

The Dude – Network management and monitoring software

Pauls Jukonis
MikroTik, Latvia

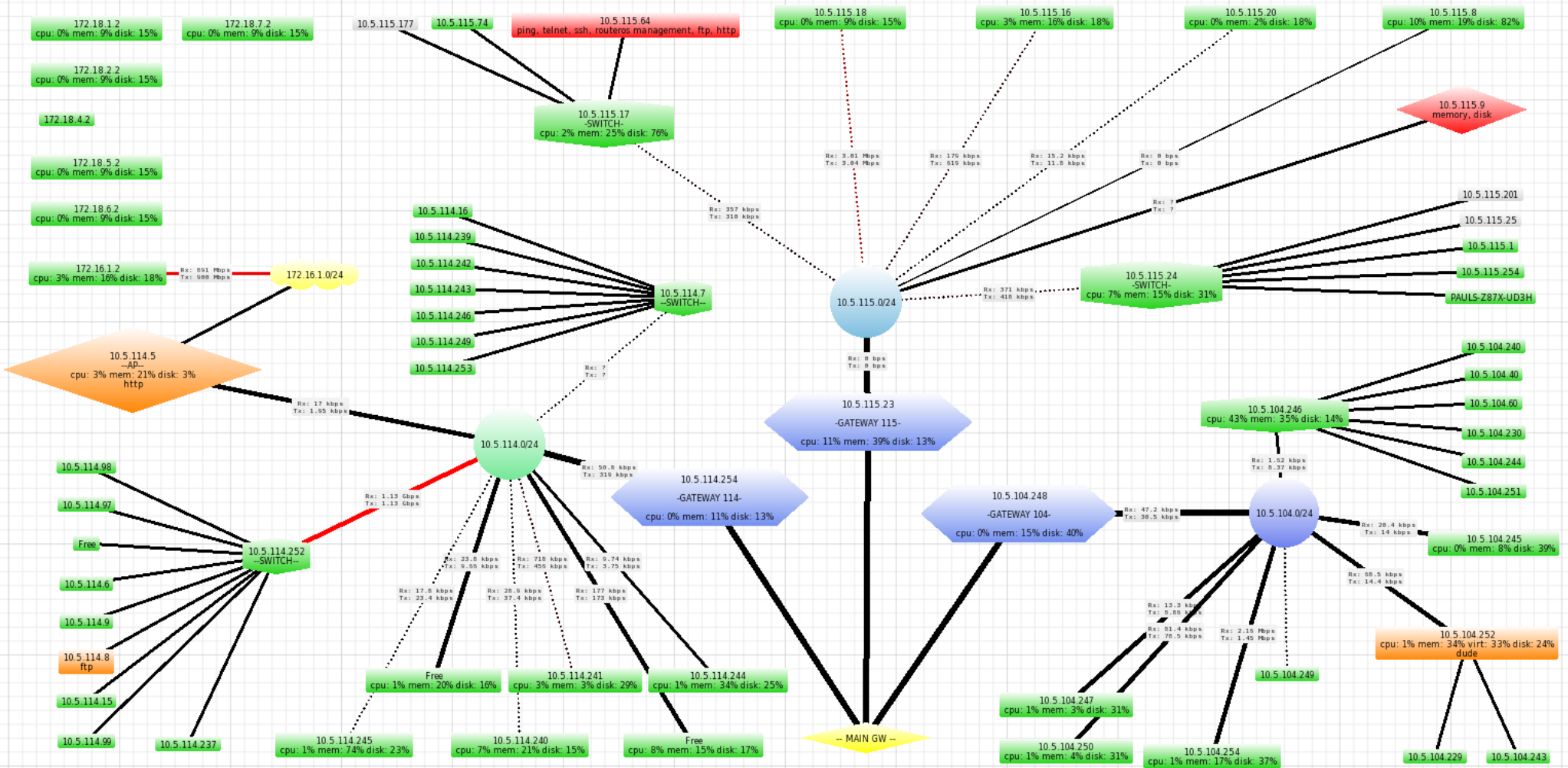
MUM Vietnam
April 2017

Overview

- The Dude - First steps
- The Dude Tools
- The Dude Notifications
- CHR overview

The Dude

One of the most powerful free network monitors



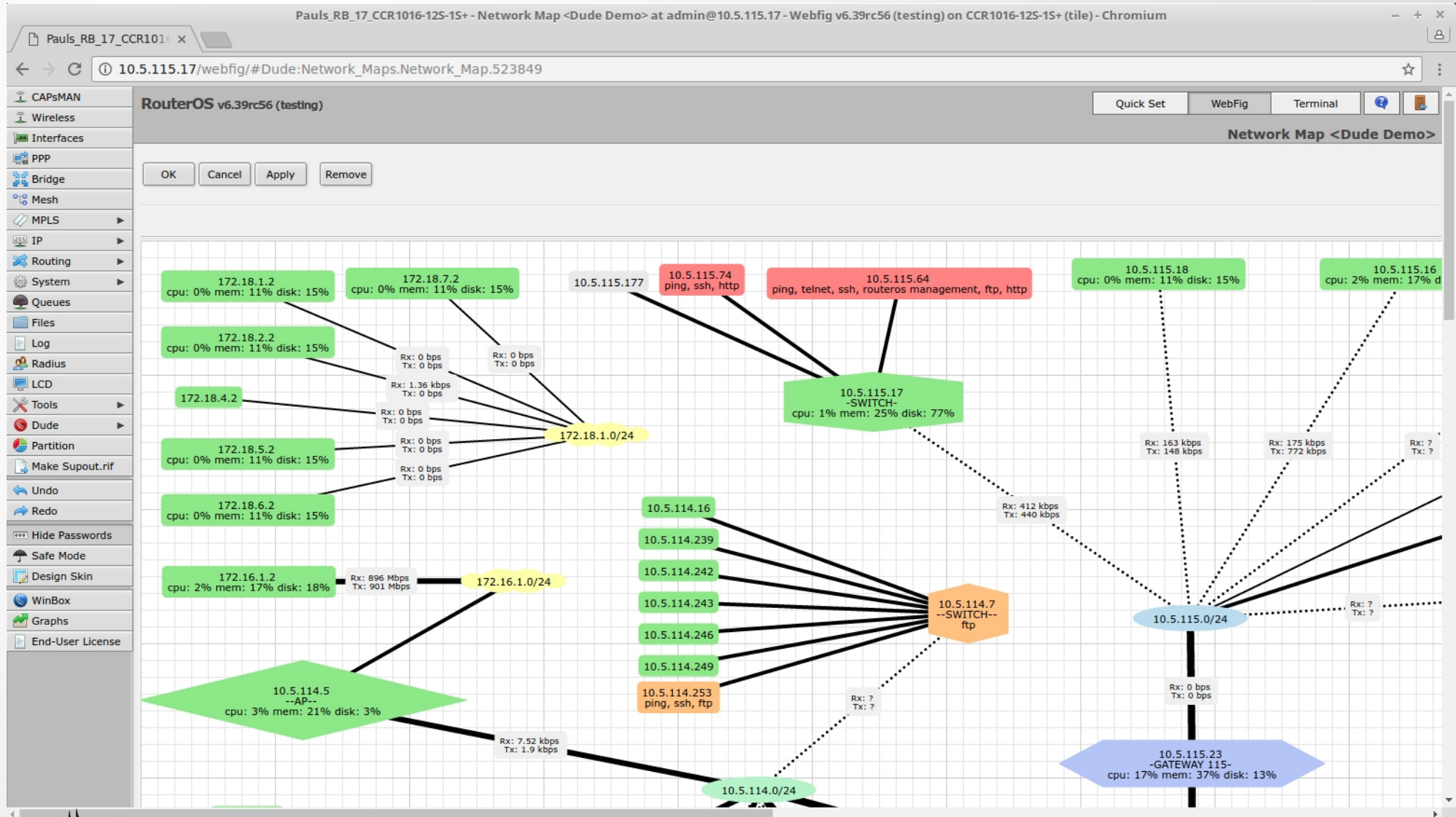
The Dude development

- First version released in 2006
- 4.0beta3 was the latest version
- Development renewed in 2016
- The Dude is integrated back to RouterOS package

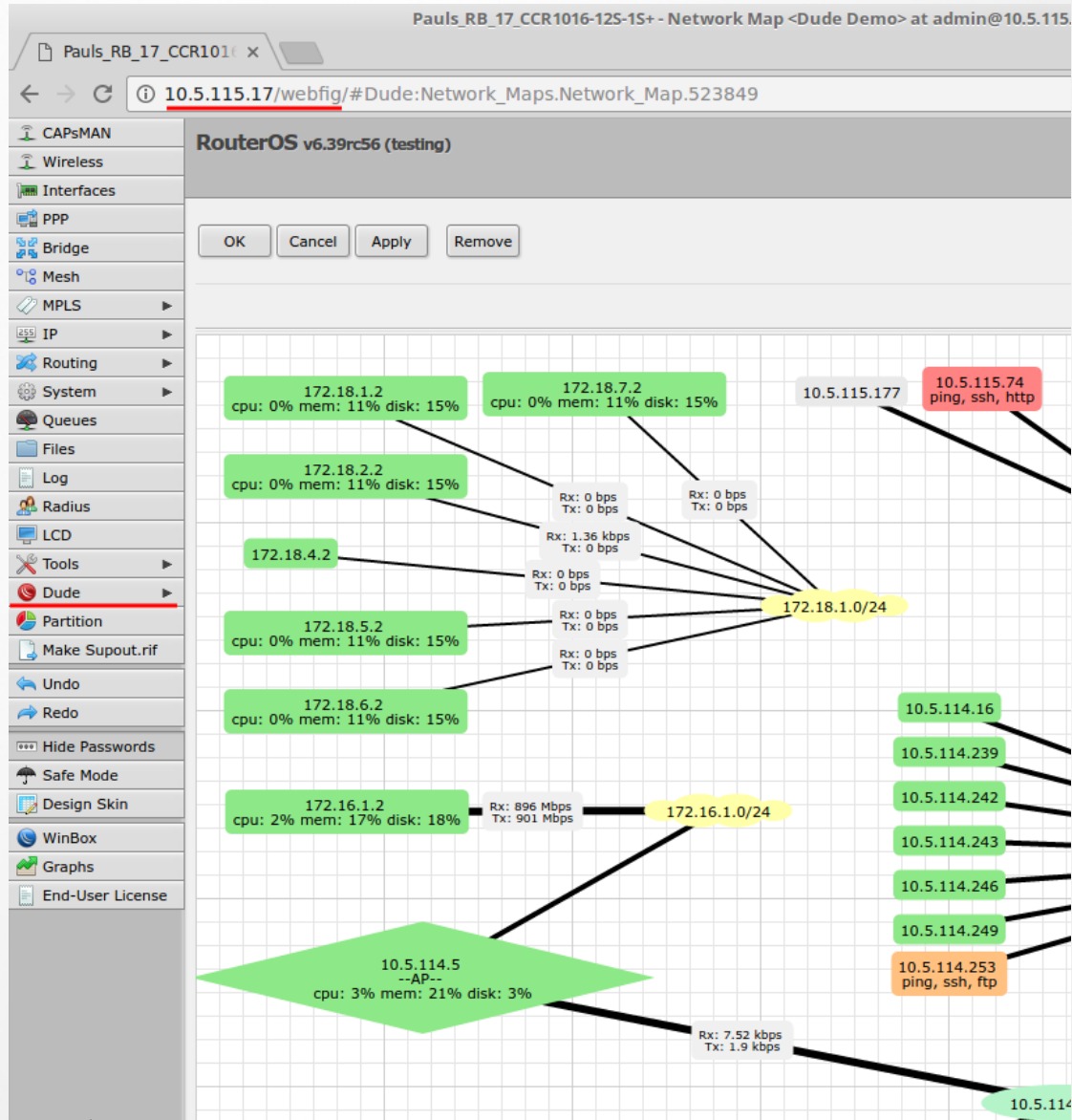
The Dude v6.x changes

The Dude server	The Dude client
ALL CCR (TILE)	Windows
CHR / x86	Webfig
RB3011/1100x4 (ARM)	Winbox
RB hEX (MMIPS)	CLI In development...

The Dude web interface



The Dude web interface



- Provides web access from any browser
- No installation required
- Provides configuration options
- Accessible at **<ServerIP>/webfig/**
- Starting from RouterOS 6.38

Prepare a server...

Install The Dude server:

- 1) Download The Dude server package (mikrotik.com/download)
- 2) Upload it to the RouterOS file system
- 3) Reboot RouterOS

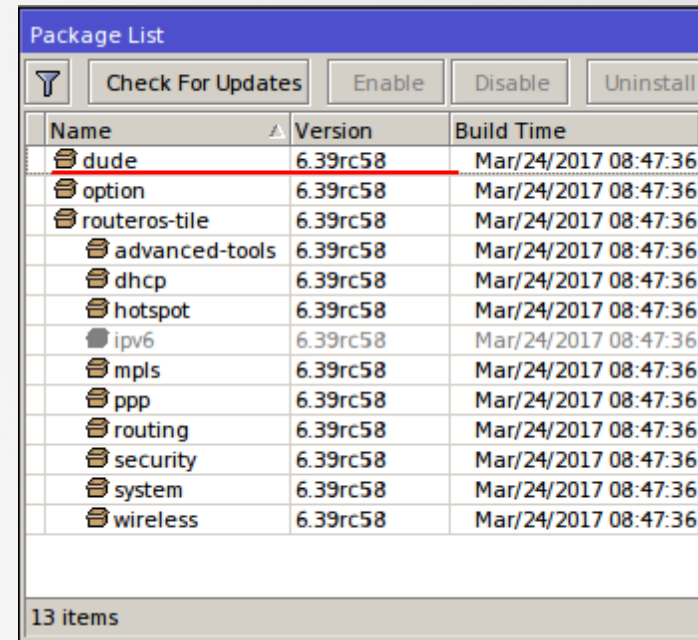
Enable The Dude server:

- 1) Specify directory where the database will be stored

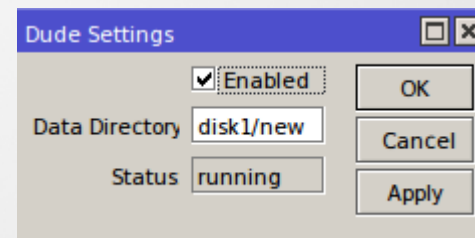
```
/dude set data-directory= disk1/new
```

- 2) Enable The Dude

```
/dude set enabled=yes
```



Name	Version	Build Time
dude	6.39rc58	Mar/24/2017 08:47:36
option	6.39rc58	Mar/24/2017 08:47:36
routeros-tile	6.39rc58	Mar/24/2017 08:47:36
advanced-tools	6.39rc58	Mar/24/2017 08:47:36
dhcp	6.39rc58	Mar/24/2017 08:47:36
hotspot	6.39rc58	Mar/24/2017 08:47:36
ipv6	6.39rc58	Mar/24/2017 08:47:36
mpls	6.39rc58	Mar/24/2017 08:47:36
ppp	6.39rc58	Mar/24/2017 08:47:36
routing	6.39rc58	Mar/24/2017 08:47:36
security	6.39rc58	Mar/24/2017 08:47:36
system	6.39rc58	Mar/24/2017 08:47:36
wireless	6.39rc58	Mar/24/2017 08:47:36



Recommendations for a server...

File system:

It's suggested to use **ext3** type file system, it's more reliable as a The Dude storage file system

Format drive using command:

```
/disk format-drive disk1 file-system=ext3
```

File storage:

If custom files are used, keep them in "**(data-path)/files/**" directory

Backups:

Make sure that regular database backups are made

Disk drive:

Always use industrial grade microSD cards and USB flashes as they have longer life cycle

If possible use USB instead of micro SD card, this will give you better read/write performance

Prepare a client...

Install The Dude client:

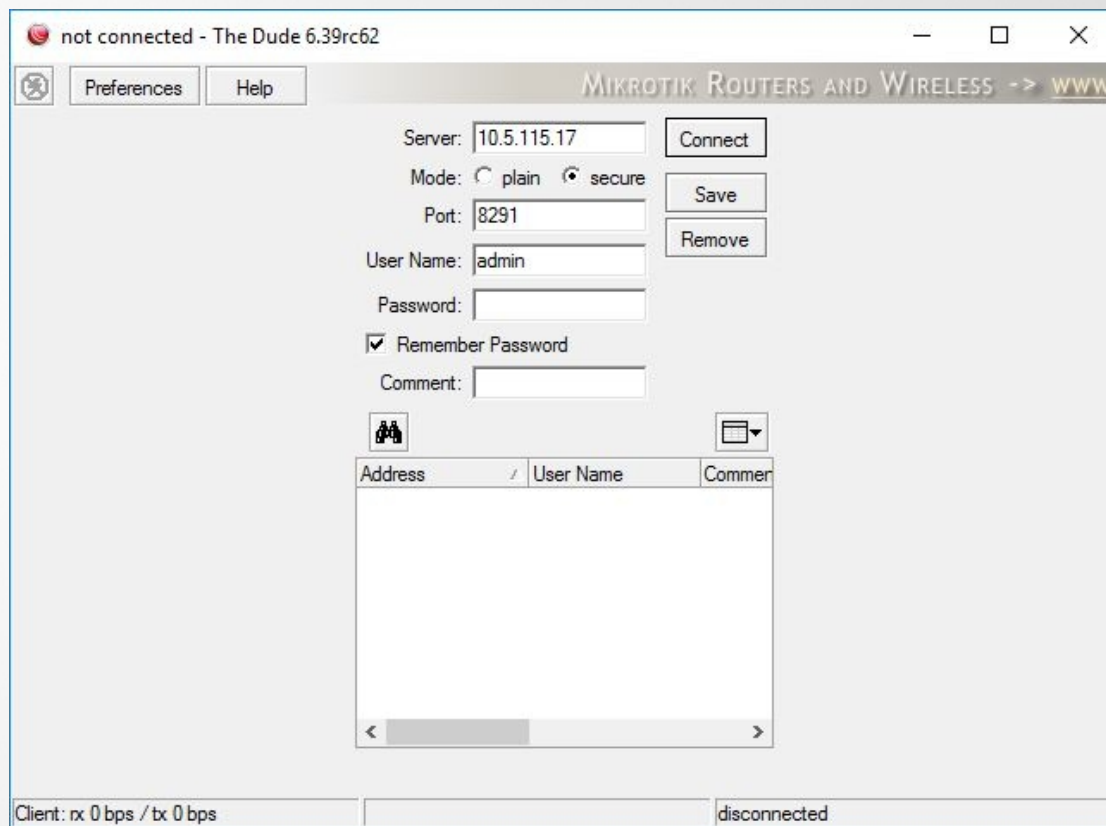
- 1) Download The Dude client
- 2) Install
- 3) Access the server

Use Webfig interface:

- 1) <Server IP>/webfig/

Use Winbox interface:

- 1) Winbox tool -> Dude



- The Dude login uses RouterOS users
- To log in the Dude server, dude policy is required
- The Dude client uses default Winbox port - 8291

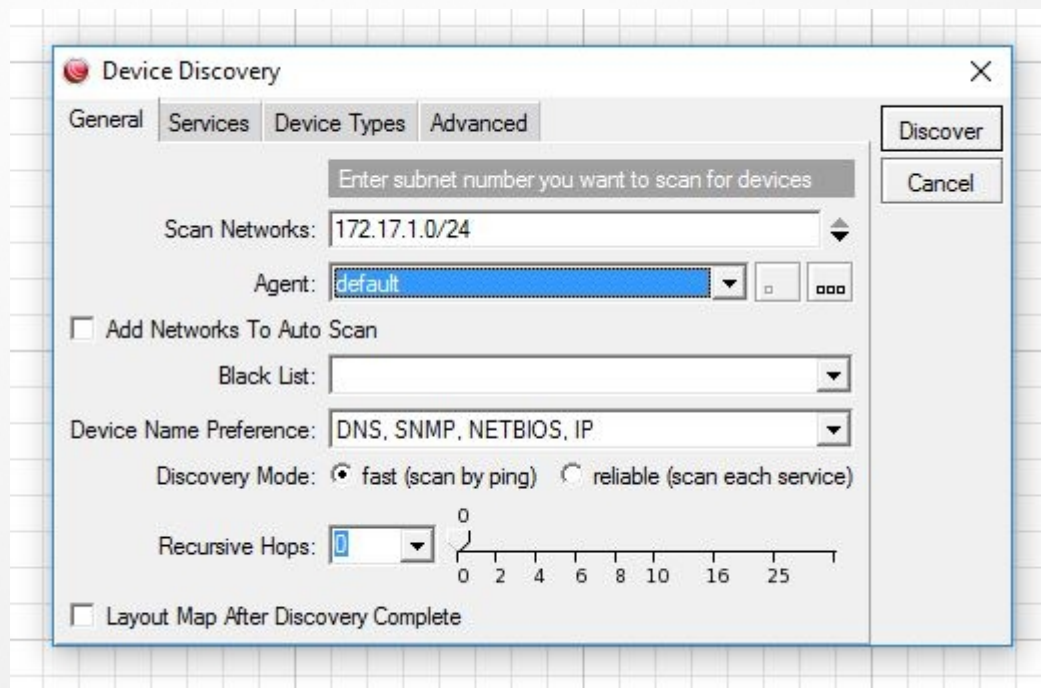
Discover your network

Discovery features:

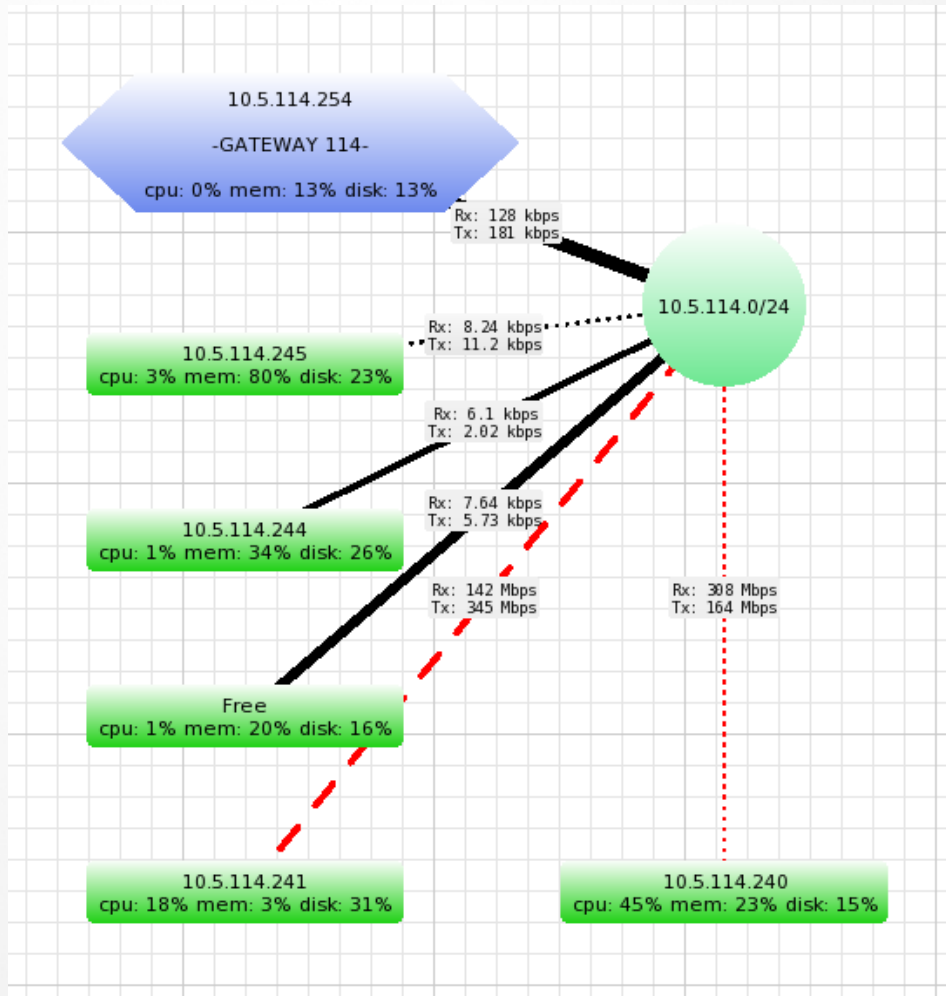
- Automatically scan specified network
- Automatically layout map after scan
- Can use agents to scan networks that are not directly accessible to server

Discovered services:

- ICMP (ping)
- DNS
- SSH
- Telnet
- HTTP
- FTP
- RouterOS management
- Other...



Edit network map



- Add new networks
- Add new devices
- Add submaps
- Add customised backgrounds (Maps / Plannings)
- Add customised info on the labels (Text / SNMP / RouterOS data)
- Edit layout
- Edit appearance of devices / links (shapes, colours)
- Edit device / link speed warnings

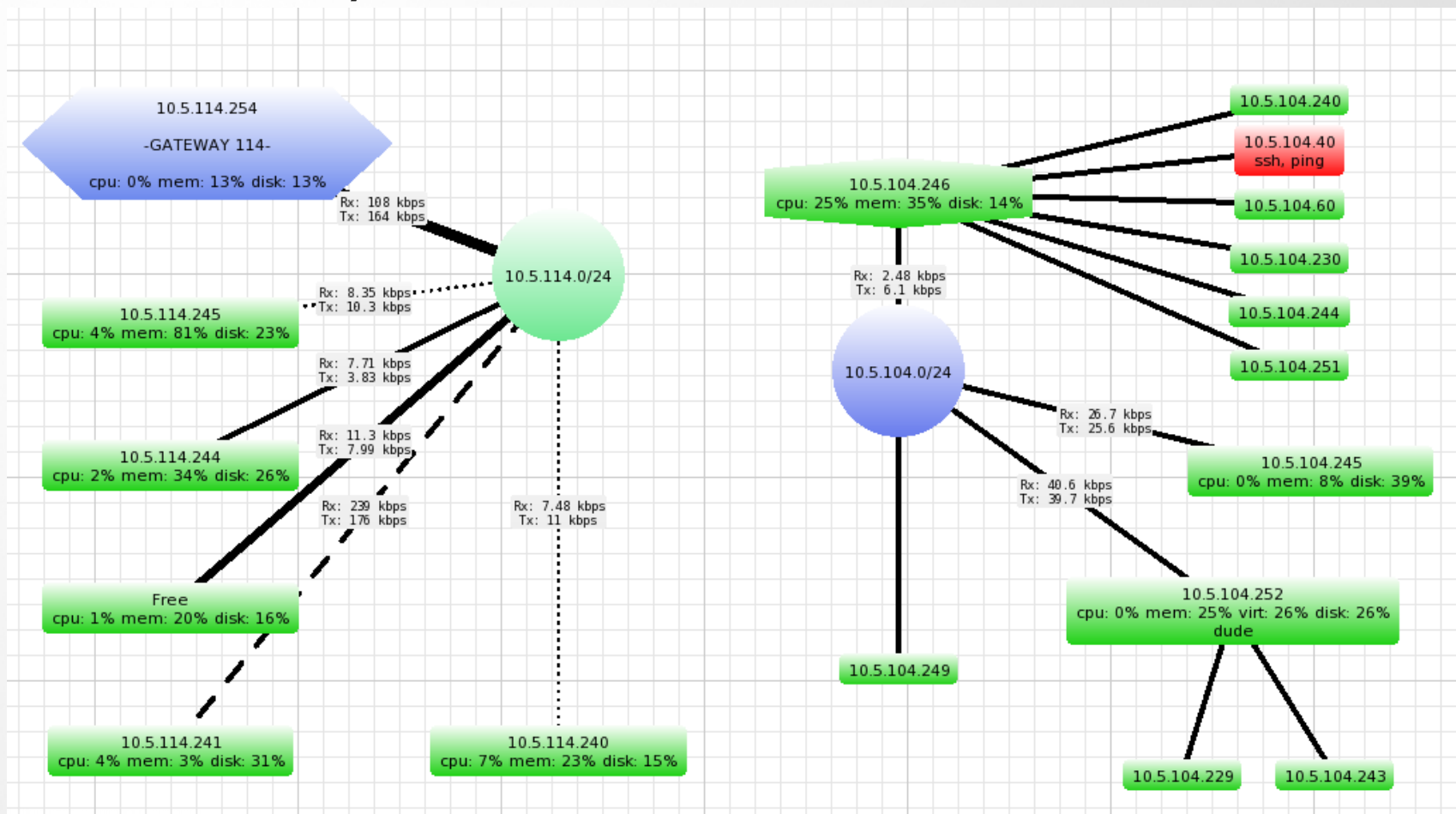
Use custom backgrounds

The Dude allows to use custom images in background, for example maps, that will create more understandable view of your network



Use agents in The Dude

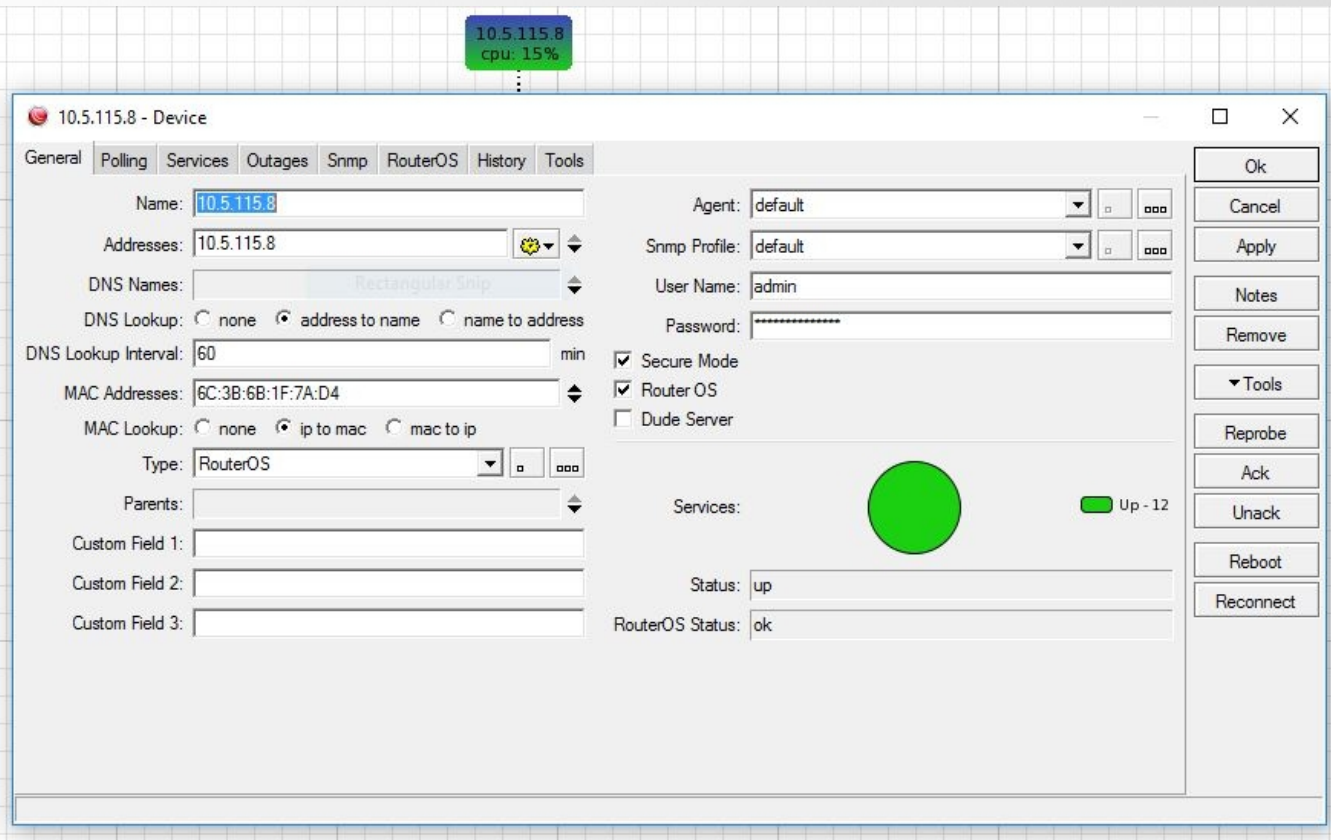
Agents allow to monitor remote sites which are not directly accessible from The Dude server



Device

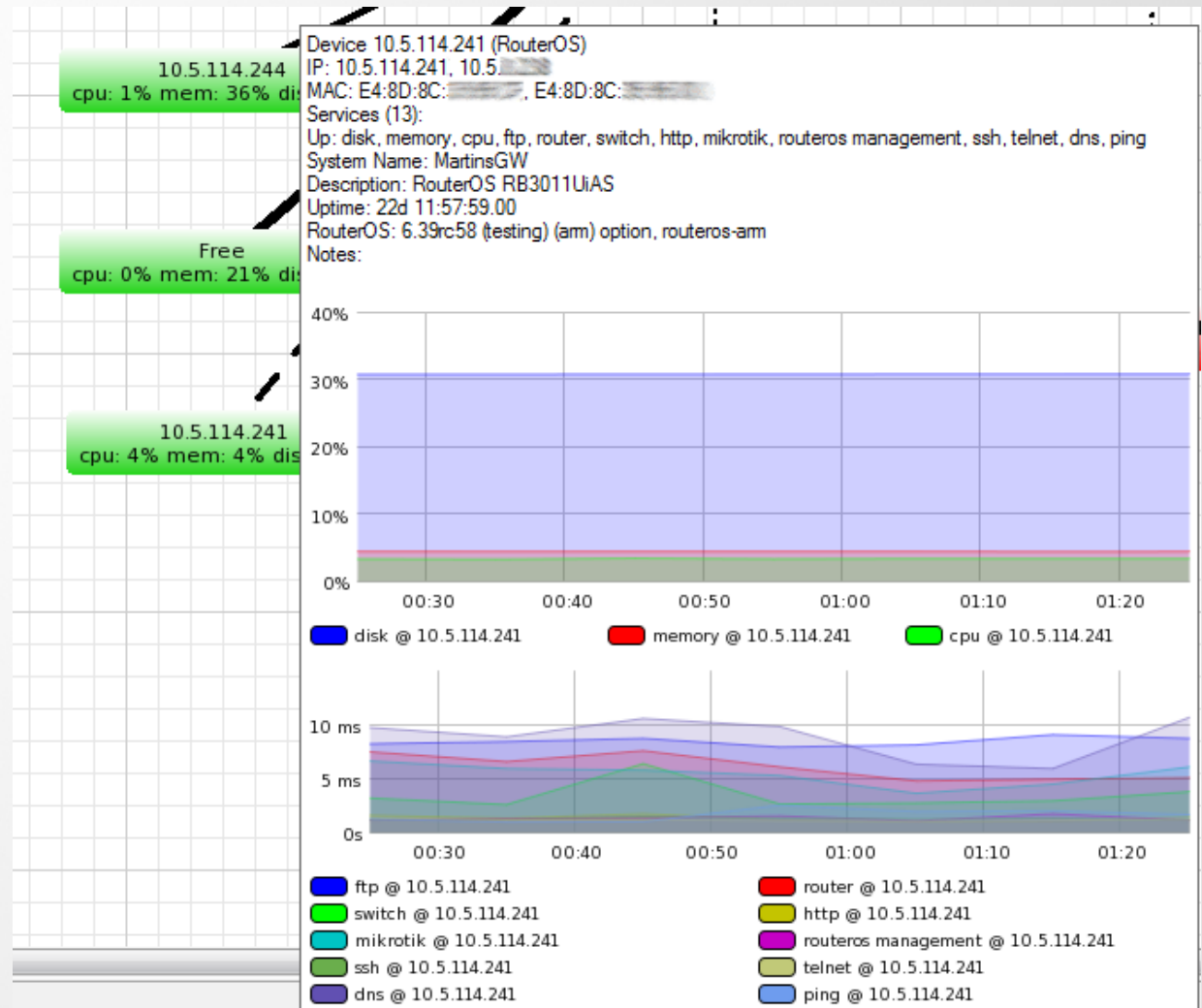
Device is network object which is monitored by The Dude server

- Stores general info of a device
- Shows all available SNMP data (Interface list, traffic, route list, ip list, arp table, etc.)
- Shows additional RouterOS data
- Stores all graphs of service latencies
- Stores custom data graphs (CPU / Disk / Memory usage)
- Stores all service outage history
- Editable services which will be monitored



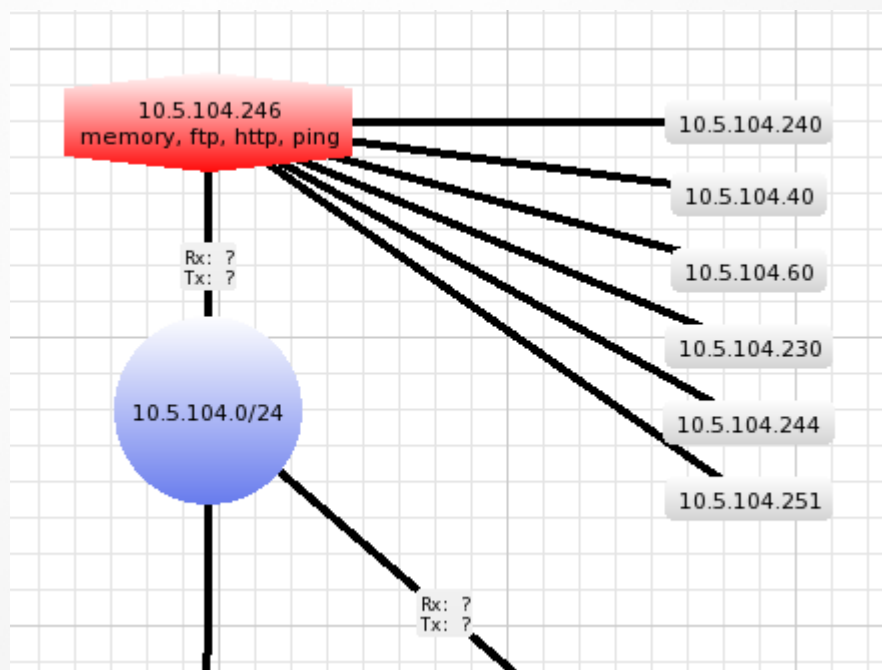
Device

Hovering your mouse cursor over a device item will show you a quick graph of the service availability



Device tree structure

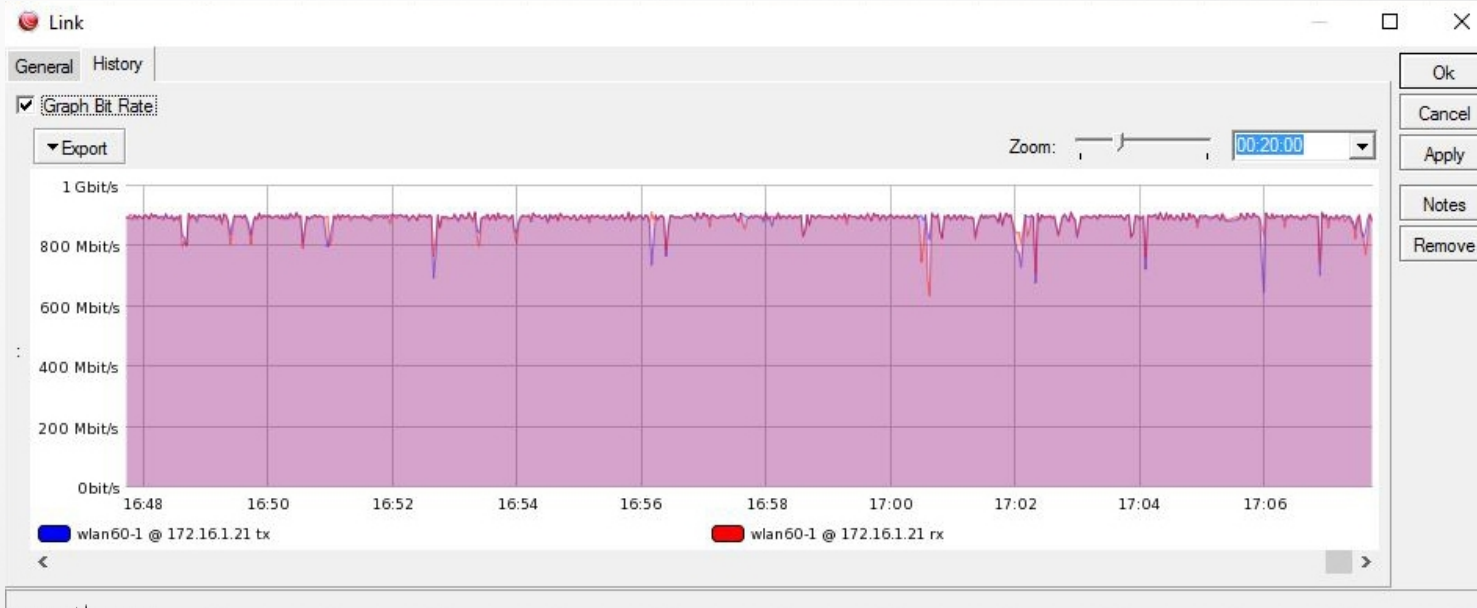
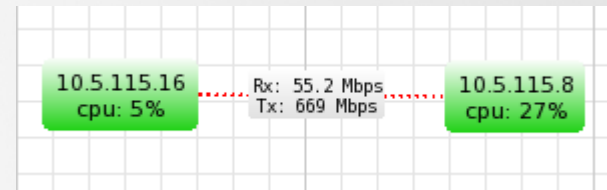
Device tree structure allows to create logical **child - parent** hierarchy in the network, which later can be used for advanced notification delivering.



- Notifications will be sent only about parent devices
- If parent is down, child status will stay "Unknown"
- Easy configurable – select multiple devices, set one parent
- Helps to avoid unnecessary notification messages

Links

- Links represent a connection between devices/networks
- Automatically stores graphical history of a data transmit
- Indicates link usage by changing the link colour
- Links speeds can be monitored using SNMP or RouterOS management.
- Choose from predefined link types (shapes / speeds)

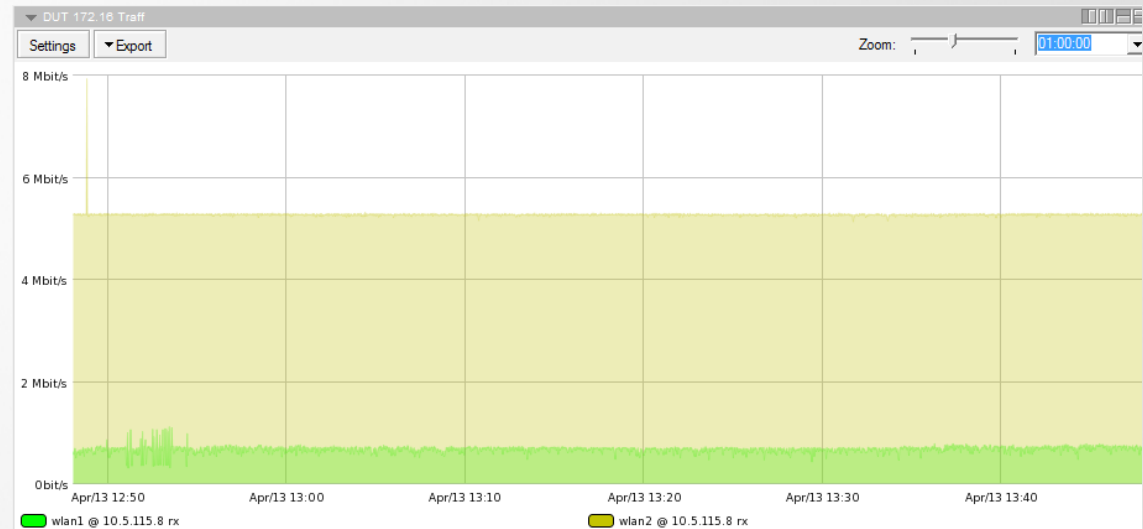


Charts

Charts allows to create a graphical representation of data stored by The Dude server

Charts can be made from:

- Automatically stored data, for example link speeds and device service latencies
- Manually created data sources (SNMP / RouterOS data)



- Unlimited number of charts can be created in The Dude server
 - Each chart can have multiple data sources

Charts

To create custom graphs, first you need to create new **data source** from SNMP or function command output

Get data using SNMP:

- Check available OIDs

```
[admin@Pauls_RB_17_CCR1016-12S-1S+] > system health print oid
  active-fan: .1.3.6.1.4.1.14988.1.1.3.9.0
    voltage: .1.3.6.1.4.1.14988.1.1.3.8.0
  temperature: .1.3.6.1.4.1.14988.1.1.3.10.0
processor-temperature: .1.3.6.1.4.1.14988.1.1.3.11.0
  current: .1.3.6.1.4.1.14988.1.1.3.13.0
  fan-speed: .1.3.6.1.4.1.14988.1.1.3.17.0
  fan-speed2: .1.3.6.1.4.1.14988.1.1.3.18.0
power-consumption: .1.3.6.1.4.1.14988.1.1.3.12.0
  psu1-state: .1.3.6.1.4.1.14988.1.1.3.15.0
  psu2-state: .1.3.6.1.4.1.14988.1.1.3.16.0
```

- Use OID code to store values

```
oid("1.3.6.1.4.1.14988.1.1.3.10.0")
```

The screenshot shows a configuration form for a data source. The fields are: Name: Temperature; Unit: (empty); Device: 10.5.115.16; Code: oid("1.3.6.1.4.1.14988.1.1.3.10.0").

Get data using functions:

- Function `ros_command` allows to capture output from RouterOS device and use it as a value for charts

```
ros_command("/interface wireless
registration-table print count-only
where interface=wlan1")
```

The screenshot shows a configuration form for a data source. The fields are: Name: Data Source; Unit: (empty); Device: 10.5.115.16; Code: ros_command("/interface wireless registration-table print count-only where interface=wlan1"); Interval: 00:00:30. A graph preview shows a 30s interval.

Files

The Dude allows to use custom files:

- Images – Background maps, device labels
- MIBS – To get custom SNMP data from devices
- RouterOS packages – To perform auto upgrade on monitored devices

Files can be managed using:

- Winbox / Webfig / FTP / SMB / Tool fetch

File structure:

- The Dude default files are stored in **(data-path)/files/default** and have read-only permission
- Custom files should be stored in **(data-path)/files/** directory

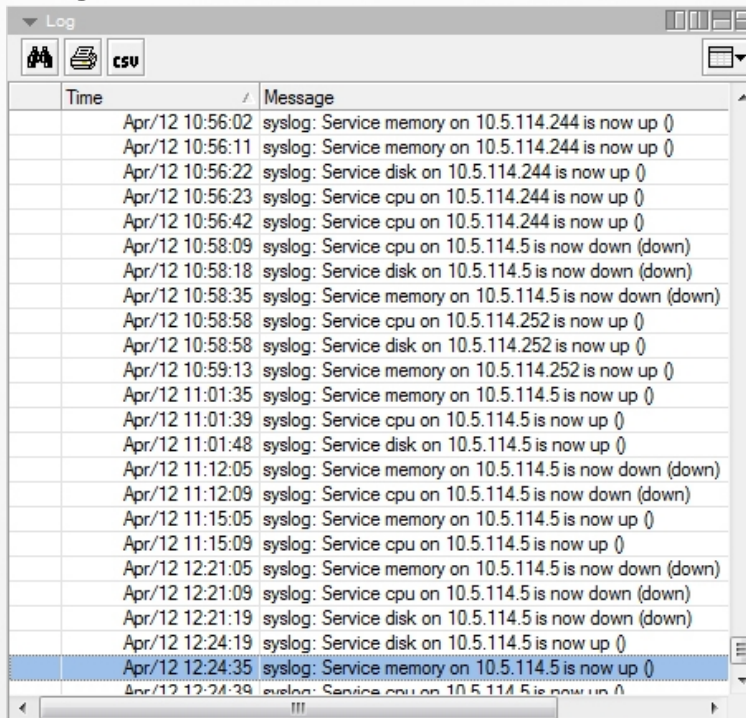
Backups:

- Backups also include files

Logs

- Logs store records of device status history
- The Dude can be used as a syslog server
- All logs can be viewed from the RouterOS and The Dude server

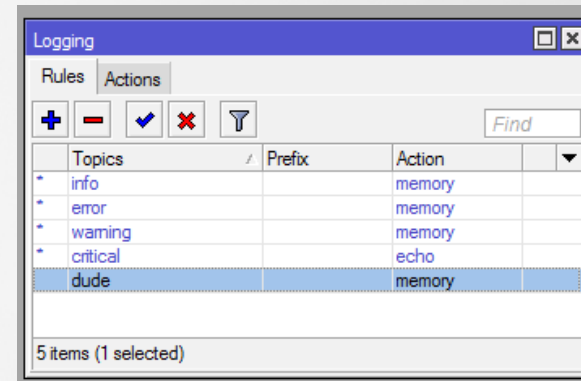
Logs in The Dude



The screenshot shows a window titled 'Log' with a table of syslog messages. The table has two columns: 'Time' and 'Message'. The messages are sorted chronologically and include status updates for various services like memory, disk, and CPU on different IP addresses.

Time	Message
Apr/12 10:56:02	syslog: Service memory on 10.5.114.244 is now up ()
Apr/12 10:56:11	syslog: Service memory on 10.5.114.244 is now up ()
Apr/12 10:56:22	syslog: Service disk on 10.5.114.244 is now up ()
Apr/12 10:56:23	syslog: Service cpu on 10.5.114.244 is now up ()
Apr/12 10:56:42	syslog: Service cpu on 10.5.114.244 is now up ()
Apr/12 10:58:09	syslog: Service cpu on 10.5.114.5 is now down (down)
Apr/12 10:58:18	syslog: Service disk on 10.5.114.5 is now down (down)
Apr/12 10:58:35	syslog: Service memory on 10.5.114.5 is now down (down)
Apr/12 10:58:58	syslog: Service cpu on 10.5.114.252 is now up ()
Apr/12 10:58:58	syslog: Service disk on 10.5.114.252 is now up ()
Apr/12 10:59:13	syslog: Service memory on 10.5.114.252 is now up ()
Apr/12 11:01:35	syslog: Service memory on 10.5.114.5 is now up ()
Apr/12 11:01:39	syslog: Service cpu on 10.5.114.5 is now up ()
Apr/12 11:01:48	syslog: Service disk on 10.5.114.5 is now up ()
Apr/12 11:12:05	syslog: Service memory on 10.5.114.5 is now down (down)
Apr/12 11:12:09	syslog: Service cpu on 10.5.114.5 is now down (down)
Apr/12 11:15:05	syslog: Service memory on 10.5.114.5 is now up ()
Apr/12 11:15:09	syslog: Service cpu on 10.5.114.5 is now up ()
Apr/12 12:21:05	syslog: Service memory on 10.5.114.5 is now down (down)
Apr/12 12:21:09	syslog: Service cpu on 10.5.114.5 is now down (down)
Apr/12 12:21:19	syslog: Service disk on 10.5.114.5 is now down (down)
Apr/12 12:24:19	syslog: Service disk on 10.5.114.5 is now up ()
Apr/12 12:24:35	syslog: Service memory on 10.5.114.5 is now up ()
Apr/12 12:24:39	syslog: Service cpu on 10.5.114.5 is now up ()

Enable logs in RouterOS



The screenshot shows the 'Logging' configuration window in RouterOS. It has tabs for 'Rules' and 'Actions'. Below the tabs are several icons for adding, removing, enabling, disabling, and filtering rules. A table lists the configured logging rules with columns for 'Topics', 'Prefix', and 'Action'. The 'dude' rule is selected.

Topics	Prefix	Action
info		memory
error		memory
warning		memory
critical		echo
dude		memory

5 items (1 selected)

Enable RouterOS logs from CLI

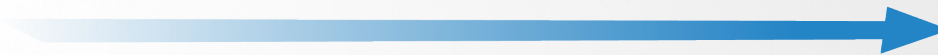
```
> system logging add topics=dude action=memory
```

The Dude Tools

Tools

The Dude Tools

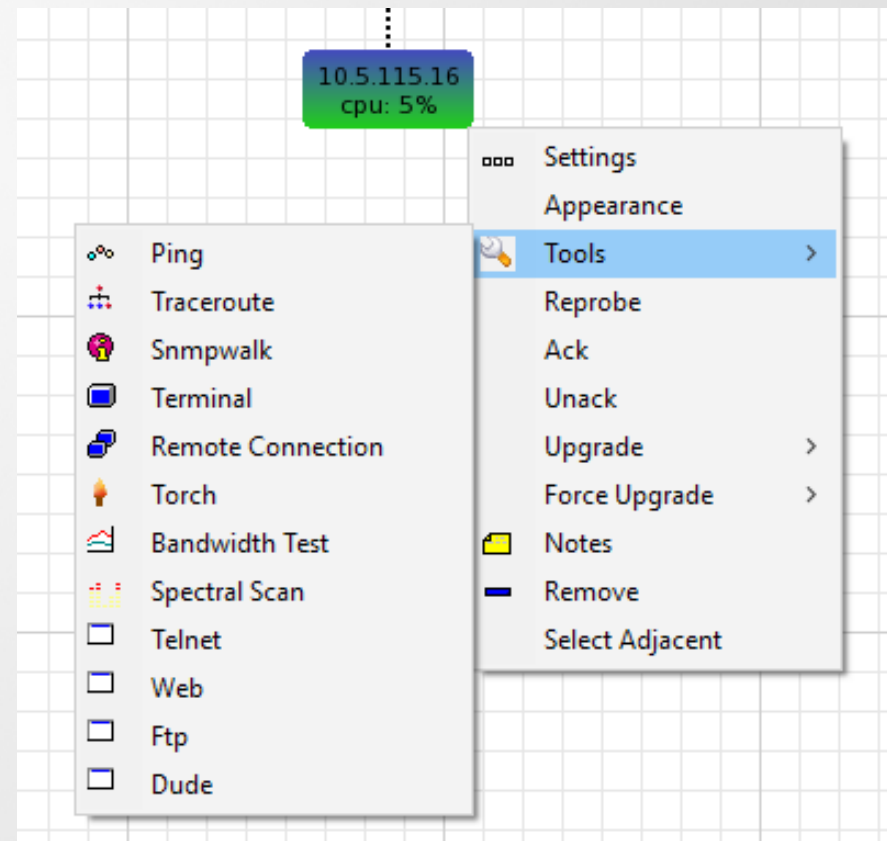
Variables
(Device) (User) (Password)



Arguments
192.168.1.1 admin nopass

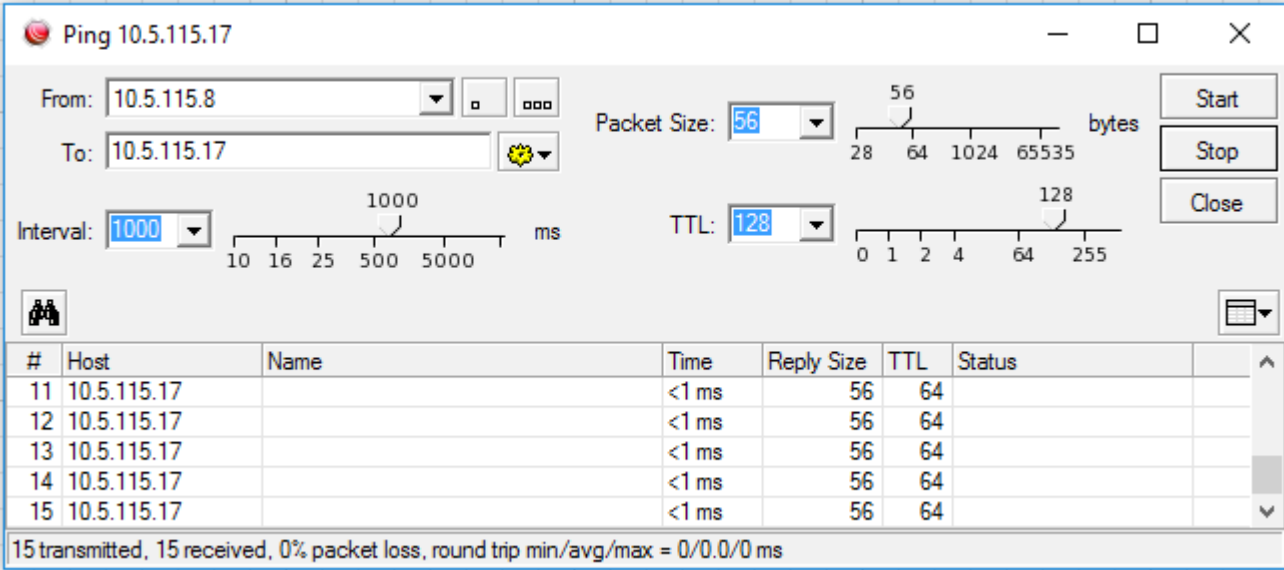
Remote program

- Tools manage utilities and programs that can be executed from The Dude
- Tools can use variables from monitored devices, and pass them to other programs
- The Dude allows to add your own custom tools
- Custom tools are added by specifying a command or path to the program



Ping

- Allows to quickly check response time between devices
- Allows to change some ICMP packet settings: packet size, TTL, speed
- Ping is sent from a selected device



The screenshot shows the 'Ping 10.5.115.17' window in Mikrotik WinBox. The 'From' field is set to 10.5.115.8 and the 'To' field is set to 10.5.115.17. The 'Packet Size' is set to 56 bytes and the 'TTL' is set to 128. The 'Interval' is set to 1000 ms. The window displays a table of ping results for 15 consecutive attempts, all showing a response time of <1 ms, a reply size of 56 bytes, and a TTL of 64. The status for all attempts is successful. A summary at the bottom indicates 15 transmitted, 15 received, 0% packet loss, and a round trip time of 0/0.0/0 ms.

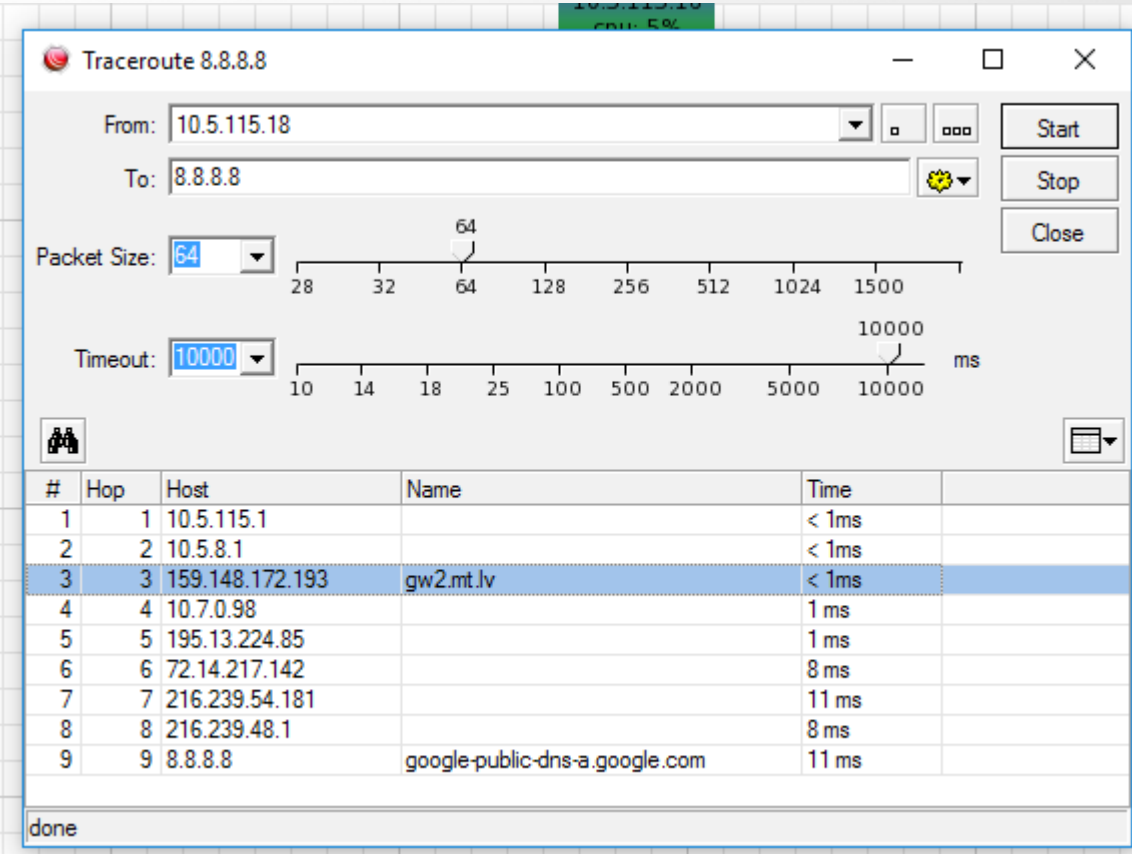
#	Host	Name	Time	Reply Size	TTL	Status
11	10.5.115.17		<1 ms	56	64	
12	10.5.115.17		<1 ms	56	64	
13	10.5.115.17		<1 ms	56	64	
14	10.5.115.17		<1 ms	56	64	
15	10.5.115.17		<1 ms	56	64	

15 transmitted, 15 received, 0% packet loss, round trip min/avg/max = 0/0.0/0 ms

Traceroute

Diagnostic tool for displaying a route and measure transit delays

- Uses ICMP to determine the path from selected device to specified host



The screenshot shows the Traceroute 8.8.8.8 window with the following configuration:

- From: 10.5.115.18
- To: 8.8.8.8
- Packet Size: 64
- Timeout: 10000

The results table is as follows:

#	Hop	Host	Name	Time
1	1	10.5.115.1		< 1ms
2	2	10.5.8.1		< 1ms
3	3	159.148.172.193	gw2.mt.lv	< 1ms
4	4	10.7.0.98		1 ms
5	5	195.13.224.85		1 ms
6	6	72.14.217.142		8 ms
7	7	216.239.54.181		11 ms
8	8	216.239.48.1		8 ms
9	9	8.8.8.8	google-public-dns-a.google.com	11 ms

done

SNMP walk

Use SNMP walk tool to overview the information available over SNMP

Snmp Walk 10.5.115.16

From: default To: 10.5.115.16 Profile: v1-public Type: all Timeout: 3000 ms Tries: 3

List Tree Table Module: all

Oid	Type	Value
iso.std.iso.8802.ieee802dot1.ieee802d...	octet string	Pauls_cnr
iso.std.iso.8802.ieee802dot1.ieee802d...	octet string	MikroTik Router...
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	20
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	20
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	1
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	2
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	3
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	4
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	5
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	6
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	7
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	8
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	9
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	10
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	11
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	12
iso.std.iso.8802.ieee802dot1.ieee802d...	integer	13

walking...

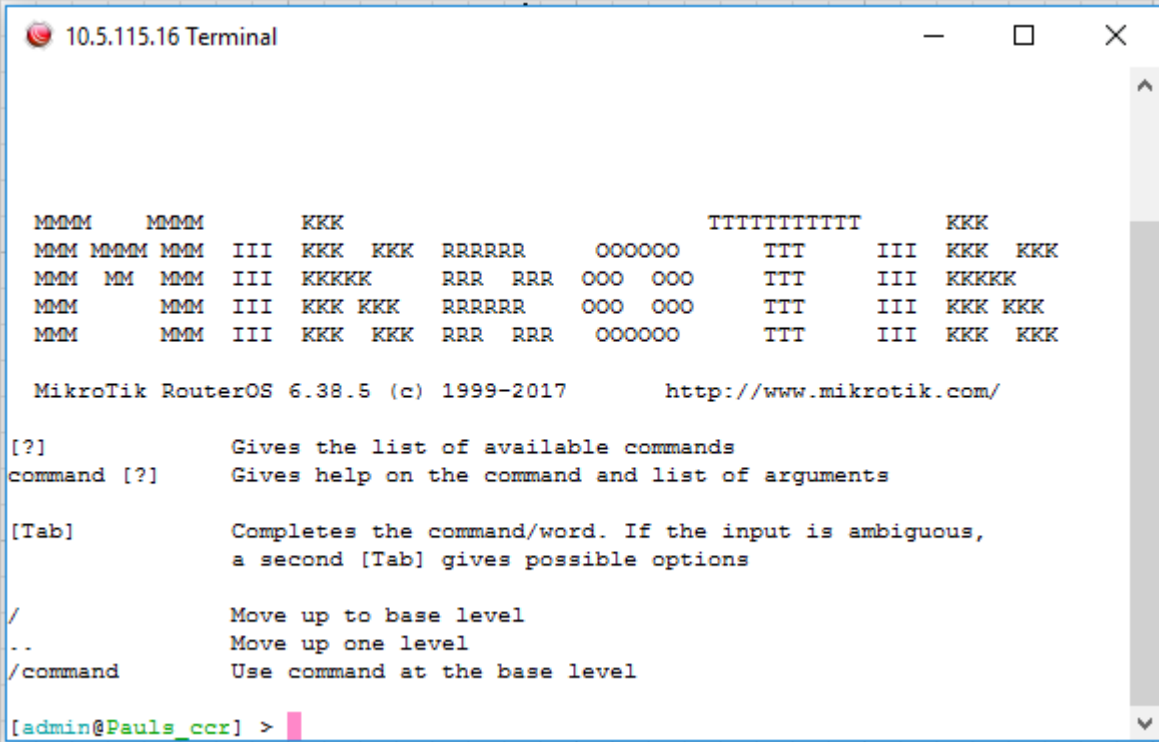
Returned results can be viewed in 3 ways:

- List – Shows OID, type and value
- Tree – Results are ordered in a tree look structure
- Table shows corresponding MIB module and a description

Each returned OID can be used to create a custom data source, for example to create a chart

Terminal

- Gives a quick way to login in selected device using terminal
- Tool opens a new CLI window that can be used to execute commands and receive responses



```
10.5.115.16 Terminal

MMMM  MMMM  KKK
MMM MMMM MMM III KKK KKK RRRRRR  OOOOOO  TTTTTTTTTT  KKK
MMM MM  MMM III KKKKK  RRR RRR  OOO OOO  TTT  III KKK KKK
MMM  MMM  III KKK KKK  RRRRRR  OOO OOO  TTT  III KKK KKK
MMM  MMM  III KKK KKK  RRR RRR  OOOOOO  TTT  III KKK KKK

MikroTik RouterOS 6.38.5 (c) 1999-2017      http://www.mikrotik.com/

[?]          Gives the list of available commands
command [?]  Gives help on the command and list of arguments

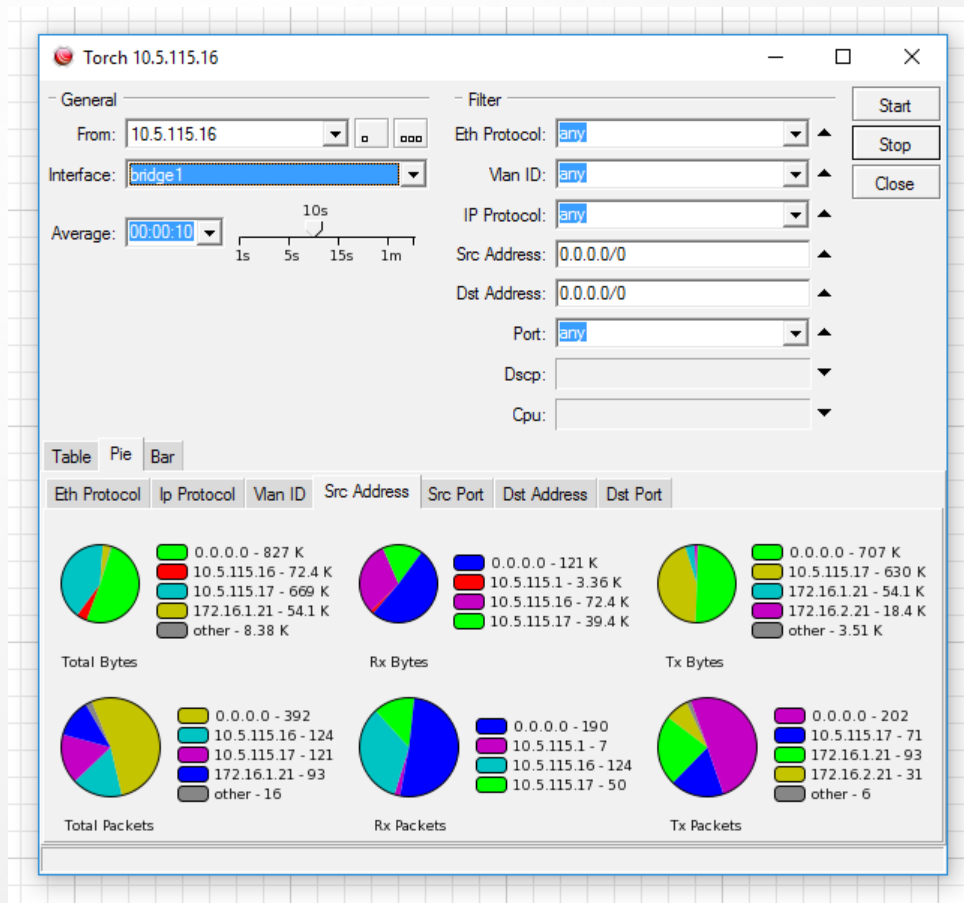
[Tab]       Completes the command/word. If the input is ambiguous,
            a second [Tab] gives possible options

/           Move up to base level
..         Move up one level
/command    Use command at the base level

[admin@Pauls_ccr] >
```

Torch

Tool torch is a real time traffic monitor which can be called directly from The Dude

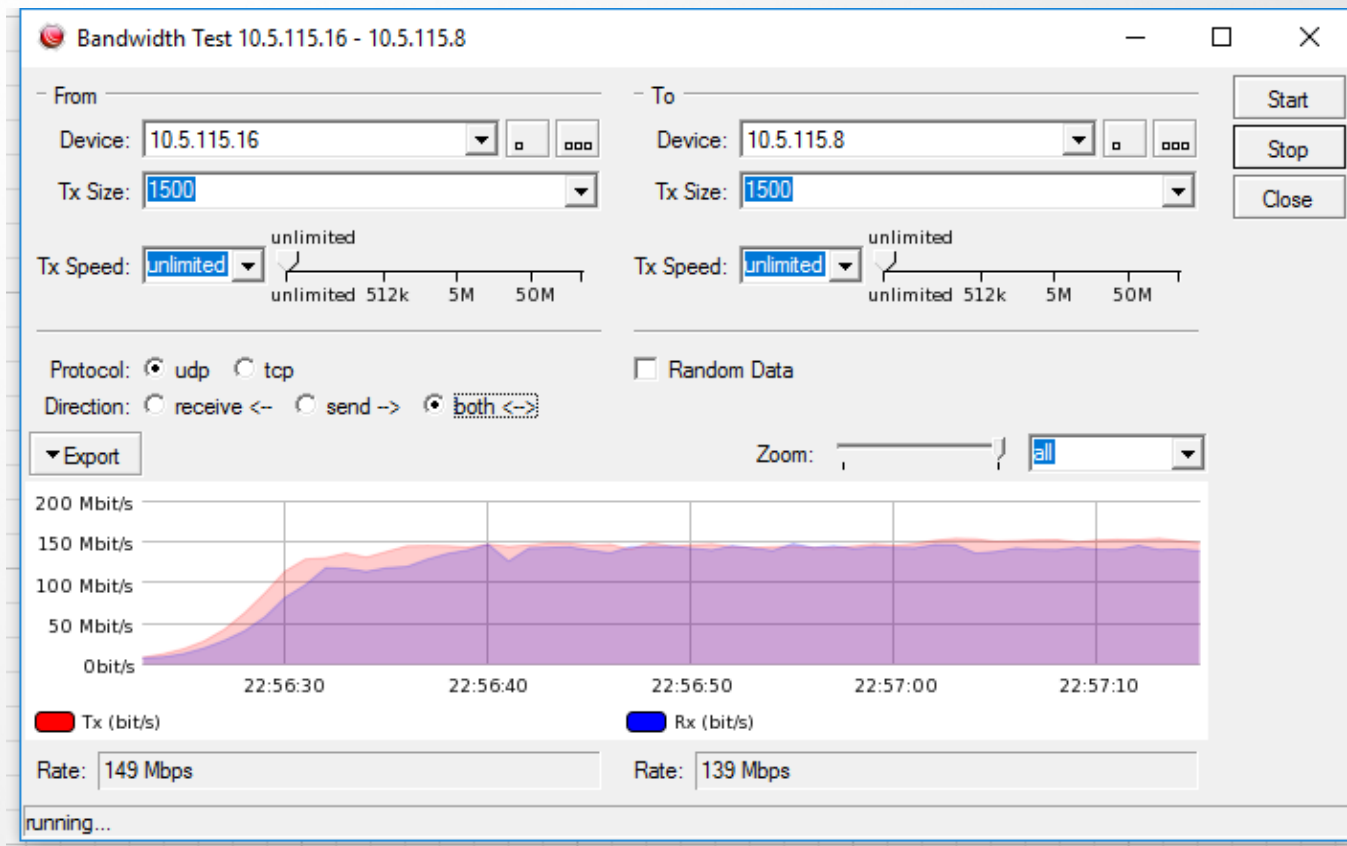


Returned results can be viewed in 3 ways:

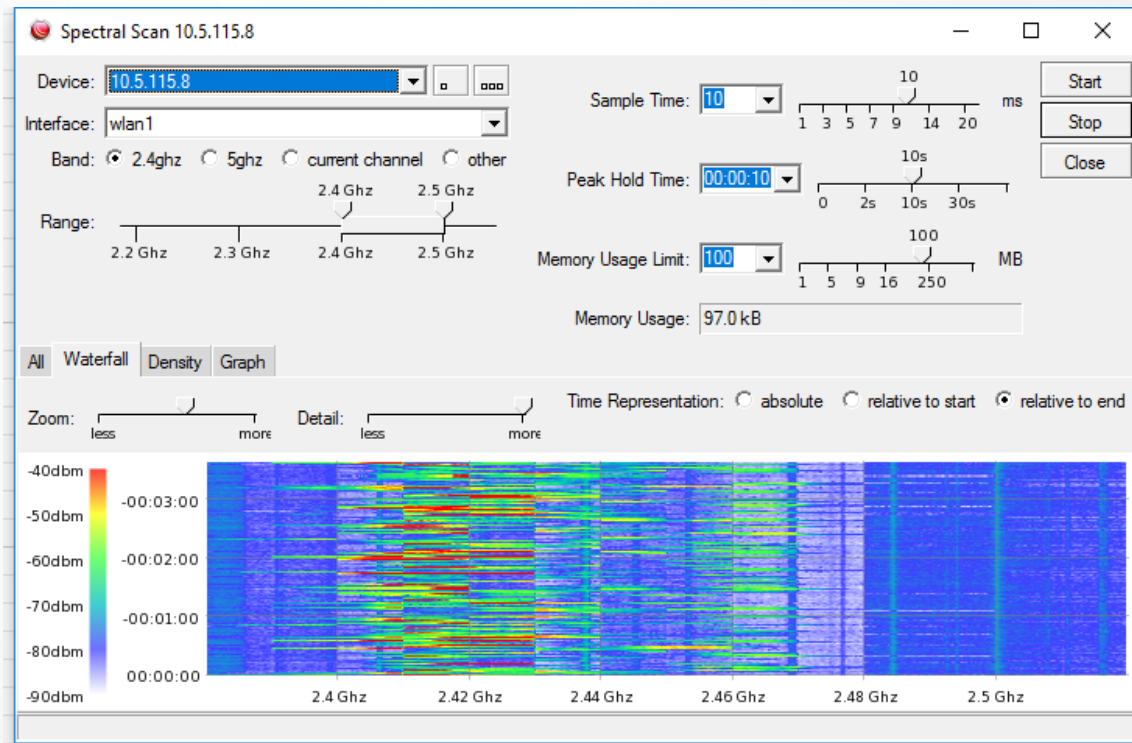
- Table – traffic flow shown in a list
- Pie - shows graphical representation of current traffic flow
- Bar – traffic shown in bar type graphical images

Bandwidth-test

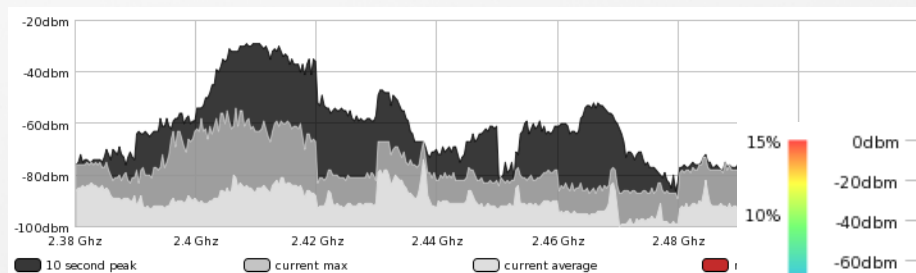
The Dude allows to quickly measure available bandwidth between devices with a single click



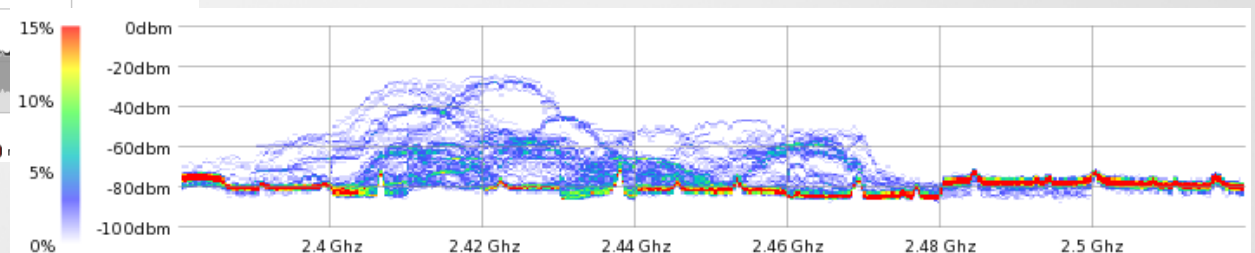
Spectral-scan



Spectral scan can scan all frequencies supported by the selected device



Different graphs are available



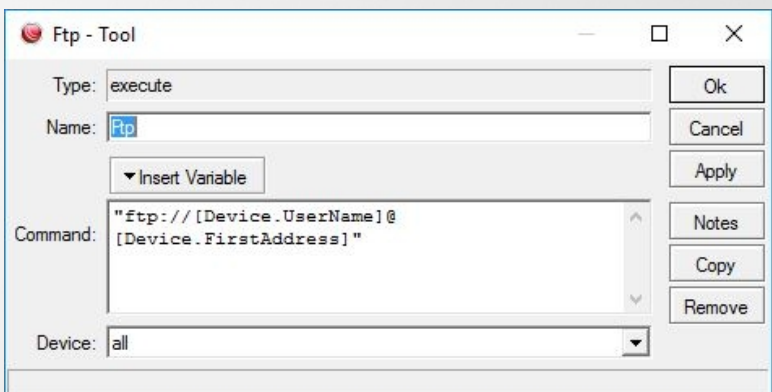
External tools

The Dude provides some predefined tools that will be executed on OS where The Dude client is installed on.

This provides a quick way to access selected device with a chosen service using one click.

- FTP – Opens FTP connection to selected device
- Web – Opens web browser with device URL
- Telnet – Connects using telnet protocol
- Dude – Opens a new Dude client if selected device is a Dude server

FTP command example



The screenshot shows a dialog box titled "Ftp - Tool". It has the following fields and controls:

- Type:** execute
- Name:** Ftp
- Command:** "ftp://[Device.UserName]@[Device.FirstAddress]"
- Device:** all

On the right side of the dialog, there are several buttons: Ok, Cancel, Apply, Notes, Copy, and Remove.

Custom tools

Winbox tool – requires path to executable Winbox file on your OS

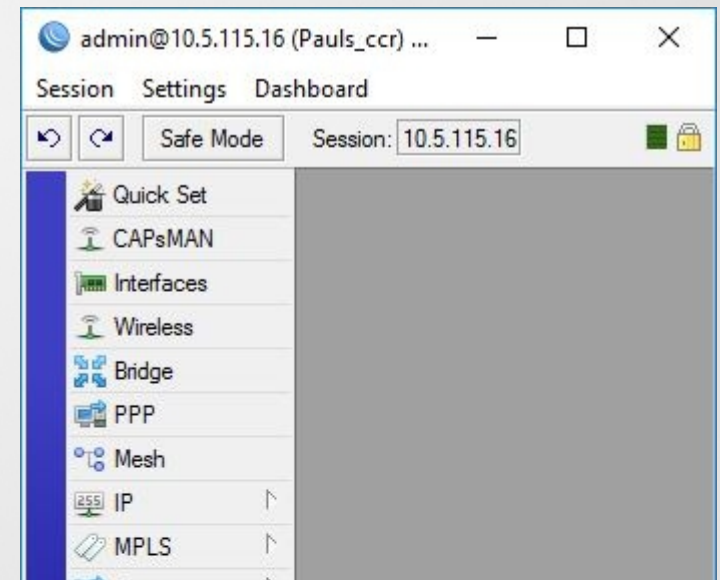
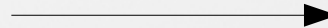
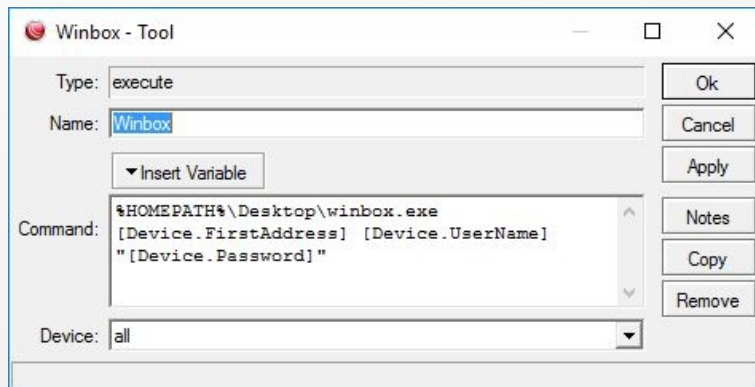
Hard path to winbox.exe

```
C:\Users\support\Desktop\winbox.exe (Device.FirstAddress) (Device.UserName) "(Device.Password)"
```

Path to winbox.exe using OS variables

```
%HOMEPATH%\Desktop\winbox.exe (Device.FirstAddress) (Device.UserName) "(Device.Password)"
```

This tool will open Winbox with already provided IP, Username and Password



Custom tools

Remote shutdown – When clicked, The Dude executes command that will shut down Windows device

Hard path to winbox.exe

```
c:\windows\system32\shutdown.exe -s -t 30 /m \\(Device.FirstAddress)
```

Connect with VNC – Command allows to connect to a remote PC via UltraVNC client

```
D:\programs\uvnc\vncviewer.exe -connect (Device.FirstAddress) -password (Device.Password)
```

Windows PSremote – Open Powershell window and use Windows PSremote feature from The Dude client to access Windows workstations where PSremoting is enabled.

```
D:\programs\uvnc\vncviewer.exe -connect (Device.FirstAddress) -password (Device.Password)
```

- The Dude supports any executable process/programm that can be started from OS CLI

The Dude Notifications

Notifications

Notifications in The Dude can trigger some action if a device or some service is not responding

Available predefined notification actions:

- **Beep** – Makes a beeping sound from the PC speaker of the server PC
- **Flash** – Flashes The Dude taskbar menu
- **Log** – Saves message in The Dude log file
- **Syslog** – Saves information to a remote Syslog server
- **Popup** – Opens a small notification window
- **Email** – Sends email to a specified recipient
- **Execute locally** – Runs command on the local Windows machine (where The Dude viewer runs)
- **Execute on server** – Execute RouterOS command/script on The Dude server
- **Sound** – Plays sound from specified sound file
- **Group** – Executes a group of actions
- **Speak** – Uses Windows speech to play digital voice message

Notifications

Notifications can be created with a few clicks

1. Choose device/s
2. Specify notification type
3. Specify services that will be monitored

10.5.115.8 - Device

General Polling Services Outages Snmp RouterOS History Tools

(Enabled)

Probe Interval: default (dropdown) | default 2s (slider)

Probe Timeout: default (dropdown) | default 2s (slider)

Probe Down Count: default (dropdown) | default 1 2 (slider)

Use Notifications

Notifications:

- Email-Notification
- Telegram notification
- beep
- flash
- log to events
- log to syslog
- popup

10.5.115.8 - Device

General Polling Services Outages Snmp RouterOS History Tools

Type	Problem
<input checked="" type="checkbox"/> cpu	
<input checked="" type="checkbox"/> disk	
<input checked="" type="checkbox"/> ftp	
<input checked="" type="checkbox"/> http	
<input checked="" type="checkbox"/> memory	
<input checked="" type="checkbox"/> mikrotik	
<input checked="" type="checkbox"/> ping	
<input checked="" type="checkbox"/> router	
<input checked="" type="checkbox"/> routeros manage...	
<input checked="" type="checkbox"/> ssh	
<input checked="" type="checkbox"/> switch	
<input checked="" type="checkbox"/> telnet	

10.5.115.8 - Device

General Polling Services Outages Snmp RouterOS History Tools

Remove Resolved

Status	Time	Duration	Service
active	13:27:37	00:09:43	http
resolved	13:33:20	00:03:45	ping

Telegram messenger notification example

Using notification type - **execute on server**, The Dude can send messages using HTTP protocol, for example to **Telegram messenger**.

- 1) Make a server or web app that can receive messages sent by The Dude, in this case a Telegram bot (<https://core.telegram.org/api>)
- 2) Specify URL where the messages will be sent to

```
/tool fetch url="https://api.telegram.org/bot309683994:AAFh8645FeAOgUbcOgUVIhtclbmzpwIXpAB/sendMessage\?chat_id=312605050&text=ExampleMessage" keep-result=no
```

- 3) Add variables to a message for more useful notification

```
...&text= Time: (Time); Device: (Device.FirstAddress); Status: (Service.Status) "
```

Telegram messenger notification example

The screenshot shows the Mikrotik WinBox interface for Firewall and Bandwidth Control. The main window displays a list of notifications, with 'Notification-Telegram' selected. A dialog box titled 'Notification-Telegram - Notification' is open, showing the configuration for this notification.

Notification List:

Name	Type	Notes
beep	beep	
Notification-Telegram	execute on server	
flash	flash	
log to syslog	log	
log to events	log	
popup	popup	

Notification Configuration Dialog:

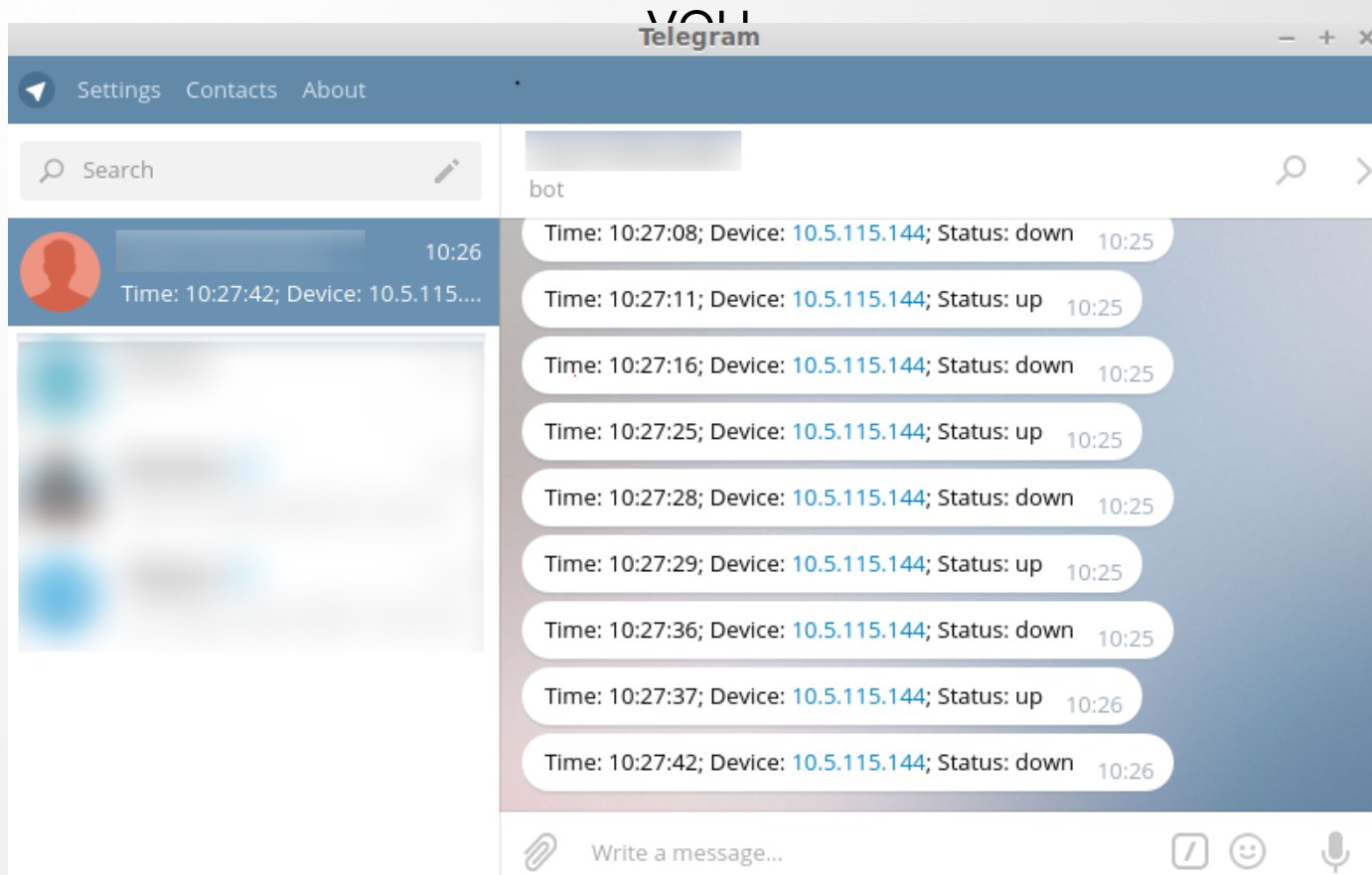
- Name:** Notification-Telegram
- Enabled:**
- Type:** execute on server
- Command:**

```
/tool fetch  
url="https://api.telegram.org/bot309683994:AAFh8645FeAOgUbcOgUVIhtclbmzpwI  
XpA/sendMessage?chat_id=312605050&text=Time: [Time]; Device:  
[Device.FirstAddress]; Status: [Service.Status]"
```

Client: rx 1.34 kbps / tx 353 bps Server: rx 0 bps

Telegram messenger notification example

When a notification command will be triggered by the Dude, the Telegram messenger will receive and resend a message for



Facebook notification example

Since The Dude can execute commands with arguments locally on a operating system where The Dude client is installed, this option can be used to create a custom notification methods



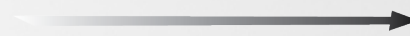
The Dude

1) executes local shell script and passes arguments



Custom program
(e.g. Shell script)

2) Uses arguments to create and send HTTP POST request



Facebook
Messenger bot

3) Delivers notification message to a specified recipient

Facebook notification example



The Dude

To call a custom program on local OS, The Dude need to know a path of the program, and the text which will be passed as argument

Code example:

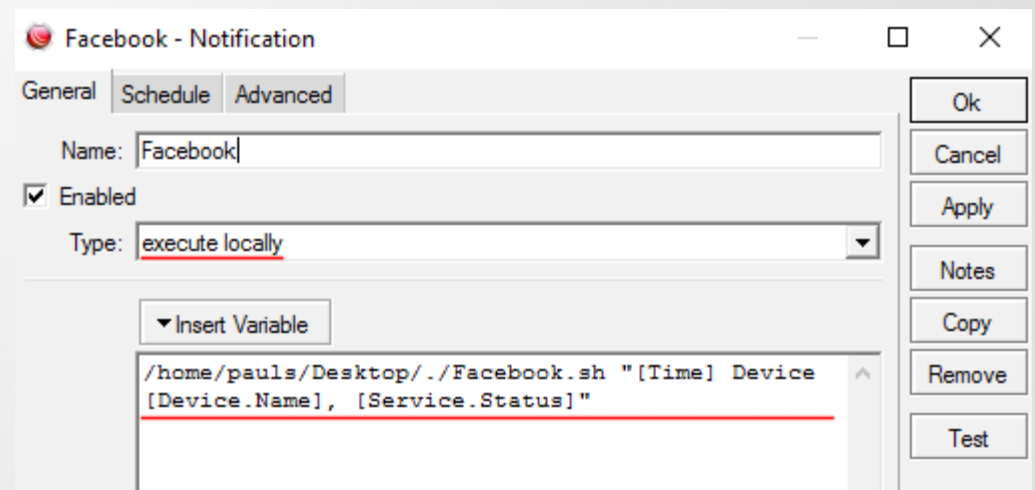
```
/home/pauls/Desktop/./Facebook.sh "(Time) Device (Device.Name), (Service.Status)"
```

Where a path to program is:

```
/home/pauls/Desktop/./Facebook.sh
```

Argument that will be passed:

```
"(Time) Device (Device.Name), (Service.Status)"
```



Facebook notification example



Custom program
(e.g. Shell script)

Since delivering messages over Facebook requires **HTTP POST** request with content type: **application/json**, we need to use some custom program to create such request.
e.g. simple **shell script**

Code executed by The Dude:

```
/home/pauls/Desktop/./Facebook.sh "(Time) Device (Device.Name), (Service.Status)"
```

Argument received by Facebook.sh from The Dude:

```
21:37:36 Device 10.5.115.8, down
```

Facebook.sh example

```
1  #!/bin/bash
2
3  curl -X POST -H "Content-Type: application/json" -d "{
4    \"recipient\": {
5      \"id\": \"1048570848608888\"
6    },
7    \"message\": {
8      \"text\": \"$1\"
9    }
10 }" "https://graph.facebook.com/v2.6/me/messages?access_token=EAY..."
```

Facebook notification example

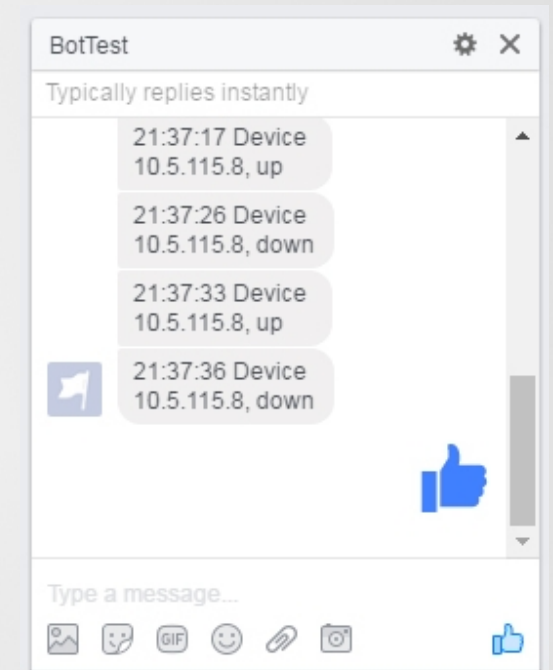


Facebook
Messenger bot

Shell script uses **curl** to create a HTTP POST request and send it to a specified recipient. Message is sent each time when specified device changes status

- Message content is created by **The Dude server**
- Message is sent using **custom program** (e.g shell script)
- Message is delivered using **Facebook messenger bot**

Received message



Additional resources

Forum – User discussions about The Dude, version changelogs
<https://forum.mikrotik.com>

Wiki – The Dude manuals
https://wiki.mikrotik.com/wiki/Manual:The_Dude

Support – Feature requests, suggestions, bug reports
support@mikrotik.com

CHR

CHR – Cloud Hosted Router

- CHR is a RouterOS version intended for running as a virtual machine
- Runs on VMware Workstation and ESXi, VirtualBox, HyperV, KVM, XenServer, etc
- Cloud services - Amazon, Azure and others

CHR – Cloud Hosted Router

License	Speed limit	Price
Free	1Mbit	FREE
P1	1Gbit	\$45
P10	10Gbit	\$95
P-Unlimited	Unlimited	\$250

























- 60 day trial available
- Purchased license can be transferred

CHR – Cloud Hosted Router

CHR can be installed in few steps:

- 1) Download a virtual disk image (mikrotik.com/download)
- 2) Create a guest virtual machine
- 3) Use previously downloaded image file as a virtual disk drive
- 4) Start the guest CHR virtual machine
- 5) Log in to your new CHR. Default user is 'admin', without password

Cloud Hosted Router ?

	6.37.5 (Bugfix only)	6.38.5 (Current)	5.26 (Legacy)	6.39rc72 (Release candidate)
Images	vmdk, vhdx, vdi, img			
VHDX image			-	
VMDK image			-	
VDI image			-	
Raw disk image			-	
Extra packages			-	
The Dude server			-	
The Dude client			-	
Changelog			-	
Checksum	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	<input checked="" type="checkbox"/>

<https://wiki.mikrotik.com/wiki/Manual:CHR>

Suggestions?
Feature requests?

THANK YOU!