TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Síťové příkazy

Datum zpracování: 28. 2. 2022

Zpracovali: Kevin Daněk

Zadání

Vaším úkolem je vyzkoušet si příkazy, které souvisejí se sítí. Pokud není uvedeno jinak, pracujte v hlavním operačním systému. Některé dílčí úlohy mohou vyžadovat i spuštěný virtuální počítač (image disku, nekomprimovaný image). Spusťte jej před zahájením práce na úlohách.

1. Příkaz ip a (ip je příkaz, a přepínač) vypíše nastavení vašich síťových rozhraní
    a. zjistěte a do protokolu zkopírujte kompletní nastavení všech sítových adaptérů
    b. okomentujte jednotlivé položky
    c. proveďte totéž pomocí příkazu ifconfig, porovnejte výstupy a krátce okomentujte
2. Příkaz ip n vypíše MAC adresy okolních počítačů a jejich přiřazení k IP adresám a kanonickým jménům
    a. vložte výstup
    b. pomocí online zdrojů zjistěte z MAC adres výrobce zjištěného síťového hardware
    c. proveďte totéž pomocí příkazu arp, porovnejte výstupy a krátce okomentujte
3. Příkaz ss vypíše provoz na síťových rozhraních
    a. vypište a okomentujte seznam aktuálních spojení
    b. vypište a okomentujte seznam otevřených portů
    c. na závěr cvičení vypište souhrnné statistiky všech protokolů a hlavní bloky okomentujte
    d. totéž proveďte příkazem netstat
4. Příkaz ping – zkusí kontaktovat konkrétní server a zjistí čas jeho odpovědi. Tato hodnota vypovídá o vzdálenosti k serveru, jeho vytížení, případně o vytížení datové linky, kterou jste připojení.
    a. Zjišťujte odezvu na servery www.tul.cz, www.seznam.cz, www.google.cz, www.facebook.com za následujících podmínek: (a porovnejte výsledky)
        i. při odesílání paketů velkých 1024 bajtů (jedním z parametrů pingu lze nastavit velikost odesílaných paketů (nejběžnější jsou 64 B velké).
    b. Zjistěte pokusy, jak maximálně velký datový paket lze odeslat pomocí PING. (Větší už neprojdou sítí a tedy na ně nepřijde odpověď). Zdůvodněte velikost limitů.
5. Příkaz traceroute - zjistí cestu k danému serveru, tedy směrovače, které jsou po cestě.
    a. Proveďte trasování k www.zoznam.sk.
    b. Odhadněte cestu, kudy data prochází. Využijte k tomu web společnosti Ripe NCC, kde můžete zadat libovolnou Evropskou adresu a databáze Vám vrátí záznam, komu IP adresa patří a jakou má adresu.

Postup

Příkaz *ip* s přepínačem *a*

Příkaz ip a vypíše nastavení všech přítomných síťových rozhraní.

Prvním přítomným zařízením byl tzv. **loopback**. Jedná se o softwarové rozhraní, standardně s adresou 127.0.0.1 (RFC5735, sekce 4), které je používáno jako „interní host“ (RFC1122, sekce 3.2.1.3) pro dané zařízení, tedy komunikace skrz toto rozhraní nikdy neopouští lokální zařízení a slouží jako smyčka pro testování a simulování provozu na síťové kartě.

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

Druhým rozhraním je PCI přídavná karta

```
2: p2p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether b4:96:91:26:1a:e9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.156/24 brd 192.168.1.255 scope global noprefixroute dynamic
p2p1
       valid_lft 42820sec preferred_lft 42820sec
    inet6 fd03:8b53:4d8b:0:b696:91ff:fe26:1ae9/64 scope global mngtmpaddr
dynamic
       valid_lft forever preferred_lft forever
    inet6 fe80::b696:91ff:fe26:1ae9/64 scope link
       valid_lft forever preferred_lft forever
```

Třetí rozhraní je ethernetové rozhraní.

```
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 54:bf:64:62:ce:4b brd ff:ff:ff:ff:ff:ff
    inet 147.230.78.78/21 brd 147.230.79.255 scope global noprefixroute dynamic
enp0s31f6
       valid_lft 36532sec preferred_lft 36532sec
    inet6 fe80::7ffe:8963:bc14:f9ce/64 scope link tentative noprefixroute
dadfailed
       valid_lft forever preferred_lft forever
    inet6 fe80::6671:3582:2c7e:c8ac/64 scope link tentative noprefixroute
dadfailed
       valid_lft forever preferred_lft forever
    inet6 fe80::53df:8fde:4335:467/64 scope link tentative noprefixroute
dadfailed
       valid_lft forever preferred_lft forever
```

Čtvrtým rozhraním je bezdrátové rozhraní na druhé PCI sběrnici.

```
4: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 7c:2a:31:1f:63:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.174.82/21 brd 192.168.175.255 scope global noprefixroute
dynamic wlp2s0
       valid_lft 429sec preferred_lft 429sec
    inet6 fe80::f252:f0a9:fda:77f6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Pátým rozhraním je virtuální most, který běžně slouží pro překlad síťových adres (NAT) virtuálních počítačů.

```
5: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
group default qlen 1000
    link/ether 52:54:00:b8:63:ee brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
```

Šesté rozhraní je podobně jako páté rozhraní virtuální most pro virtuální počítače. Toto by mělo sloužit ke komunikaci mezi síťovou kartou hostitelského počítače a virtuální síťovou kartou virtualizovaného počítače. (Previtera, 2016)

```
6: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0
state DOWN group default qlen 1000
    link/ether 52:54:00:b8:63:ee brd ff:ff:ff:ff:ff:ff
```

Sedmým a posledním rozhraním je virtuální rozhraní pro Docker a jeho containery.

```
7: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN group default
    link/ether 02:42:4f:01:78:c4 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
       valid_lft forever preferred_lft forever
```

Příkaz *ifconfig*

```
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 0.0.0.0
        ether 02:42:4f:01:78:c4  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s31f6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 147.230.78.78  netmask 255.255.248.0  broadcast 147.230.79.255
        inet6 fe80::7ffe:8963:bc14:f9ce  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::6671:3582:2c7e:c8ac  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::53df:8fde:4335:467  prefixlen 64  scopeid 0x20<link>
        ether 54:bf:64:62:ce:4b  txqueuelen 1000  (Ethernet)
        RX packets 22813958  bytes 9880534949 (9.2 GiB)
        RX errors 387  dropped 0  overruns 0  frame 309
        TX packets 10545882  bytes 12958855801 (12.0 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xee300000-ee320000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12392  bytes 4411552 (4.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12392  bytes 4411552 (4.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

p2p1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.156  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fd03:8b53:4d8b:0:b696:91ff:fe26:1ae9  prefixlen 64  scopeid
0x0<global>
        inet6 fe80::b696:91ff:fe26:1ae9  prefixlen 64  scopeid 0x20<link>
        ether b4:96:91:26:1a:e9  txqueuelen 1000  (Ethernet)
        RX packets 1898  bytes 327729 (320.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 344  bytes 49157 (48.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xee100000-ee1fffff

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:b8:63:ee  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.174.82  netmask 255.255.248.0  broadcast 192.168.175.255
        inet6 fe80::f252:f0a9:fda:77f6  prefixlen 64  scopeid 0x20<link>
        ether 7c:2a:31:1f:63:23  txqueuelen 1000  (Ethernet)
        RX packets 59665  bytes 5644286 (5.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 51191  bytes 3875145 (3.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Nástroj *ip* a *ifconfig* se zdají jako totožné nástroje, ale na první pohled se mi zdá, že IP je spíše nástroj analytický (pro zjištění podrobné konfigurace rozhraní), zatímco ifconfig nástroj statistický. Program *ip* je kombinací několika starších nástrojů do jedné šikovné sady.

Příkaz *ip n*

```
[kevin.danek@a0307 ~]$ ip n
147.230.72.250 dev enp0s31f6 lladdr d0:c7:89:a9:d2:80 DELAY
192.168.175.250 dev wlp2s0  FAILED
192.168.1.1 dev p2p1 lladdr cc:2d:e0:e0:ba:fa STALE
147.230.75.227 dev enp0s31f6 lladdr 6c:f0:49:73:5d:01 STALE
147.230.77.233 dev enp0s31f6 lladdr 10:98:36:a2:75:fb REACHABLE
147.230.78.71 dev enp0s31f6 lladdr 54:bf:64:62:cb:51 STALE
147.230.77.30 dev enp0s31f6 lladdr 10:65:30:d6:2c:d6 STALE
fe80::d2c7:89ff:fea9:d280 dev enp0s31f6 lladdr d0:c7:89:a9:d2:80 router STALE
fe80::ce2d:e0ff:fee0:bafa dev p2p1 lladdr cc:2d:e0:e0:ba:fa router STALE
```

MAC Adresa	Výrobce
d0:c7:89:a9:d2:80	Cisco Systems, Inc
cc:2d:e0:e0:ba:fa	Routerboard.com
6c:f0:49:73:5d:01	GIGA-BYTE TECHNOLOGY CO., LTD.
10:98:36:a2:75:fb	Dell Inc.
54:bf:64:62:cb:51	Dell Inc.
10:65:30:d6:2c:d6	Dell Inc.

Příkaz *arp*

```
[kevin.danek@a0307 ~]$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
router-b.tul.cz          ether   d0:c7:89:a9:d2:80   C
enp0s31f6
gateway                          (incomplete)
wlp2s0
gateway                  ether   cc:2d:e0:e0:ba:fa   C                     p2p1
alva.nti.tul.cz          ether   6c:f0:49:73:5d:01   C
enp0s31f6
share.nti.tul.cz         ether   10:98:36:a2:75:fb   C
enp0s31f6
a0300.nti.tul.cz         ether   54:bf:64:62:cb:51   C
enp0s31f6
dockms.nti.tul.cz        ether   10:65:30:d6:2c:d6   C
enp0s31f6
```

Rozdíl mezi výstupem *ip n* a *arp* je na první pohled v přeložených IP adresách na názvy hostů, dále také ve formátování výstupu.

Příkaz *ss*

Příkaz aktuálně otevřených spojení, lze vidět, že jsou skoro všechny v rámci TUL sítě (začínající na 147.230), až na otevřená HTTPS spojení. Dále je vidět ssh spojení, kterým jsem připojen vzdáleně na danou stanici.

```
tcp    0      0                                        147.230.78.78:39428
147.230.18.154:ldaps
timer:(keepalive,20min,0)
tcp    0      98572                                    147.230.78.78:ssh
147.230.231.20:59943
timer:(on,205ms,0)
tcp    0      0                                        147.230.78.78:kink
147.230.77.233:nfs
timer:(keepalive,58sec,0)
tcp    0      0                                        147.230.78.78:59640
34.209.131.4:https
timer:(keepalive,1min30sec,0)
tcp    0      0                                        147.230.78.78:39424
147.230.18.154:ldaps
timer:(keepalive,79min,0)
tcp    0      0                                        147.230.78.78:39432
147.230.18.154:ldaps
timer:(keepalive,69min,0)
tcp    0      0                                        147.230.78.78:39466
147.230.18.154:ldaps
timer:(keepalive,29min,0)
tcp    0      0                                        147.230.78.78:39426
147.230.18.154:ldaps
timer:(keepalive,69min,0)
tcp    0      0                                        147.230.78.78:39478
52.38.198.132:https
timer:(keepalive,4min30sec,0)
```

Seznam otevřených portů zahrnuje standardně *ssh* pro vzdálený přístup, *smtp* pro mailové služby či *ipp* pro komunikaci s tiskárnami.

```
tcp    LISTEN     0      100                                  127.0.0.1:smtp
*:*
tcp    LISTEN     0      128                                        *:sunrpc
*:*
tcp    LISTEN     0      5                                192.168.122.1:domain
*:*
tcp    LISTEN     0      128                                        *:ssh
*:*
tcp    LISTEN     0      128                                  127.0.0.1:ipp
*:*
tcp    LISTEN     0      100                                      [::1]:smtp
[::]:*
tcp    LISTEN     0      128                                       [::]:sunrpc
[::]:*
tcp    LISTEN     0      128                                       [::]:ssh
[::]:*
tcp    LISTEN     0      128                                      [::1]:ipp
[::]:*
```

Souhrnné statistiky příkazu *ss* ukazují krátký souhrn toho, co zrovna zpracovává. Jedná se spíše o orientační souhrn, který nám prozradí, kolik tak je otevřených připojení.

```
[kevin.danek@a0307 ~]$ ss -s
Total: 2701 (kernel 0)
TCP:   19 (estab 9, closed 1, orphaned 0, synrecv 0, timewait 0/0), ports 0

Transport Total     IP        IPv6
*         0         -         -
RAW       2         0         2
UDP       13        10        3
TCP       18        14        4
INET      33        24        9
FRAG      0         0         0
```

Příkaz *netstat*

Příkazem netstat s přepínači *a* a *t* se mi podařilo vypsat všechna otevřená (*established*) připojení, tak také všechny otevřené porty (*listen*).

```
[kevin.danek@a0307 ~]$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:sunrpc          0.0.0.0:*               LISTEN
tcp        0      0 a0307.nti.tul.cz:domain 0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 a0307.nti.tul.cz:39428  ldap-proxy.nti.tu:ldaps ESTABLISHED
tcp        0      0 a0307.nti.tul.cz:kink   share.nti.tul.cz:nfs    ESTABLISHED
tcp        0      0 a0307.nti.tul.cz:59640  ec2-34-209-131-4.:https ESTABLISHED
tcp        0      0 a0307.nti.tul.cz:39424  ldap-proxy.nti.tu:ldaps ESTABLISHED
tcp        0    208 a0307.nti.tul.cz:ssh    147.230.231.20:58649    ESTABLISHED
tcp        0      0 a0307.nti.tul.cz:39432  ldap-proxy.nti.tu:ldaps ESTABLISHED
tcp        0      0 a0307.nti.tul.cz:39466  ldap-proxy.nti.tu:ldaps ESTABLISHED
tcp        0      0 a0307.nti.tul.cz:39426  ldap-proxy.nti.tu:ldaps ESTABLISHED
tcp        0      0 a0307.nti.tul.cz:39478  ec2-52-38-198-132:https ESTABLISHED
tcp6       0      0 localhost:smtp          [::]:*                  LISTEN
tcp6       0      0 [::]:sunrpc             [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 localhost:ipp           [::]:*                  LISTEN
```

Souhrnné statistiky příkazu *netstat* už toho pověděli trochu víc, než statistiky příkazu *ss*. Vzhledem k délce výstupu jsem si dovolil ho trochu zkrátit.

```
[kevin.danek@a0307 ~]$ netstat -s
Ip:
    9218048 total packets received
    0 forwarded
    0 incoming packets discarded
    5844695 incoming packets delivered
    2538800 requests sent out
    16 dropped because of missing route
    2 fragments dropped after timeout
    52 reassemblies required
    25 packets reassembled ok
    2 packet reassembles failed
Tcp:
    6249 active connections openings
    12 passive connection openings
    0 failed connection attempts
    128 connection resets received
    9 connections established
    5767021 segments received
    10441835 segments send out
    5362 segments retransmited
    1 bad segments received.
    605 resets sent
Udp:
    53758 packets received
    9 packets to unknown port received.
    0 packet receive errors
    58983 packets sent
    0 receive buffer errors
    0 send buffer errors
…
```

Příkaz *ping*

Příkazem *ping* se dá zjistit, jaká je odezva nějakého daného serveru. Odezva se může odvíjet od jeho geografické vzdálenosti, ale nebo také od jeho vytíženosti. Obvykle nás zajímá tzv. RTT, neboli *round-trip time*, což je čas, který packet potřeboval k tomu, než dorazil do cíle a zpátky k nám.

Server	Packet Loss	Min. RTT [ms]	Avg. RTT [ms]	Max. RTT [ms]	RTT Dev [ms]
www.tul.cz	0/5	0.409	0.455	0.527	0.047
www.seznam.cz	0/5	3.773	3.935	4.001	0.088
www.google.cz	0/5	3.645	3.808	3.873	0.099
www.facebook.com	0/5	3.670	3.815	3.894	0.120

Experimenty jsem dospěl k závěru, že lze pingem poslat packet o maximální velikosti 1472B (při pingování na 8.8.8.8).

Příkaz *traceroute*

Příkaz *traceroute* slouží k zmapování cesty packetu od naší stanice k cíli. Cesta odhadem rozhodně půjde mimo síť TUL, tedy projde přes nějaké prvky sítě LIANE ven, následně poputuje po nějakých rozvodech na Slovensko, kde by měla dorazit na vstupní bránu (gate) zoznamu.

Domněnku nyní ověříme výstupem z příkazu *traceroute*

```
[kevin.danek@a0307 ~]$ traceroute www.zoznam.sk
traceroute to www.zoznam.sk (213.81.185.168), 30 hops max, 60 byte packets
 1  router-b.tul.cz (147.230.72.250)  0.373 ms  0.318 ms  0.239 ms
 2  router-h.tul.cz (147.230.250.18)  0.405 ms  0.375 ms  0.444 ms
 3  147.230.250.49 (147.230.250.49)  0.501 ms  0.457 ms  0.449 ms
 4  195.113.235.99 (195.113.235.99)  4.355 ms  4.322 ms  4.304 ms
 5  nix4.telekom.sk (91.210.16.23)  4.084 ms  4.087 ms  4.101 ms
 6  st-static-srk238.87-197-252.telecom.sk (87.197.252.238)  8.566 ms  8.872 ms
8.918 ms
```

Domněnka byla téměř správná, ale *traceroute* nás nechal na serveru telecom.sk, ne zoznam.sk. Zkusil jsem tedy výstup z *traceroute* porovnat proti výstupu ekvivalentu v prostředí Windows 10, *tracert*, který mě zavedl ještě o ten jeden hop dál.

```
  1    812 ms     22 ms     14 ms  147.230.239.250
  2     44 ms     36 ms    511 ms  router-h.tul.cz [147.230.250.18]
  3     10 ms     34 ms      5 ms  147.230.250.49
  4    590 ms    608 ms    204 ms  195.113.235.99
  5     41 ms    550 ms    237 ms  nix4.telekom.sk [91.210.16.23]
  6     15 ms     37 ms     12 ms  st-static-srk238.87-197-252.telecom.sk
[87.197.252.238]
  7     13 ms     14 ms     14 ms  rev-213-81-185-168.zoznam.sk [213.81.185.168]
```

Důsledky tohoto rozdílu neznám, první, kam bych se rozhodně díval, tak jsou asi nějaké přepínače příkazu *traceroute*. Popř. to jistě nějací hodní lidé již prodiskutovali na StackOverflow.

Závěr

Cílem cvičení bylo osahat si základní síťové příkazy v Linuxu, ať už to ty starší, či novější iterace. Během tohoto cvičení jsem se dozvěděl například o nástroji *ss* či *ip.* Tyto nástroje budou skvělé v kombinaci s příkazy na zpracování textu (grep, awk, tr, …) a případné další skriptování.