



Radek Krejčí

rkrejci@cesnet.cz

NETCONF a YANG

NETCONF

29. listopadu 2014
Praha, IT 14.2

Jak funguje protokol NETCONF

Základní charakteristiky



- klient-server protokol
- XML kódování zpráv
- RPC mechanismus
- definuje komunikace mezi klientem a serverem
- definuje způsob manipulace s konfiguračními daty a úložištmi
- RFC 6241 (v1.1) a další

Vrstvy protokolu



Transportní vrstva

- spojovaná komunikace (TCP) mezi klientem (NMS) a serverem (spravované zařízení)
- zabezpečení (autentizace, integrita, důvěrnost a ochrana proti přehrání)
- **SSH, TLS**

Vrstvy protokolu



Vrstva zpráv

- mechanismus RPC

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <close-session/>
</rpc>
```

```
<rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

- <rpc>, <rpc-reply>, <notification>

- zahrnuje i framing

- v1.0: pomocí sekvence]]>]]>

- v1.1: počet znaků ke čtení

\n#22\n

```
<rpc message-id="102"
\n#79\n
```

```
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">\n
```

```
        <close-session/>\n
```

```
</rpc>
```

```
\n##\n
```

Vrstvy protokolu



Vrstva operací

- RPC metody proveditelné serverem

- rozšiřitelná základní sada operací:

```
<get>, <get-config>, <edit-config>, <copy-config>, <delete-config>,
<lock>, <unlock>, <close-session>, <kill-session>
```

- XML formát

- párování požadavek/odpověď pomocí message-id

```
<rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>
```

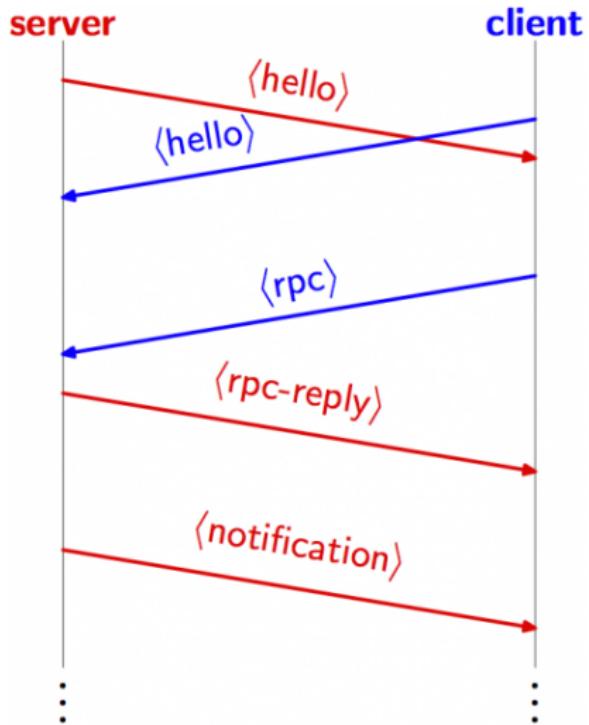
Vrstvy protokolu



Vrstva obsahu

- konfigurační a stavová data
- popis pomocí YANG modelů

Komunikace



Rozšiřitelnost (capabilities)



- možnost rozšířit základní funkcionalitu
- seznam podporovaných rozšíření součástí <hello> zprávy

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
    <capability>urn:ietf:params:netconf:base:1.1</capability>
    ...
    <capability>urn:ietf:params:netconf:capability:with-defaults:1.0?
      basic-mode=explicit&also-supported=report-all,explicit</capability>
    <capability>http://example.net/turing-machine?
      module=turing-machine&revision=2013-12-27</capability>
    <capability>urn:cesnet:tmc:netopeer:1.0?
      module=netopeer-cfgnetopeer&revision=2013-02-14</capability>
    <capability>urn:ietf:params:xml:ns:yang:ietf-netconf-server?
      module=ietf-netconf-server&revision=2014-01-24&
      features=ssh,inbound-ssh,outbound-ssh</capability>
    ...
  </capabilities>
  <session-id>10174</session-id>
</hello>
```

Konfigurační úložiště



- **running** (zapisovatelné s *:writable-running*)
- startup (*:startup*)
- candidate (*:candidate*)

Konfigurační vs. Stavová data



- dáno datovým modelem (config false/true)
- stavová data pouze pomocí <get>
- jednou operací lze měnit více konfiguračních položek najednou, klient se nemusí starat o správné pořadí

Manipulace s konfiguračními daty



získání konfigurace

- <get> nebo <get-config>
- možnost filtrování

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <get-config>
    <source><running/></source>
    <filter type="subtree">
      <turing-machine/>
    </filter>
  </get-config>
</rpc>
```

Manipulace s konfiguračními daty



editace dat

- <edit-config>
- operace merge, replace, create, delete, remove

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <edit-config>
    <target><running/></target>
    <config>
      <turing-machine xmlns="http://example.net/turing-machine"
                      xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
        <transition-function>
          <delta>
            <label>1</label>
            <input><state>1</state></input>
          </delta>
          <delta nc:operation="delete">
            <label>2</label>
          </delta>
        </transaction-function>
      </turing-machine>
    </config>
  </edit-config>
</rpc>
```

Notifikace



- rozšíření protokolu (RFC 5277)
- asynchronní zprávy o událostech
- pouze po předchozím přihlášení k odběru (`subscribe`)

Knihovna libnetconf a projekt Netopeer

libnetconf



<http://libnetconf.googlecode.com>

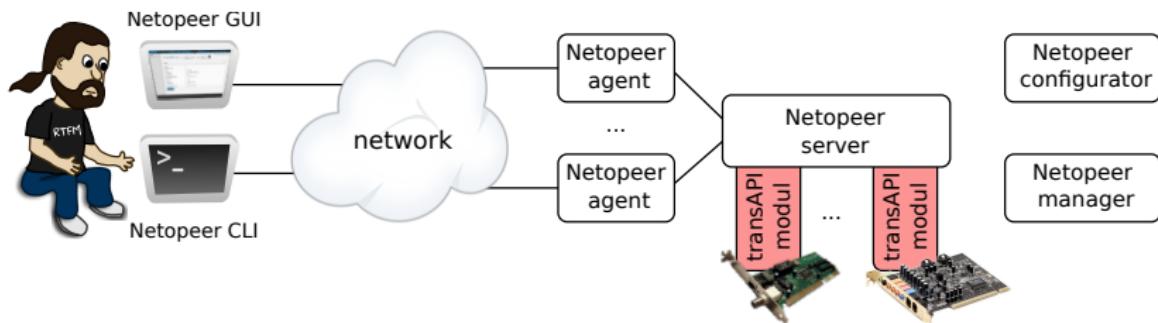
- implementace NETCONFu pro Linux
- rozhraní pro jazyk C (a Python)
- implementace klientské i serverové části
- vlastnosti:
 - verze 1.0 (RFC 4741) i 1.1 (RFC 6241)
 - SSH i TLS transport
 - Notifikace
 - NETCONF Access Control
 - Call Home
 - :writable-running, :candidate, :startup, :validate,
:url, :with-defaults

Netopeer



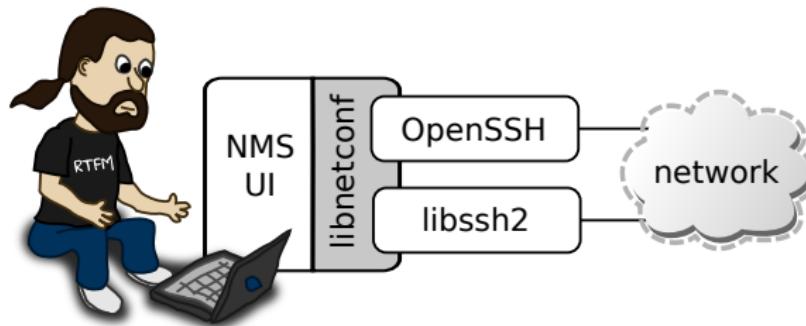
<http://netopeer.googlecode.com>
<https://github.com/CESNET/Netopeer-GUI>

- NETCONF aplikace pro Linux



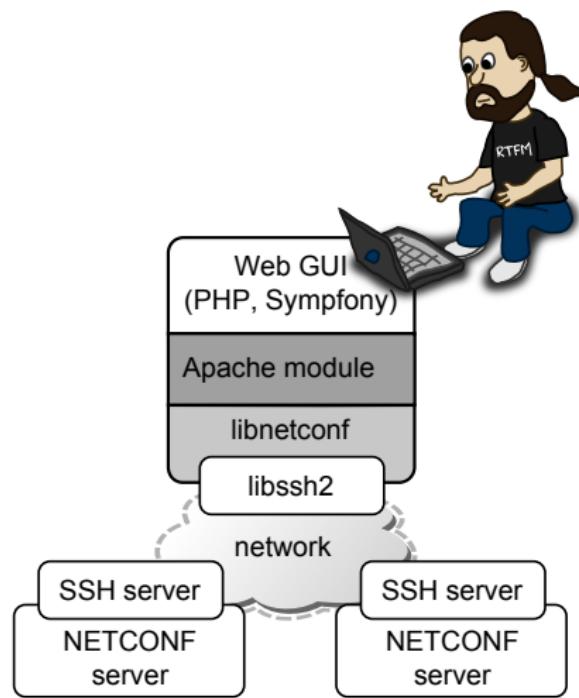
netopeer-cli

- jednoduchá příkazová řádka



Netopeer GUI

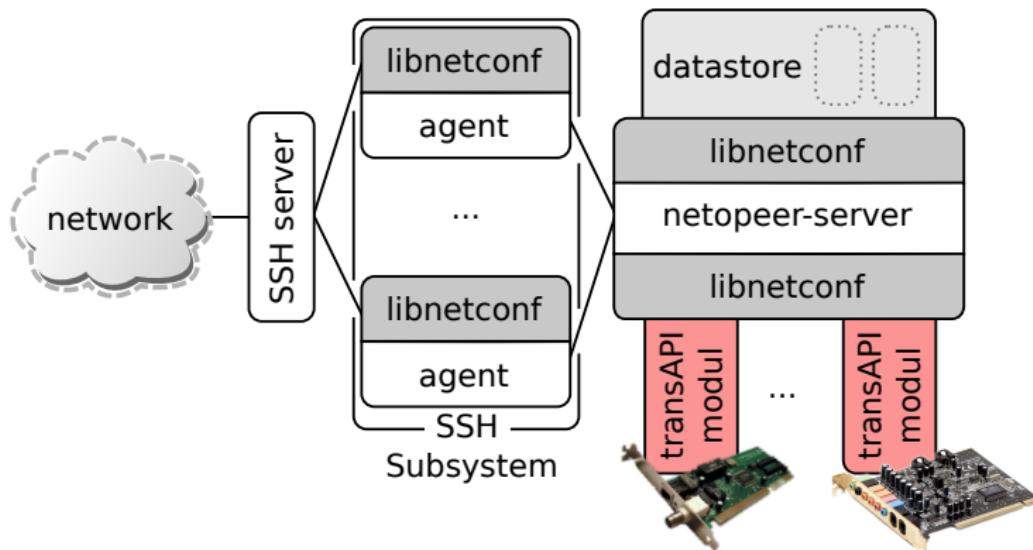
- PHP frontend s backendem udržujícím persistentní spojení



netopeer-server



- systémový daemon s moduly pro jednotlivá zařízení



TransAPI moduly



- ietf-netconf-server
- ietf-system
- ietf-interfaces, ietf-ip
- toaster, turing-machine

TransAPI moduly

Implementace Turingova stroje jako TransAPI modulu

TransAPI



- součást knihovny libnetconf
- mechanismus callbacků spojených s konkrétním datovým modelem
- generátor *Inctool(1)*

TransAPI



Globální proměnné a inicializace

- pro čtení
 - `transapi_version (r)`
 - `config_modified (w)`
- pro zápis
 - `callbacks_order (w)`
 - `erropt (r)`
- `int transapi_init(xmlDocPtr *running);`
- `void transapi_close(void);`

Datové callbacky

- volané při změně v konkrétním místě konfiguračního stromu
- informace o typu změny – přidání, smazání, změna, zpracovaná změna v potomku, změna pořadí sourozenců, změna pořadí potomků
- ```
int callback(void **data, XMLELTYPE op, xmlNodePtr node,
 struct nc_err **error);
```
- speciální callback pro získání stavových informací:  

```
xmlDocPtr get_state_data(xmlDocPtr model, xmlDocPtr running,
 struct nc_err **err);
```

# TransAPI



## RPC callbacky

- implementace RPC metod definovaných datovým modelem
- nc\_reply \*rpc(xmlNodePtr input[]);

## Souborové callbacky

- callback je vyvolaný změnou externího souboru
- ```
int file_callback(const char *filepath, xmlDocPtr *edit_config,
                  int *exec);
```

Závěr

Zdroje informací a komunikační kanály

- stránky projektů
- mailing listy

Další plány

- transAPI modul pro Open vSwitch
- knihovna libyang
- odstranění závislosti na libxml2

Díky za pozornost

NETCONF a YANG
NETCONF



Radek Krejčí
rkrejci@cesnet.cz