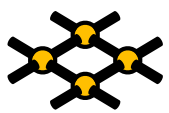
Měření propustnosti sítě a analýza paketů

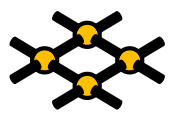
Petr Holub, David Rohleder

PB156cv
2018–03–14

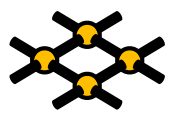


Cíle cvičení

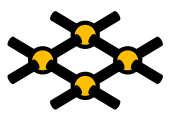
- Získat zpětnou vazbu na tvorbu protokolů na základě protokolů odevzdaných po minulém cvičení.
- Získat zkušenosti s analýzovou síťového provozu.
- Získat zkušenosti se základním testováním výkonu sítě.



Odevzdané protokoly



Základy konfigurace

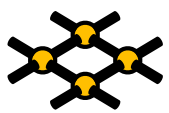


Dataplane a controlplane

Fungování síťových zařízení, zejm. přepínačů:

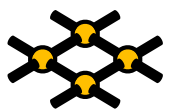
- **Data plane** – vrstva, která pomocí speciálního hardware (ASIC/FPGA) provádí hlavní činnost, tj. předávání rámců z jednoho portu na jiný.
- **Control plane** – vrstva řídící přepínač a konfigurující data plane s relativně slabým procesorem.
- Možnost zahlcení control plane.

$\Rightarrow$ vyšší výkon při dané ceně a spotřebě



Běžné druhy pamětí

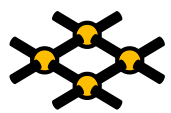
- **ROM** – obsahuje základní zavaděč OS (bootstrap).
- **RAM** – slouží k uložení běžícího operačního systému, programů a dat.
- **flash** – uložení OS, náhrada HDD.
- **NVRAM** (Nonvolatile RAM) – ukládání konfigurace, není smazána při ztrátě napájení.
- **TCAM** – rychlé vyhledávání (např. tabulky pro přepínání/směrování).



Spouštění systému

Spouštění systému sítových prvků probíhá obvykle následovně:

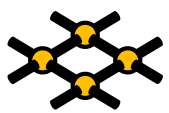
1. spustí se základní zavaděč operačního systému (bootstrap), který provede základní kontrolu inicializaci hardware. U cisco zařízení je možné v této fázi zastavit zasláním signálu BREAK po sériové lince (kermit: Ctrl-
   Ctrl-B). V bootstrapu je možné provádět pouze některé základní operace s nastavením (smazat konfiguraci, nahrát jiný OS)
2. zavaděč zjistí, co má dělat – podle nastavení registrů a pod. nahraje do paměti plnohodnotný operační systém.
3. operační systém provede inicializaci celého hardware, nahraje konfiguraci z NVRAM, spustí systémové procesy (podle konfigurace) – STP, ssh, telnet, NTP, konzolový přístup, atd.



Přístup do systému

- konzola – sériová linka RS-232. Přistupuje se pomocí programů typu kermit nebo minicom. Různí výrobci používají různá nastavení parametrů. Nejčastější je 9600 bps, 8 bits, no parity, 1 stopbit, no flow control (Cisco, Juniper, některá novější HP (starší používají buď autodetekci rychlosti nebo 19200 bps))). Cisco používá ne úplně standardní konektor RJ-45. Volba tohoto konektoru minimalizuje použitou plochu (narozdíl od standardního konektoru Canon 9) a umožnuje jednodušší propojení na konzolové servery standardními ethernetovými kabely.
- vzdálený terminálový přístup (telnet, SSH)
- webové rozhraní

`http://www.cisco.com/c/en/us/td/docs/ios/12_2/configfun/command/reference/ffun_r/frf001.html`



Úrovně uživatelských oprávnění

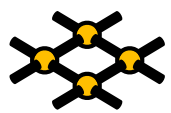
- standardní – umožňuje provádět základní operace (show, ping, traceroute, …). Cisco tento režim označuje user EXEC level, na příkazové řádce se pozná podle zobáku >

```
switch>
```

- privilegovaný uživatelský režim – umožňuje provádět změnu konfigurace, ladění a další nastavení. Cisco tento režim označuje jako enable EXEC level. Enable má u cisco IOS 15 úrovní, do kterých je možné jednotlivé příkazy zařadit. Na příkazové řádce se pozná podle mřížky #. Do privilegovaného režimu se přepíná příkazem `enable` s případným označením úrovně (standardně 15):

```
switch> enable
password:
switch#
```

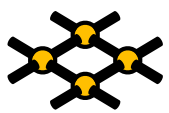
Vrátit se do standardního uživatelského režimu je možné příkazem `disable`.



Režimy uživatelského přístupu

- základní (exec) režim – umožnuje provádět nekonfigurační příkazy.
- konfigurační režim – umožnuje provádět konfiguraci, tj. měnit nastavení parametrů systému (hardware, software, rozložení pamětí, atd)

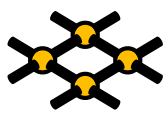
```
switch# configure terminal
switch(config)# interface FastEthernet 0/1
switch(config-if)#exit
switch(config)#end (nebo Ctrl-z)
switch#
```



Hierarchie konfigurace

Cisco IOS

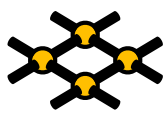
```
hostname sw12
aaa group server radius radius-servers
 server 1.2.3.4 auth-port 1812 acct-port 1813
 server 5.6.7.8 auth-port 1812 acct-port 1813
 deadtime 5
!
interface FastEthernet0/1
 description C101.1A pokusy
 switchport access vlan 71
 switchport mode access
 load-interval 30
 macro description cisco-desktop
 ip verify source
!
ntp access-group peer 77
ntp server 1.2.3.4
ntp server 5.6.7.8
```



Hierarchie konfigurace

Juniper JunOS

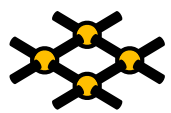
```
version 12.1R2.9;
system {
    host-name fwtest;
    domain-name test.muni.cz;
    authentication-order [ tacplus password ];
    root-authentication {
        encrypted-password "tadybylonejakeheslo";
    }
    name-server {
        1.2.3.4;
    }
    services {
        ssh;
        web-management {
            https {
                system-generated-certificate;
            }
        }
    }
```



Hierarchická konfigurace – Cisco

Cisco má poměrně plochou hierarchii, ačkoliv se postupně vyvíví:

- global – nastavování globálních parametrů (hostname, SNMP, služby, …)
- interface – nastavování parametrů fyzických i logických rozhraní
- line – nastavování parametrů sériových a virtuálních linek
- s vývojem vznikají další zanořené části konfigurace (MSTP, class-mapy, access-listy, atd.), úroven obvykle není příliš hluboká.

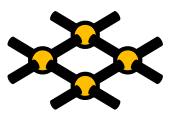


Konfigurace

Nastavení parametrů v konfiguraci probíhá pouze v RAM, rozlišujeme dva druhy konfigurace:

- running-config – tato konfigurace je uložená pouze v RAM, v případě restartu switche dochází ke smazání. Proto je nutné tuto konfiguraci nejdřív zapsat do
- startup-config – která je uložena v NVRAM (nvram:startup-config). Tato konfigurace se nahrává po spuštění systému. Zápis running config do startup-config je možný následujícími způsoby

```
switch# write memory (nebo)
switch# copy running-config startup-config
```



Ovládání CLI – Cisco

Všechny příkazy mohou být zapisovány jednoznačnými zkratkami:

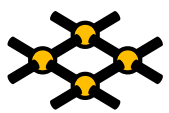
```
     sh conf  show configuration
sh int Te1/6  show interface TenGigabitEthernet 1/6
```

- no příkaz v konfiguračním režimu zruší zadaný příkaz

  ```
  switch(config-if)# no speed 100
  ```

- default XXX nastaví defaultní hodnoty

  ```
  switch(config)# default int range f0/1 - 4 , f0/6 - 48
  ```



Ladění

Každý rozumně použitelný síťový prvek musí být vybaven nástroji pro ladění problémů. Čím více možností a specifičtější zadání, tím lépe.

- zapínání ladění

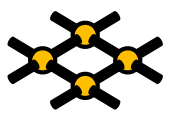
```
switch# debug ?
```

- vypínání ladění

```
switch# no debug ?
switch# undebug ?
```

- POZOR: debugování je náročné na zpracování CPU, může dojít k zahlcení systému. Proto nikdy na provozním stroji nezkoušejte

```
switch# debug all
```



Užitečná vylepšení

- čas vzniku události je důležitý – zvlášť pro porovnání vzniku události na různých zařízeních

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
```

- synchronizace času – je vhodné, aby čas na jednotlivých zařízeních byl stejný.

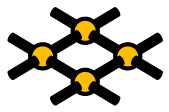
```
ntp server 1.2.3.4
ntp server 5.6.7.8
```

- posílání logů na vzdálený syslog server

```
logging trap debugging
logging 1.2.3.4
logging facility <syslog facility>
```

- na virtuálních terminálech je nutné vypisování na terminál nejdřív zapnout (jinak se zapisuje pouze do bufferu v paměti nebo na syslog server)

```
switch# terminal monitor
switch# show logging
```



Užitečná vylepšení

- čas vzniku události je důležitý – zvlášť pro porovnání vzniku událostí na různých zařízeních

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
```

- synchronizace času – je vhodné, aby čas na jednotlivých zařízeních byl stejný.

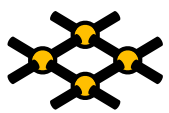
```
ntp server 1.2.3.4
ntp server 5.6.7.8
```

- posílání logů na vzdálený syslog server

```
logging trap debugging
logging 1.2.3.4
logging facility <syslog facility>
```

- na virtuálních terminálech je nutné vypisování na terminál nejdřív zapnout (jinak se zapisuje pouze do bufferu v paměti nebo na syslog server)

```
switch# terminal monitor
switch# show logging
```



Užitečná vylepšení

- čas vzniku události je důležitý – zvlášť pro porovnání vzniku událostí na různých zařízeních

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
```

- synchronizace času – je vhodné, aby čas na jednotlivých zařízeních byl stejný.

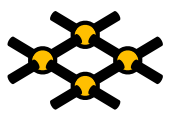
```
ntp server 1.2.3.4
ntp server 5.6.7.8
```

- posílání logů na vzdálený syslog server

```
logging trap debugging
logging 1.2.3.4
logging facility <syslog facility>
```

- na virtuálních terminálech je nutné vypisování na terminál nejdřív zapnout (jinak se zapisuje pouze do bufferu v paměti nebo na syslog server)

```
switch# terminal monitor
switch# show logging
```



Užitečná vylepšení

- čas vzniku události je důležitý – zvlášť pro porovnání vzniku událostí na různých zařízeních

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
```

- synchronizace času – je vhodné, aby čas na jednotlivých zařízeních byl stejný.

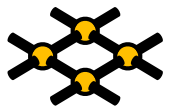
```
ntp server 1.2.3.4
ntp server 5.6.7.8
```

- posílání logů na vzdálený syslog server

```
logging trap debugging
logging 1.2.3.4
logging facility <syslog facility>
```

- na virtuálních terminálech je nutné vypisování na terminál nejdřív zapnout (jinak se zapisuje pouze do bufferu v paměti nebo na syslog server)

```
switch# terminal monitor
switch# show logging
```

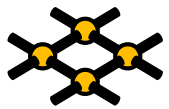


Mikrotik

- Stromová struktura příkazů

```
[admin@nekde] > /
certificate  ip    port    routing        system  blink   password  setup
driver       ipv6  ppp     snmp           tool    export  ping      undo
file         log   queue   special-login  user    import  quit
interface    mpls  radius  store          beep    led     redo

/            delay    find     if       parse    set      toid   tostr
:            do       for      len      pick     time     toip   totime
environment  error    foreach  local    put      toarray  toip6  typeof
terminal     execute  global   nothing  resolve  tobool   tonum  while
```



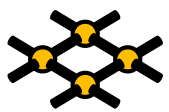
Mikrotik

```
[admin@nekde] > /interface bridge print
Flags: X - disabled, R - running
 0  R name="trunk-br" mtu=1500 l2mtu=2290 arp=enabled
      mac-address=00:0C:42:23:CF:7B protocol-mode=none priority=0x8000
      auto-mac=yes admin-mac=00:00:00:00:00:00 max-message-age=20s
      forward-delay=15s transmit-hold-count=6 ageing-time=5m

[admin@nekde] > /interface bridge set name="neco"
```

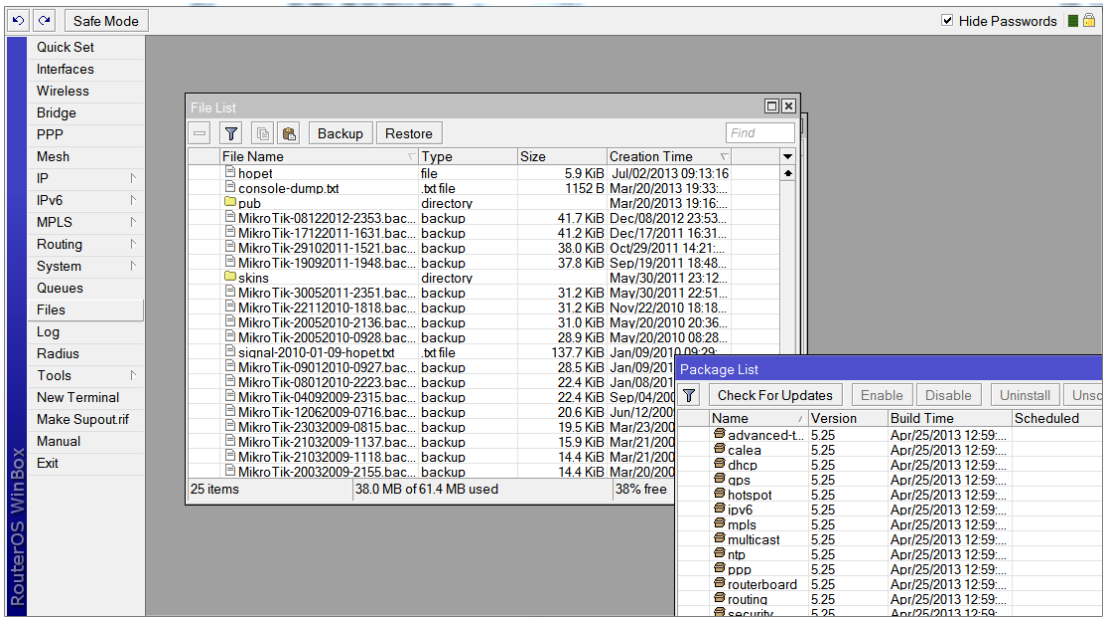
- Zálohování

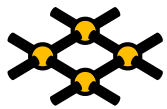
```
[admin@nekde] > /export
```



Mikrotik

- WinBox
  - jednoduché GUI pro správu
  - možnost připojení jen přes L2 MAC
  - levé menu imituje strukturu CLI + některé zkratky navíc (např. bridge)
  - drag & drop funkcionalita: → Files

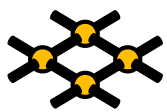




Mikrotik

- Možnosti skriptování

```
:local interface "wlan2";
/interface wireless monitor $interface once do={
:local status $"status";
:local txrate $"tx-rate";
:local rxrate $"rx-rate";
:local signal $"signal-strength";
:local snr $"signal-to-noise";
:local noise $"noise-floor";
:local thruput $"p-throughput";
:local freq $"frequency";
:local txccq $"tx-ccq";
:local rxccq $"rx-ccq";
:log info ([/system identity get name] . " " . "status: $status, \
rates tx/rx: $txrate/$rxrate, freq: $freq MHz, SNR: $snr dB, signal: \
$signal dBm, noise: $noise dBm, CCQ tx/rx: $txccq%/$rxccq%, thruput: \
$thruput");
};
```



Analýza síťového provozu

Konfigurace počítačů

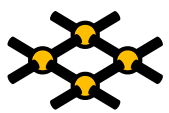
- Základní konfigurace sítě

```
ifconfig eth0 inet 10.1.1.2 netmask 255.255.255.0 up
netstat -rn
```

- Konfigurace sítě na Windows pomocí příkazové řádky

```
ipconfig /all
netsh interface ip show config
netsh interface ip set address name="Local Area Connection"
        static 192.168.1.2 255.255.255.0 192.168.1.1 1
netsh interface ip set dns "Local Area Connection" static 192.168.

netsh interface ip set address "Local Area Connection" dhcp
netsh interface ip set dns "Local Area Connection" dhcp
```



Konfigurace počítačů

- Informace o Ethernetových rozhraních:

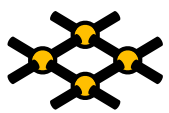
```
# ethtool eth2
# ethtool -S eth2
# netstat -s
```

- Nastavení MTU:

```
ifconfig eth0 mtu 9000
```

- Kontrola síťových bufferů `sysctl`:

```
net.core.wmem_max
net.core.wmem_default
net.core.rmem_max
net.core.rmem_default
```



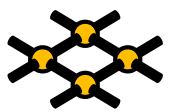
Konfigurace počítačů

- Test průchodu paketů bez fragmentace:

```
ping -M do -s 8500 -c 5 1.2.3.4
From 1.2.3.4 icmp_seq=1 Frag needed and DF set (mtu = 1500)
```

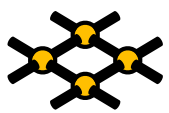
- tcpdump

```
tcpdump -i eth0 -c 1000 -s 100 -w /tmp/file icmp
```



Zadání

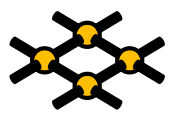
- Vytvořte z dostupných switchů dvě L2 podsítě propojené jednou linkou, kde každá z podsítí má fyzickou topologii hvězdy.
- V síti bude připojen generátor rámců. Odchytněte minimálně náhodných 10 rámců (ideálně z různých síťových toků) a proveďte jejich analýzu.
  - Při odchytávání provozu dbejte na to, aby vaše počítače neposílaly do sítě zbytečné rámce.
  - Vyzkoušejte si zachycení provozu na počítači bez GUI a následnou analýzu na jiném počítači.



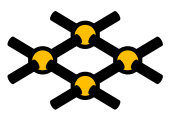
Protokol

Každý samostatně zpracuje a odevzdá protokol. Protokol musí obsahovat minimálně následující části:

- analýzu obsahu zachycených paketů, součástí protokolu bude soubor obsahující analyzované pakety ve formátu PCAP,

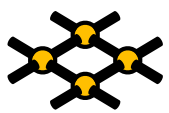


Měření end-to-end propustnosti sítě



Trocha “teorie”

- Omezení propustnosti spojů
- Omezení propustnosti síťových prvků
- Omezení propustnosti koncových zařízení
- Závislost na velikosti paketů



Provádění měření

- iperf UDP

```
iperf -s -u -i 1 -l 8500
iperf -u -c hostname -i 1 -l 8500 -b 10M
```

- iperf TCP

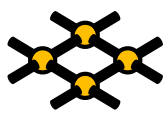
```
iperf -s -i 1 -w 8M
iperf -c hostname -i 1 -w 8M
```

- netperf UDP

```
netserver -n 4
netperf -H 10.0.10.1 -n 4 -t UDP_STREAM -- -s 8M -S 8M -m nnnn -M nnnn
```

- netperf TCP

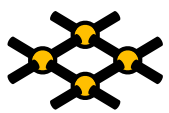
```
netserver -n 4
netperf -H 10.0.10.1 -n 4 -t TCP_STREAM -- -s 8M -S 8M -m nnnn -M nnnn
```



Provádění měření

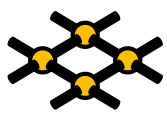
- nuttcp – trocha zábavy:

```
for h in 1.2.3.4 2.3.4.5; do for j in r t;
        do echo "";
        if [ "$j" = "r" ]; then echo "From $h to server";
        else echo "From server to $h"; fi;
                (for i in 200 400 600 800;
                        do ./nuttcp -i5 -T10 -u -R${i}M -v -v \
                                -${j} ${h};
                done ) | fgrep loss ;
        done;
done
```



Zadání

- V síti přiřaďte L3 adresy (pro jednoduchost IPv4 a není třeba žádných sofistikovaných dělení).
- Pomocí programu iperf na UDP změřte závislost dosažené rychlosti odesílání na velikosti odesílaných paketů.
- O měření vypracujte protokol.



Protokol

Každý samostatně zpracuje a odevzdá protokol. Protokol musí obsahovat minimálně následující části:

- měření výkonnosti sítě v závislosti na velikosti posílaných paketů,
- analýzu závislosti výkonu zachytávání na velikosti zachytávaných částí rámců.