Network configuration laboratories

Authors: Piotr Copek, Zuzanna Micorek

Date: 26.03.2025

Introduction

The purpose of this laboratory is to get familiar with the fundamentals of computer networking and gain hands-on experience with essential commands for network management. Understanding these concepts is crucial for diagnosis and solving network issues.

1. Testing routers operation, ICMP protocol

```
# ping -c 5 www.polsl.pl
```

Check the servers response times: your local gateway in your network, www.polsl.pl, www.africa.com, www.japan.com. How does the ping command work?

```
# ping -c 5 www.polsl.pl

PING www.polsl.pl (157.158.70.85) 56(84) bytes of data.

--- www.polsl.pl ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4124ms
```

If a device is unreachable, `ICMP` sends Destination Unreachable messages.

```
# ping -c 5 www.africa.com

PING wp.wpenginepowered.com (141.193.213.10) 56(84) bytes of data.
64 bytes from 141.193.213.10: icmp_seq=1 ttl=51 time=29.3 ms
64 bytes from 141.193.213.10: icmp_seq=2 ttl=51 time=82.2 ms
64 bytes from 141.193.213.10: icmp_seq=3 ttl=51 time=41.2 ms
64 bytes from 141.193.213.10: icmp_seq=4 ttl=51 time=39.9 ms
64 bytes from 141.193.213.10: icmp_seq=5 ttl=51 time=38.7 ms

--- wp.wpenginepowered.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 29.279/46.274/82.216/18.457 ms
```

`ping` uses ICMP Echo Request and Echo Reply messages to check connection and measure round-trip time.

```
# ping -c 3 www.japan.com

PING www.japan.com (172.67.70.92) 56(84) bytes of data.
64 bytes from 172.67.70.92: icmp_seq=1 ttl=51 time=201 ms
64 bytes from 172.67.70.92: icmp_seq=2 ttl=51 time=65.1 ms
64 bytes from 172.67.70.92: icmp_seq=3 ttl=51 time=56.5 ms

--- www.japan.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 56.535/107.623/201.245/66.292 ms
```

It helps in network troubleshooting but can be blocked for security reasons.

2. Path of the packet in Internet, packet TTL field

```
# traceroute www.polsl.pl
# traceroute -I www.polsl.pl
```

Traceroute shows the path taken by packets between two hosts on the IP network, including all routers along the way. Find the path to the servers: www.polsl.pl, google.pl. Set different TTL (Time To Live) values for packet in the traceroute command, e.g. 4. Check that the entire route

cannot be traced. How does the traceroute command work and what protocol does it use?

```
# traceroute www.polsl.pl
traceroute to www.polsl.pl (157.158.70.85), 30 hops max, 60 byte packets
 1  LAPTOP-8JKTK9N3.mshome.net (172.25.112.1)  0.598 ms  0.570 ms  0.623 ms
 2  192.168.21.253 (192.168.21.253)  6.173 ms  7.004 ms  6.997 ms
 3  * * *
    ...
 30 * * *
```

First Hop: (`172.25.112.1` - `LAPTOP-8JKTK9N3.mshome.net`)
Second Hop: (`192.168.21.253`)
Other hops: asterics (`* * *`) indicate that the hop did not respond.

TTL is a field in the IP header that prevents packets from looping infinietly. Each time a packet passes through a router TTL value is decreased. When TTL reaches zero, the router discards the packet and sends an ICMP Time Exceeded message back. `traceroute` uses this mechanism to map the route that packet takes.

```
# traceroute -I www.polsl.pl

traceroute to www.polsl.pl (157.158.70.85), 30 hops max, 60 byte packets
 1  LAPTOP-8JKTK9N3.mshome.net (172.25.112.1)  0.754 ms  0.737 ms *
 2  192.168.21.253 (192.168.21.253)  7.396 ms  7.394 ms *
 3  192.0.0.1 (192.0.0.1)  11.911 ms * *
 4  * * *
    ...
 30 * * *
```

Flag `-I` uses ICMP instead of UDP, which may cause different results.

First Hop: (`172.25.112.1` - `LAPTOP-8JKTK9N3.mshome.net`)
Second Hop: (`192.168.21.253`)
Third hop: (`192.0.0.1`)
Other hops: asterics (`* * *`) indicate that the hop did not respond.

3. ARP protocol, maping the MAC address to IP Address

```
# ip neighbour
```

Show ARP table.
Delete the gateway entry for your network from the table. Perform any operation with an Internet or "ping" to the gateway. Check if the deleted entry appears again in the ARP table. Delete all items from the table. Observe the filling up of the array as you use the network (e.g. by pinging local computers).

```
# ip neighbour

172.25.112.1 dev eth0 lladdr 00:15:5d:d8:27:36 STALE
```

`172.25.112.1` → IP of your network gateway (router).

`eth0` → Network interface handling this connection.

`00:15:5d:d8:27:36` → MAC address of the gateway.

`STALE` → The entry is old but not removed. It will be updated when communication happens again.

```
# ip neighbour delete 172.25.112.1 dev eth0
```

This removes the entry, but the gateway is still functional.

```
# ping 172.25.112.1

PING 172.25.112.1 (172.25.112.1) 56(84) bytes of data.
^C
--- 172.25.112.1 ping statistics ---
85 packets transmitted, 0 received, 100% packet loss, time 87326ms
```

Performed some random network activity.

```
# ip neighbour show
172.25.112.1 dev eth0 lladdr 00:15:5d:d8:27:36 REACHABLE
```

The entry reappear because ARP automatically requests the MAC address again when sending data.

```
# ip neighbour flush all
```

Now `ip neighbour show` does nothing.

I used the same `ping 172.25.112.1` and after using 'ip neighbour show' such result was shown:

```
172.25.112.1 dev eth0 lladdr 00:15:5d:d8:27:36 REACHABLE
```

ARP resolves IP addresses to MAC addresses within a local network. The device response with the matching IP responds with its MAC address. The ARP table stores these mappings temporarily. Dynamic ARP entries expire after some time.

4. Information about network interfaces

```
# ip link show
# ip address show
(# ifconfig)
```

What information can be read from these commands? What are the names of network interfaces in your system? What is the MAC address of the network card, IP address, netmask?

```
# ip link show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether 00:15:5d:e6:b9:82 brd ff:ff:ff:ff:ff:ff
```

- `lo` - loopback interface
- `eth0` - main network interface
- `MAC address` - `00:15:5d:e6:b9:82`
- `Broadcast MAC` - `ff:ff:ff:ff:ff:ff`

```
ip address show
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
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:15:5d:e6:b9:82 brd ff:ff:ff:ff:ff:ff
    inet 172.25.119.133/20 brd 172.25.127.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fee6:b982/64 scope link
       valid_lft forever preferred_lft forever
```

- `IP` - `172.25.119.133`
- `netmask` - `/20`
- `link loacal IPv6` - `fe80::215:5dff:fee6:b982/64`
- `IPv4` - `127.0.0.1/8`
- `IPv6` - `::1/128`

5. TCP and UDP transmissions performed

```
# ip link
# ip -s -s -h address
# ss
```

Check how many bytes were transmitted correctly and with errors in TCP and UDP protocols on individual interfaces (TX = send, RX = receive).

```
# ip link

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether 00:15:5d:e6:b9:82 brd ff:ff:ff:ff:ff:ff
```

```
# ip -s -s -h address

 ip -s -s -h address
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
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
    RX:  bytes packets errors dropped  missed   mcast
         18.5M   3.58k      0       0       0       0
    RX errors:  length    crc   frame    fifo overrun
                     0      0       0       0       0
    TX:  bytes packets errors dropped carrier collsns
         18.5M   3.58k      0       0       0       0
    TX errors: aborted   fifo  window heartbt transns
                     0      0       0       0       0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
```

- Recieved data RX:
  - Loopback (lo):
    - Bytes received: 18.5M
    - Packets received: 3.58k
    - Errors: 0
    - Dropped packets: 0
  - Ethernet (eth0):
    - Bytes received: 79.2k
    - Packets received: 238
    - Errors: 0
    - Dropped packets: 0
    - Multicast packets: 99
- Transmitted data TX:
  - Loopback (lo):

        - Bytes sent: 18.5M
        - Packets sent: 3.58k
        - Errors: 0
        - Dropped packets: 0
    - Ethernet (eth0):
        - Bytes sent: 34.8k
        - Packets sent: 416
        - Errors: 0
        - Dropped packets: 0
        - Retransmissions: 1

```
# ss
```

Displays open TCP/UDP connections and shows which ports are listening or actively communicating.

6. Opened ports

Display all TCP sockets with service names:
`# ss -at`
Display all TCP sockets with port numbers:
`# ss -atn`
Display all UDP sockets:
`# ss -au`
`(# netstat -atup)`

Which external ports are open for TCP and UDP connections, e.g. when a used web browser. Which ports are just listening? Which ports have connections already established on?

```
# ss -at

State     Recv-Q    Send-Q    Local Address:Port      Peer Address:Port
LISTEN    0         5         0.0.0.0:5901            0.0.0.0:*
LISTEN    0         1000      10.255.255.254:domain   0.0.0.0:*
ESTAB     0         0         172.25.119.133:51420    192.168.21.77:4713
LISTEN    0         5         [::]:5901               [::]:*
```

TCP Ports `5901` and `53` are listening and waiting for connections.
Local Port `51420` communicating with `192.168.21.77` on Port `4713` is established.

```
# ss -atn

State     Recv-Q    Send-Q    Local Address:Port      Peer Address:Port
LISTEN    0         5         0.0.0.0:5901            0.0.0.0:*
LISTEN    0         1000      10.255.255.254:53       0.0.0.0:*
ESTAB     0         0         172.25.119.133:51420    192.168.21.77:4713
LISTEN    0         5         [::]:5901               [::]:*
```

TCP Ports `5901` and `53` are listening and waiting for connections.
Local Port `51420` (Local) is communicating with `4713` (Remote).

7. Static, local routing:

```
# ip route
(# route -n)
```

Show the routing table of your computer. To what address will outgoing packets be directed to the Internet? Through which interfaces?

```
# ip route

default via 172.25.112.1 dev eth0 proto kernel
172.25.112.0/20 dev eth0 proto kernel scope link src 172.25.119.133
```

Outgoing packets with destinations not matching any other route will be directed to the default gateway `172.25.112.1` via the eth0.

8. DNS, domain names and IP address translation:

```
# host -a polsl.pl
# host -a 157.158.3.28
# nslookup polsl.pl
```

Show the IP address of the domain: google.com, www.polsl.pl, polsl.pl, platforma.polsl.pl. Which DNS servers did the answers come from? How does the DNS service work?

```
# host -a polsl.pl

Trying "polsl.pl"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34861
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;polsl.pl.                      IN      ANY

;; ANSWER SECTION:
polsl.pl.               2134    IN      AAAA    2a01:1dc:1:15::1:1
polsl.pl.               2134    IN      AAAA    2a01:1dc:0:3::1:1
polsl.pl.               2134    IN      AAAA    2a01:1dc:0:3::1:2
polsl.pl.               964     IN      A       157.158.3.1
polsl.pl.               964     IN      A       157.158.3.2
polsl.pl.               964     IN      A       157.158.99.73

Received 158 bytes from 10.255.255.254#53 in 159 ms
```

```
# host -a 157.158.3.28

Trying "28.3.158.157.in-addr.arpa"
Host 28.3.158.157.in-addr.arpa. not found: 3(NXDOMAIN)
Received 102 bytes from 10.255.255.254#53 in 290 ms
```

```
# nslookup polsl.pl

Server:         10.255.255.254
Address:        10.255.255.254#53

Non-authoritative answer:
Name:   polsl.pl
Address: 157.158.99.73
Name:   polsl.pl
Address: 157.158.3.1
Name:   polsl.pl
Address: 157.158.3.2
Name:   polsl.pl
Address: 2a01:1dc:1:15::1:1
Name:   polsl.pl
Address: 2a01:1dc:0:3::1:1
Name:   polsl.pl
Address: 2a01:1dc:0:3::1:2
```

IP Addresses for Domain Names:

- google.com:
  - `IPv4`: `142.250.180.110`
  - `IPv6`: `2a00:1450:4008:806::200e`
- www.polsl.pl (Resolved as polsl.pl):
  - `IPv4`: `157.158.3.1, 157.158.3.2, 157.158.99.73`
  - `IPv6`: `2a01:1dc:1:15::1:1`, `2a01:1dc:0:3::1:1`, `2a01:1dc:0:3::1:2`
- polsl.pl:
  - `IPv4`: `157.158.3.1`, `157.158.3.2`, `157.158.99.73`
  - `IPv6`: `2a01:1dc:1:15::1:1`, `2a01:1dc:0:3::1:1`, `2a01:1dc:0:3::1:2`
- platforma.polsl.pl:
  - `IPv4`: `157.158.108.131`

The DNS answers came from `10.255.255.254` (private IP address).

DNS translates domain names into IP addresses by querying DNS servers. If the local server doesn't have the information, it sends a request to higher-level servers until it receives the correct IP address.