- Konec hlaviček – prázdný řádek
- Zbytek je tělo zprávy

Formát HTTP požadavku a odpovědi

- RFC 822 plus MIME
- Příkaz – metoda, stav
- Hlavička
  - řádky tvaru *klíč*: *parametry*
  - pokračovací řádky – začínají bílým znakem
- Konec hlaviček – prázdný řádek
- Zbytek je tělo zprávy

Hlavičky pro HTTP 1.0

`Accept: text/html, image/*, */*` – priorita MIME typů a podtypů, které klient je ochoten zpracovávat.

`Accept-Charset: iso-8859-2, utf-8, us-ascii` – seznam/priorita znakových sad, podporovaných klientem.

`Accept-Encoding: 8bit, base64` – přenosová kódování podporovaná klientem.

`Accept-Language: cs, en, *` – priorita jazyků pro textové dokumenty.

Hlavičky pro HTTP 1.0

`User-Agent: Mozilla/1.1 (X11, Linux)` – identifikace typu a verze klienta.

`From: kas@fi.muni.cz` – adresa uživatele.

`Referer: http://www.linux.cz/` – odkud jsme se na tuto stránku dostali.

`Content-Type: image/jpeg` – význam dat (MIME typ). Problém u MS Windows (MIME typ versus přípona).

`Content-Language: cs` – jazyk dokumentu.

Hlavičky pro HTTP 1.0

`If-Modified-Since: Sat, 29 Oct 1994 ...` – poslat, jen pokud byl dokument od této doby modifikován.

`Date: Sat, 29 Oct 1994 19:43:31 GMT` – současné datum.

`Expires: Sat, 29 Oct 1994 19:43:31 GMT` – datum vypršení platnosti dokumentu.

`Last-Modified: Sat, 29 Oct 1994 18:15:24 GMT` – datum modifikace dokumentu.

Hlavičky pro HTTP 1.0

`MIME-Version: 1.0` – používaná verze standardu MIME.

`Location: http://www.fi.muni.cz/tech/` – kanonická adresa dokumentu.

`Pragma: no-cache` – řízení vyrovnávací paměti.

`Server: Apache/2.0.16` – verze a typ serveru.

`Retry-After: 60` – kdy bude dokument přístupný.

HTTP – úspěšné návratové kódy

200 OK
201 Created
202 Accepted
203 Non-Authoritative Information
204 No Content
205 Reset Content
206 Partial Content

HTTP – mezilehlé návratové kódy

300 Multiple Choices
301 Moved Permanently
302 Moved Temporarily
303 See Other
304 Not Modified
305 Use Proxy

HTTP – chyba klienta

400 Bad Request
401 Unauthorized
402 Payment Required
403 Forbidden
404 Not Found
405 Method Not Allowed
406 Not Acceptable
407 Proxy Authentication Required

HTTP – chyba klienta

408 Request Time-Out
409 Conflict
410 Gone
411 Length Required
412 Precondition Failed
413 Request Entity Too Large
414 Request-URL Too Large
415 Unsupported Media Type

HTTP – chyba serveru

500 Server Error
501 Not Implemented
502 Bad Gateway
503 Out of Resources
504 Gateway Time-Out
505 HTTP Version not supported

Základní autentizace

- Autentizace klienta
- Realm – část WWW serveru pro kterou je autentizovaný přístup.
- Username – jméno uživatele.
- Password – heslo. Šifruje se pomocí `crypt(3)`.

Příklad: Konfigurace v Apache

```
AuthName ”Tajne stranky”
AuthType Basic
AuthUserFile .../htpasswd
require valid-user
```

Hesla pro Apache

Příklad: Soubor .htpasswd

```
luke:msfjGRNhRo9as
```

Vytvoříme např. příkazem `htpasswd -n luke c3po`

Základní autentizace – komunikace

- První přístup – kód 401 a důvod:
  `WWW-Authenticate: Basic realm="Tajne"`
- Přístup se jménem a heslem: řetězec *login:heslo* kódovaný Base64:
  `Authorization: Basic bHVrZTpjM3Bv`

Základní autentizace – komunikace

- První přístup – kód 401 a důvod:
  `WWW-Authenticate: Basic realm="Tajne"`
- Přístup se jménem a heslem: řetězec *login:heslo* kódovaný Base64:
  `Authorization: Basic bHVrZTpjM3Bv`

Cookies

- Protokol HTTP – bezstavový.
- Doplnění stavovosti: úprava URL, cookies
- Cookies: server nastavuje, klient zopakuje
- Obsah cookie: název, hodnota (pro klienta neprůhledná)
- Parametry cookie: životnost, rozsah platnosti

Cookies

- Protokol HTTP – bezstavový.
- Doplnění stavovosti: úprava URL, cookies
- Cookies: server nastavuje, klient zopakuje
- Obsah cookie: název, hodnota (pro klienta neprůhledná)
- Parametry cookie: životnost, rozsah platnosti

Cookies

- Protokol HTTP – bezstavový.
- Doplnění stavovosti: úprava URL, cookies
- Cookies: server nastavuje, klient zopakuje
- Obsah cookie: název, hodnota (pro klienta neprůhledná)
- Parametry cookie: životnost, rozsah platnosti

Cookies

- Protokol HTTP – bezstavový.
- Doplnění stavovosti: úprava URL, cookies
- Cookies: server nastavuje, klient zopakuje
- Obsah cookie: název, hodnota (pro klienta neprůhledná)
- Parametry cookie: životnost, rozsah platnosti

Cookies

- Protokol HTTP – bezstavový.
- Doplnění stavovosti: úprava URL, cookies
- Cookies: server nastavuje, klient zopakuje
- Obsah cookie: název, hodnota (pro klienta neprůhledná)
- Parametry cookie: životnost, rozsah platnosti

Příklad: Použití cookies

```
HTTP/1.0 200 OK
...
Set-Cookie: barvapozadi=modra
Set-Cookie: session=a9d822150891fc1b47;
   Expires=Wed, 17 Oct 2018 13:49:28

GET /neco.html HTTP/1.0
...
Cookie: session=a9d822150891fc1b47;
   barvapozadi=modra
```

Cookies: pro a proti

- Stavovost HTTP
- Globální pro celý prohlížeč!
  - Mezistav aplikace raději držet v parametrech GET/POST.
- Riziko pro soukromí

Cookies: pro a proti

- Stavovost HTTP
- Globální pro celý prohlížeč!
  - Mezistav aplikace raději držet v parametrech GET/POST.
- Riziko pro soukromí

Cookies: pro a proti

- Stavovost HTTP
- Globální pro celý prohlížeč!
  - Mezistav aplikace raději držet v parametrech GET/POST.
- Riziko pro soukromí

HTTPS a SSL/TLS

- Problém – komunikace je odposlechnutelná
- Řešení – šifrování provozu HTTP
- SSL – secure sockets layer
- TLS – transport layer security
- Metoda šifrování proudového spojení
- HTTPS – HTTP nad SSL/TLS

Otázka:

Co je základním problémem šifrování v prostředí, kde se uživatelé navzájem neznají?

HTTPS a SSL/TLS

- Problém – komunikace je odposlechnutelná
- Řešení – šifrování provozu HTTP
- SSL – secure sockets layer
- TLS – transport layer security
- Metoda šifrování proudového spojení
- HTTPS – HTTP nad SSL/TLS

Otázka:

Co je základním problémem šifrování v prostředí, kde se uživatelé navzájem neznají?

HTTPS a SSL/TLS

- Problém – komunikace je odposlechnutelná
- Řešení – šifrování provozu HTTP
- SSL – secure sockets layer
- TLS – transport layer security
- Metoda šifrování proudového spojení
- HTTPS – HTTP nad SSL/TLS

Otázka:

Co je základním problémem šifrování v prostředí, kde se uživatelé navzájem neznají?

Problém: – jak se domluvit na šifrovacím klíči?

SSL/TLS Certifikáty

- Server – pár RSA nebo EC klíčů (tajný, veřejný). Slouží pro výměnu klíče pro symetrickou šifru.
- Session cache
- Klient – volitelně také pár klíčů, jen pro autentizaci.
- Problém – jak zjistit, že mluvím s tím správným serverem?
- Stávající řešení – certifikát
  - podpis veřejného klíče a jména serveru certifikační autoritou
  - Public Key Infrastructure (PKI)

Příklad: Komunikace přes SSL/TLS

```
openssl s_client -connect \
        www.fi.muni.cz:https
```

SSL/TLS Certifikáty

- Server – pár RSA nebo EC klíčů (tajný, veřejný). Slouží pro výměnu klíče pro symetrickou šifru.
- Session cache
- Klient – volitelně také pár klíčů, jen pro autentizaci.
- Problém – jak zjistit, že mluvím s tím správným serverem?
- Stávající řešení – certifikát
  - podpis veřejného klíče a jména serveru certifikační autoritou
  - Public Key Infrastructure (PKI)

Příklad: Komunikace přes SSL/TLS

```
openssl s_client -connect \
        www.fi.muni.cz:https
```

Problémy PKI

- Seznam kořenových autorit
  - Kdo rozhodne, které tam zařadit?
- Každá CA může vydat certifikát pro kohokoliv
  - Kompromitace jedné CA kompromituje vše
- Revokační seznamy
  - Co když ještě nemám konektivitu?
- Možné řešení: key pinning
- Možné řešení: DNSSEC + DANE

Problémy PKI

- Seznam kořenových autorit
  - Kdo rozhodne, které tam zařadit?
- Každá CA může vydat certifikát pro kohokoliv
  - Kompromitace jedné CA kompromituje vše
- Revokační seznamy
  - Co když ještě nemám konektivitu?
- Možné řešení: key pinning
- Možné řešení: DNSSEC + DANE

Problémy PKI

- Seznam kořenových autorit
  - Kdo rozhodne, které tam zařadit?
- Každá CA může vydat certifikát pro kohokoliv
  - Kompromitace jedné CA kompromituje vše
- Revokační seznamy
  - Co když ještě nemám konektivitu?
- Možné řešení: key pinning
- Možné řešení: DNSSEC + DANE

Problémy PKI

- Seznam kořenových autorit
  - Kdo rozhodne, které tam zařadit?
- Každá CA může vydat certifikát pro kohokoliv
  - Kompromitace jedné CA kompromituje vše
- Revokační seznamy
  - Co když ještě nemám konektivitu?
- Možné řešení: key pinning
- Možné řešení: DNSSEC + DANE

Problémy PKI

- Seznam kořenových autorit
  - Kdo rozhodne, které tam zařadit?
- Každá CA může vydat certifikát pro kohokoliv
  - Kompromitace jedné CA kompromituje vše
- Revokační seznamy
  - Co když ještě nemám konektivitu?
- Možné řešení: key pinning
- Možné řešení: DNSSEC + DANE

REST

- Jak dělat aplikační rozhraní nad HTTP?
- Tabulky objektů, vlastnosti, vytváření, rušení, ...
- REpresentational State Transfer
- Využívá i další metody vedle GET a POST

REST

- Jak dělat aplikační rozhraní nad HTTP?
- Tabulky objektů, vlastnosti, vytváření, rušení, ...
- REpresentational State Transfer
- Využívá i další metody vedle GET a POST

REST

- Jak dělat aplikační rozhraní nad HTTP?
- Tabulky objektů, vlastnosti, vytváření, rušení, ...
- REpresentational State Transfer
- Využívá i další metody vedle GET a POST

REST

- Jak dělat aplikační rozhraní nad HTTP?
- Tabulky objektů, vlastnosti, vytváření, rušení, ...
- REpresentational State Transfer
- Využívá i další metody vedle GET a POST

Příklad: Příklad REST komunikace

- `POST /users/create`
- `201 CREATED`
- `Location: /users/42`
- `PUT /users/42`
- `204 NO CONTENT`
- `GET /users/42`
- `Content-Type: text/json`
- `DELETE /users/42`
- `204 NO CONTENT`

Příklad: Příklad REST komunikace

- POST /users/create
- 201 CREATED
- Location: /users/42
- PUT /users/42
- 204 NO CONTENT
- GET /users/42
- Content-Type: text/json
- DELETE /users/42
- 204 NO CONTENT

Příklad: Příklad REST komunikace

- `POST /users/create`
- `201 CREATED`
- `Location: /users/42`
- `PUT /users/42`
- `204 NO CONTENT`
- `GET /users/42`
- `Content-Type: text/json`
- `DELETE /users/42`
- `204 NO CONTENT`

Příklad: Příklad REST komunikace

- POST /users/create
- 201 CREATED
- Location: /users/42
- PUT /users/42
- 204 NO CONTENT
- GET /users/42
- Content-Type: text/json
- DELETE /users/42
- 204 NO CONTENT

Příklad: Příklad REST komunikace

- POST /users/create
- 201 CREATED
- Location: /users/42
- PUT /users/42
- 204 NO CONTENT
- GET /users/42
- Content-Type: text/json
- DELETE /users/42
- 204 NO CONTENT

Příklad: Příklad REST komunikace

- POST /users/create
- 201 CREATED
- Location: /users/42
- PUT /users/42
- 204 NO CONTENT
- GET /users/42
- Content-Type: text/json
- DELETE /users/42
- 204 NO CONTENT

Příklad: Příklad REST komunikace

- POST /users/create
- 201 CREATED
- Location: /users/42
- PUT /users/42
- 204 NO CONTENT
- GET /users/42
- Content-Type: text/json
- DELETE /users/42
- 204 NO CONTENT

Příklad: Příklad REST komunikace

- POST /users/create
- 201 CREATED
- Location: /users/42
- PUT /users/42
- 204 NO CONTENT
- GET /users/42
- Content-Type: text/json
- DELETE /users/42
- 204 NO CONTENT

Příklad: Příklad REST komunikace

- POST /users/create
- 201 CREATED
- Location: /users/42
- PUT /users/42
- 204 NO CONTENT
- GET /users/42
- Content-Type: text/json
- DELETE /users/42
- 204 NO CONTENT

Kapitola 2

IPv6

IPv6 – Motivace

- Větší adresní prostor.
- Mobilita – práce ve více sítích, přechod mezi sítěmi za běhu aplikací, domovský agent.
- Zabezpečení – šifrované a podepisované packety – protokol IPSEC.
- Autokonfigurace – zjištění informací o síti přímo ze sítě.
- Způsoby přenosu – unicast, multicast, anycast.
- Více IPv6 adres na jedno rozhraní.

Adresace v IPv6

- 128-bitová adresa
- Zápis – čtveřice šestnástkových čísel.
- Příklad:
  `3ffe:ffff:0000:f101:0210:a4ff:fee3:9562`
- Vypuštění úvodních nul:
  `3ffe:ffff:0:f101:210:a4ff:fee3:9562`
- Vypuštění sekvence `0000`:
  `3ffe:ffff::f101:210:a4ff:fee3:9562`.
- Prefix – podobně jako v IPv4 (např.:
  `3ffe:ffff::12/64`).

Otázka:

Která IPv6 adresa je zapsatelná na nejméně znaků?

Adresace v IPv6

- 128-bitová adresa
- Zápis – čtveřice šestnástkových čísel.
- Příklad:
  `3ffe:ffff:0000:f101:0210:a4ff:fee3:9562`
- Vypuštění úvodních nul:
  `3ffe:ffff:0:f101:210:a4ff:fee3:9562`
- Vypuštění sekvence `0000`:
  `3ffe:ffff::f101:210:a4ff:fee3:9562`.
- Prefix – podobně jako v IPv4 (např.:
  `3ffe:ffff::12/64`).

Otázka:

Která IPv6 adresa je zapsatelná na nejméně znaků?

Formát IPv6 packetů

- Hlavička pevné délky, řetězení hlaviček.
- Verze protokolu – 4 bity, hodnota vždy 6.
- Priorita – 8 bitů, třída provozu.
- Identifikace toku – 20 bitů.
- Délka packetu – 16 bitů.
- Next header – 8 bitů (identifikace další hlavičky, např. vyšší vrstvy).
- Hop limit – 8 bitů (ekvivalent TTL u IPv4).
- Zdrojová adresa – 128 bitů.
- Cílová adresa – 128 bitů.

Speciální IPv6 adresy

- Loopback – `::1` (ekvivalent `127.0.0.1` v IPv4).
- Nespecifikovaná adresa – `::` (ekvivalent `0.0.0.0` v IPv4).
- Lokální adresa linky – `fe80::/10`
- Adresy pro příklady – `3ffe:ffff::/32`.
- IPv4 kompatibilní adresy – `::/96`.
- IPv4 mapované adresy – `::ffff:0:0/96`.

Interpretace prefixu adresy

Úkol:

Link-local adresy jsou určeny prefixem `fe80::/10`. Jaká je numericky nejvyšší link-local adresa?

Interpretace prefixu adresy

Úkol:

Link-local adresy jsou určeny prefixem `fe80::/10`. Jaká je numericky nejvyšší link-local adresa?

- 10 nejvyšších bitů stejných, zbytek jedničky

Interpretace prefixu adresy

Úkol:

Link-local adresy jsou určeny prefixem `fe80::/10`. Jaká je numericky nejvyšší link-local adresa?

- 10 nejvyšších bitů stejných, zbytek jedničky
- `fe?f:ffff:ffff:ffff:ffff:ffff:ffff:ffff`

Interpretace prefixu adresy

Úkol:

Link-local adresy jsou určeny prefixem `fe80::/10`. Jaká je numericky nejvyšší link-local adresa?

- 10 nejvyšších bitů stejných, zbytek jedničky
- `fe?f:ffff:ffff:ffff:ffff:ffff:ffff:ffff`
- ? = 8, z toho 2 nejvyšší bity zachovat, zbytek 1

Interpretace prefixu adresy

Úkol:

Link-local adresy jsou určeny prefixem `fe80::/10`. Jaká je numericky nejvyšší link-local adresa?

- 10 nejvyšších bitů stejných, zbytek jedničky
- `fe?f:ffff:ffff:ffff:ffff:ffff:ffff:ffff`
- ? = 8, z toho 2 nejvyšší bity zachovat, zbytek 1
- 8 = `1000`, z toho 2 nejvyšší bity zachovat, zbytek 1

Interpretace prefixu adresy

Úkol:

Link-local adresy jsou určeny prefixem `fe80::/10`. Jaká je numericky nejvyšší link-local adresa?

- 10 nejvyšších bitů stejných, zbytek jedničky
- `fe?f:ffff:ffff:ffff:ffff:ffff:ffff:ffff`
- `?` = 8, z toho 2 nejvyšší bity zachovat, zbytek 1
- 8 = `1000`, z toho 2 nejvyšší bity zachovat, zbytek 1
- `1011` = `b`

Interpretace prefixu adresy

Úkol:

Link-local adresy jsou určeny prefixem `fe80::/10`. Jaká je numericky nejvyšší link-local adresa?

- 10 nejvyšších bitů stejných, zbytek jedničky
- `fe?f:ffff:ffff:ffff:ffff:ffff:ffff:ffff`
- `?` = 8, z toho 2 nejvyšší bity zachovat, zbytek 1
- 8 = `1000`, z toho 2 nejvyšší bity zachovat, zbytek 1
- `1011` = `b`
- `febf:ffff:ffff:ffff:ffff:ffff:ffff:ffff`

Privátní IPv6 adresy

- Site-local address – `fec0::/10`
  - ekvivalent privátních IPv4 adres dle RFC1918
  - nyní zastaralé (RFC 3879)
- Unique Local Address – RFC 4193.
  - prefix `fc00::/7`
  - 8. bit = 1 pro lokální přidělení, 0 pro globální.
  - 40 bitů: globální ID
  - 80 bitů: pro lokální přidělování (65 536 sítí /64)

Privátní IPv6 adresy

- Site-local address – `fec0::/10`
  - ekvivalent privátních IPv4 adres dle RFC1918
  - nyní zastaralé (RFC 3879)
- Unique Local Address – RFC 4193.
  - prefix `fc00::/7`
  - 8. bit = 1 pro lokální přidělení, 0 pro globální.
  - 40 bitů: globální ID
  - 80 bitů: pro lokální přidělování (65 536 sítí /64)

EUI-64 formát adresy

- EUI-64 formát adresy – lokální část se odvozuje z fyzické adresy.
- EUI-64 pro ethernet – MAC adresa, uprostřed vloženo `fffe`, 7. nejvyšší bit nastaven na 1 pro globálně přidělený identifikátor (MAC adresu) RFC 4291, sekce 2.5.1.

Příklad: EUI-64

MAC adresa `00:D0:B7:6B:4A:B2`
Prefix sítě `fe80::/10`
EUI-64 adresa je `fe80::2d0:b7ff:fe6b:4ab2`

- Autokonfigurace – směrovač vysílá *router advertisement*, kde je uveden `/64` prefix lokální sítě. Viz též `radvd(8)`.

EUI-64 formát adresy

- EUI-64 formát adresy – lokální část se odvozuje z fyzické adresy.
- EUI-64 pro ethernet – MAC adresa, uprostřed vloženo `fffe`, 7. nejvyšší bit nastaven na 1 pro globálně přidělený identifikátor (MAC adresu) RFC 4291, sekce 2.5.1.

Příklad: EUI-64

MAC adresa `00:D0:B7:6B:4A:B2`
Prefix sítě `fe80::/10`
EUI-64 adresa je `fe80::2d0:b7ff:fe6b:4ab2`

- Autokonfigurace – směrovač vysílá *router advertisement*, kde je uveden /64 prefix lokální sítě. Viz též `radvd(8)`.

Výpočet EUI-64 adresy

Úkol:

Jaká by byla EUI-64 adresa počítače `aisa.fi.muni.cz` (MAC adresa `00:25:B3:D7:D0:6A`) je-li adresní prefix `2001:718:801:230::/64`?

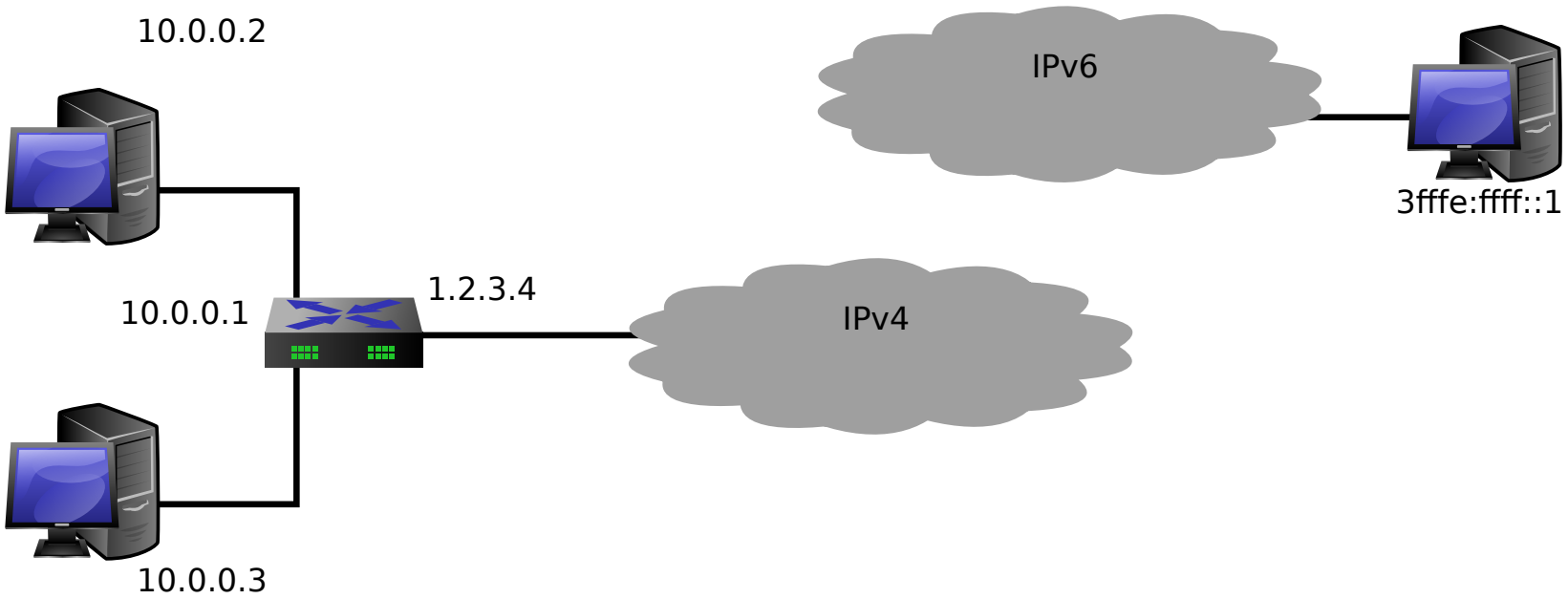
Specifické vlastnosti IPv6

- Fragmentace packetů – na směrovačích není. Vysílající musí dělat *path MTU discovery*. Fragmentace popsána v samostatné *next header*.
- Spolupráce s linkovou vrstvou – NDP (neighbour discovery protocol). Náhrada ARP. Zjišťování adresního prefixu sítě, směrovacích informací, atd.

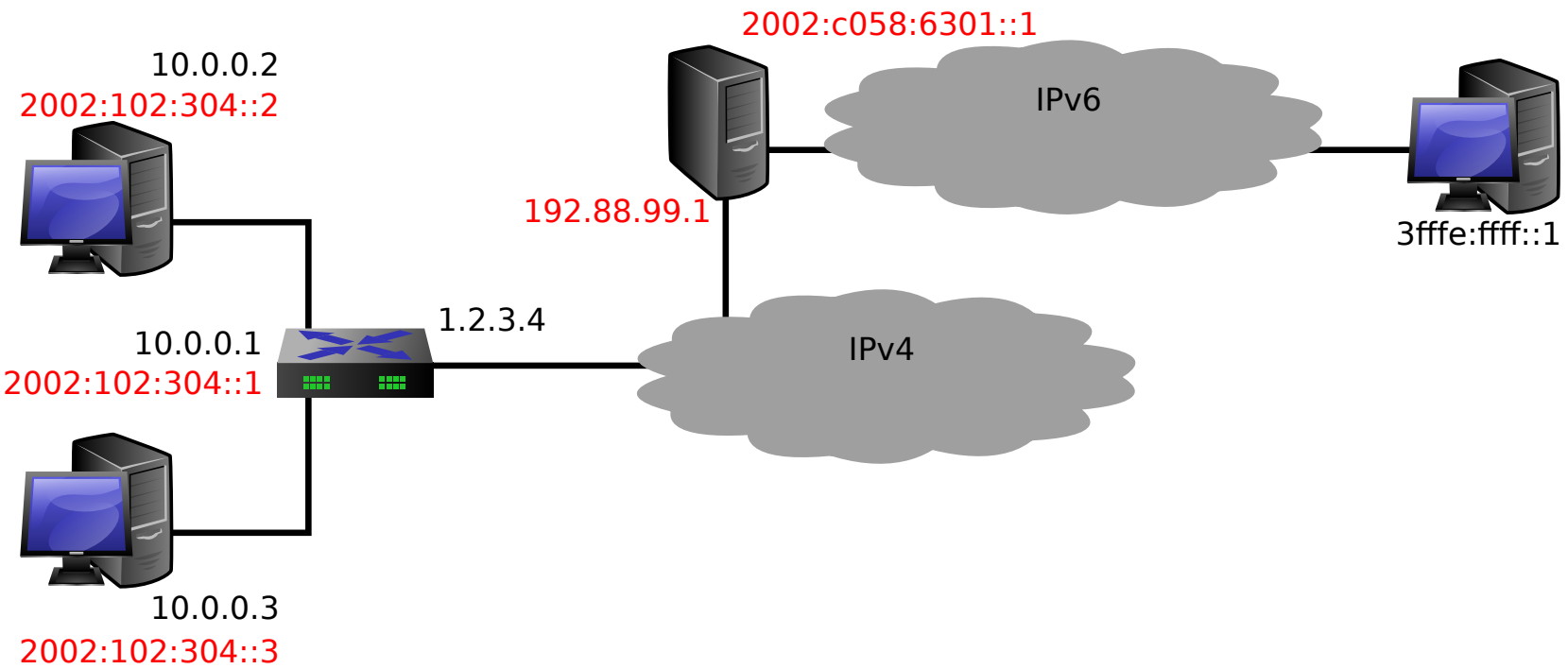
Přechodové mechanismy

- Dual stack – podpora IPv4 a IPv6 v jednom počítači.
- Tunelování – zapouzdření IPv6 packetů do IPv4 (protokol SIT, 41).
- Autotunelování – (6to4):
  - adresy `2002:xxxx:yyyy:/48` vytvořené z IPv4 adresy
  - schování bloku /48 IPv6 adres za jednu IPv4 adresu
  - komunikace do nativního IPv6 internetu přes 6to4 relay
  - adresa relay: 192.88.99.1 (2002:c058:6301::).
- 6rd – spolupráce ISP, nemá-li klient veřejnou IPv4 adresu.

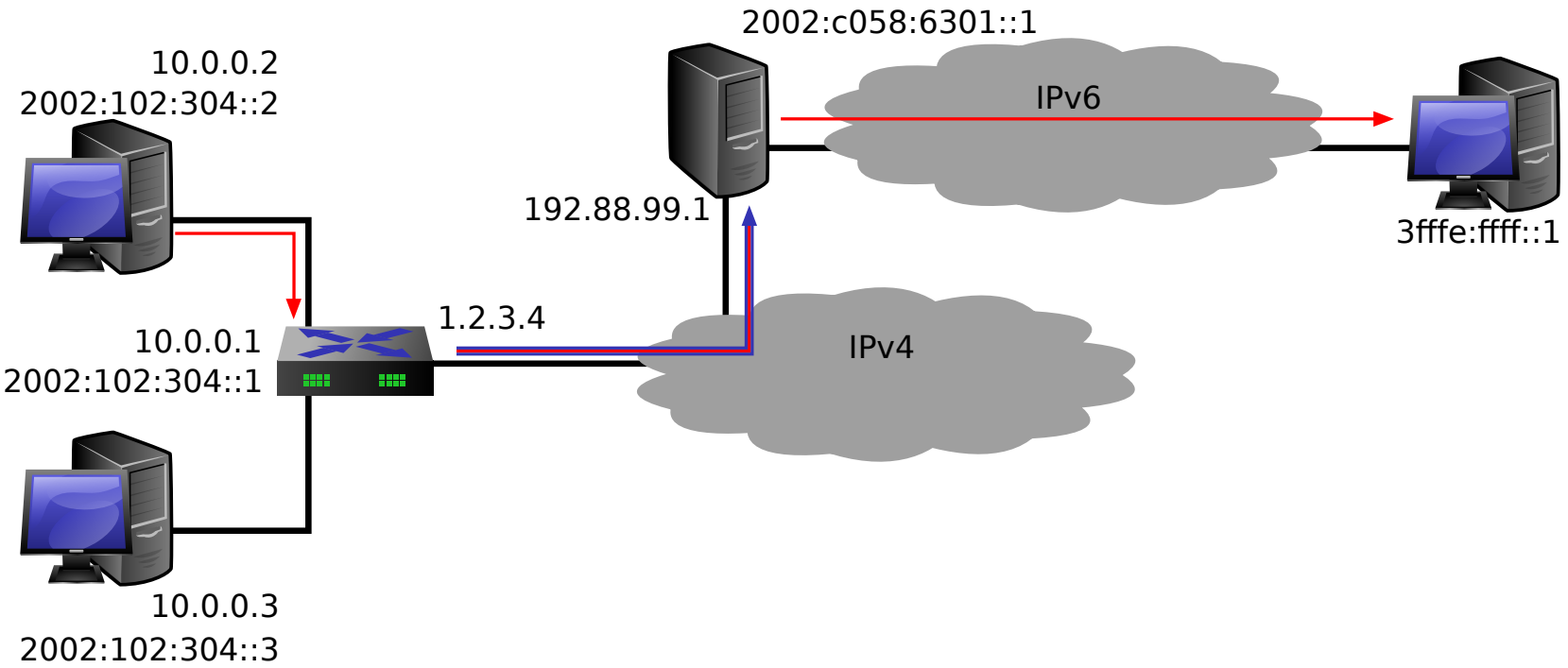
Autotunelování 6to4



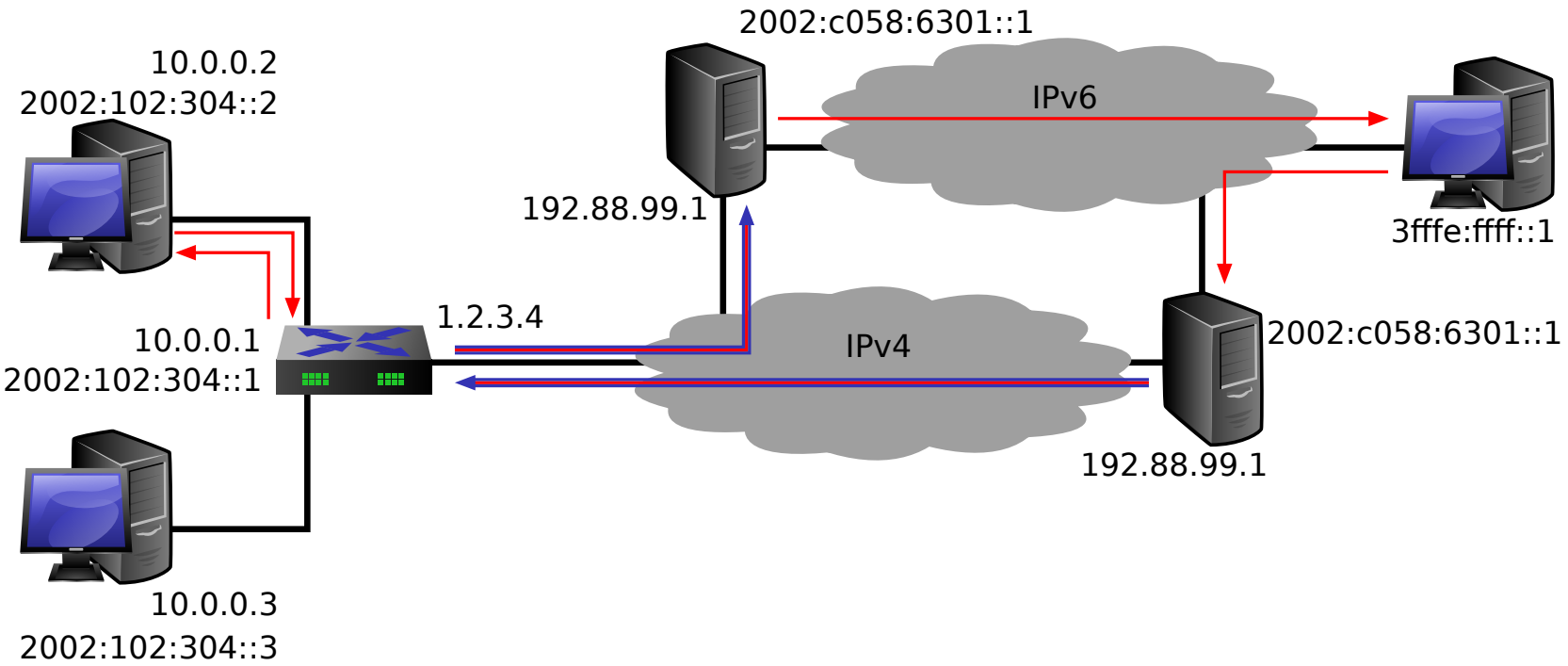
Autotunelování 6to4



Autotunelování 6to4



Autotunelování 6to4



Odkazy

- Linux and IPv6 howto – http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/
- Pavel Satrapa: IPv6 – https://knihy.nic.cz/files/edice/IPv6-2019.pdf

Čtení na dobrou noc

Zamyšlení s odstupem

The world in which IPv6 was a good design
`https://apenwarr.ca/log/20170810`