



MikroTik™ V2.0 Router Software User Guide

Revision 2000-04-07

© Mikrotiks SIA 1999

Mikrotikls SIA

No part of this document may be reproduced or transmitted in any means, electronic or mechanical, for any purpose, without the written permission of Mikrotikls. Information in this document is subject to change without notice. Mikrotikls makes no representation or warranties with respect to the contents of this manual and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

© 1999 Mikrotikls SIA.
All rights reserved.

Printed in Latvia

MikroTik™ and Mikrotikls are trademarks of Mikrotikls SIA, Riga, Latvia.

Other trademarks used are properties of their respective owners.

Portions of this software may fall under the following copyrights:

Copyright © 1991 D.L.S. Associates

GateDaemon - Copyright © 1995, 1996, 1997, 1998 The Regents of the University of Michigan. All rights reserved. GateDaemon was originated and developed through release 3.0 by Cornell University and its collaborators.

Id.so - Copyright © 1988 Regents of the University of California. All rights reserved. Id.so software was developed by the University of California, Berkeley

cmu snmp - Copyright © 1988, 1989, by Carnegie Mellon University. All rights reserved. Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is granted, provided that the above copyright notice appear in all copies and that copyright notice and this permission notice appear in the supporting documentation.

PPP - Copyright © 1993 The Australian National University

netkit, telnet - Copyright © 1983, 1991 The Regents of the University of California. This product includes software developed by the University of California, Berkeley and its contributors. This software is provided by the regents and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

bash, boa, dialog, e2fsprogs, fileutils, getty_ps, gzip, modutils, mount, net-tools, procps, shellutils, sysklogd, sysvinit, tar, textutils, updated, util-linux - For the above GPL licensed programs in accordance with the GPL license, Mikrotikls offers to organize a copy of the source code or it can be found on most standard Linux distributions. Write support@mikrotik.com for more information.

glibc, libstdc++, ncurses, termcap - For the above LGPL licensed programs in accordance with the LGPL license, Mikrotikls offers to organize a copy of the source code or it can be found on most standard Linux distributions. Write support@mikrotik.com for more information.

TABLE OF CONTENTS

<u>PREFACE</u>	5
DOCUMENT ORGANIZATION	5
DOCUMENT CONVENTIONS	5
<u>1 PPP CONNECTIONS</u>	6
1.1 GLOSSARY OF USED TERMS	6
1.2 LEASED LINE	7
1.2-1 SETTING PARAMETERS	7
1.2-2 TROUBLESHOOTING	8
1.2-3 EXAMPLE.....	9
1.3 NULL-MODEM CONNECTION	11
1.3-1 SETTING PARAMETERS	11
1.3-2 TROUBLESHOOTING	12
1.4 DIAL-IN	13
1.4-1 SETTING PARAMETERS	13
1.4-2 TROUBLESHOOTING	15
1.4-3 EXAMPLE.....	15
1.5 DIAL-OUT	17
1.5-1 SETTING PARAMETERS	17
1.5-2 TROUBLESHOOTING	19
1.5-3 EXAMPLE.....	19
<u>2 QUEUES</u>	21
2.1 UNDERSTANDING THE QUEUING	21
2.2 EXAMPLE	22
<u>3 RULES & CHAINS</u>	23
3.1 NETWORK FIREWALL	23
3.2 FILTERING RULES	23
3.2-1 PACKET DESCRIPTION.....	23
3.2-2 ACTIONS ON RULES	24
3.3 EXAMPLES OF USING FIREWALL	26
3.3-1 PROBLEM # 1	26
3.3-2 PROBLEM # 2.....	28
3.4 MASQUERADING	30
3.4-1 EXAMPLE # 1	31
3.4-2 EXAMPLE # 2.....	32

4	NETWORK ADDRESS TRANSLATION	33
4.1	WHEN TO USE NAT	33
4.2	EXAMPLE OF USING NAT	34
4.2-1	PROBLEM - HOST'S ADDRESS MATCHING	34
4.2-2	PROBLEM - HOST'S ADDRESS AND PORT MATCHING	35
4.2-3	PROBLEM - MASQUERADING AND HOST'S ADDRESS MATCHING	35
4.2-4	PROBLEM - NETWORK ADDRESS MATCHING	36
5	AIRONET WIRELESS INFRASTRUCTURE	38
5	NETWORK PREFIX	40

Preface

Document Organization

This document is divided into 5 parts:

- Part 1. “PPP Connections” describes how to configure ***MikroTik*** router to establish PPP connections, such as leased line, null modem connection, dial-in and dial-out.
- Part 2. “Queues” describes how to use and configure queuing mechanisms.
- Part 3. “Rules and Chains” describes how to turn ***MikroTik*** router into a network firewall and how to configure it.
- Part 4. “Network Address Translation” describes respectively how to configure NAT.
- Part 5. “Aironet Wireless Infrastructure” describes how to configure network of this type.
- Part 6. “Network Prefix” gives some explanation and useful examples of network preface usage.

Each section of the User Guide contains examples of ***MikroTik*** router configuration.

Document Conventions

In this publication, the following conventions are used:

- Commands, arguments, parameters, and keywords are marked out with the `Courier New` font;
- Symbol “\” in the command examples means “new line” and can be used when you enter a real command in the console;

1 PPP Connections

This section describes how to configure the router for the PPP connections.

Note

If you have any questions concerning Console and Java Console commands and arguments, or you find the information in this document insufficient then consult the User Manual for addition information.

1.1 Glossary of Used Terms

Authentication

The process of identifying an individual usually based on a username and password.

CHAP

Short for *Challenge Handshake Authentication Protocol*, a type of authentication in which the authentication agent sends the client program a key to be used to encrypt the password. This is one way encryption, so the password cannot be decrypted. Client program sends the server the encrypted password. The received password and the encrypted password from server's database are compared. The user name is not being encrypted. Contrast with PAP.

Idle Time

Idle time specifies how long the link will be kept up without activity. If there is no traffic for this period of time the link will be closed.

Null-modem cable

A specially designed cable that allows you to connect two computers directly to each other via their communication ports (RS-232 ports).

PAP

Short for *Password Authentication Protocol*, the most basic form of authentication, in which a user's name and password are transmitted over a network and compared to a table of name-password pairs. Typically, the passwords stored in the table are encrypted. The Basic Authentication feature built into the HTTP protocol uses PAP. The main weakness of PAP is that both the username and password are transmitted "in the clear" - that is, in an unencrypted form. Contrast with CHAP.

1.2 Leased Line

Leased line is a permanent telephone connection between two points set up by the organization that provides telecommunication services to the public. Unlike normal dial-up connections, a leased line is always active.

1.2-1 Setting Parameters

To set up connection using leased line you need to set those values:

1. PPP Interface Parameters

Make sure that the type of PPP connection is set to "Direct". In the console you can do that in the "interfaces ppp" menu or in the PPP interface edit box in the Java Console.

2. Authentication

Set the authentication to a desired value. This can be done in the "interfaces ppp" menu in the Console. If you want to do it using the Java Console then select the "Interfaces" menu and then edit the interface (of a type PPP). When you open an interface edit box, authentication is under "PPP" tag. Authentication can be set to *none*, *CHAP* or *PAP*. See *Used Terms Glossary* for explanations. If authentication is set to *none* then the router will not refer to a user name and password.

3. Server Name

When a leased line connection is being established, the router tries to initiate a connection. If it fails, it enters *passive* mode - waits for the other end to connect. If it does not fail (the router on the other end has failed and is the *passive* mode) then both ends try to establish a connection. If the authentication is set to *PAP* or *CHAP*, then user name and server name should be set correctly.

The *Server name* that is defined for the PPP interface should coincide with the one that is defined for the user, which should coincide with the user name. The router will try to connect using the *server name* as a *user name*, if it fails it enters the *passive* (dial-in) mode, using *user name* as a *server name*.

It is necessary to add a new user so that its name and server name is equal or set these parameters for an existing user. User parameters can be edited in the "sys user" menu in the Console. If using a Java Console then select the "System" menu and then on the "Users" menu. Set user's group to PPP and add/edit user parameters as needed.

Now you need to set the server name for the PPP interface equal to the user name and server name. This can be done in the "interfaces ppp" menu in

the Console. If you are using the Java Console, simply open PPP-interface edit box as described in the previous paragraph. Server name is set under the “PPP” tag.

4. Line Speed

Line speed should be set to the maximum your modem supports (most modern modems support 115200**bps** but be sure to check your modem's manual). Line speed is set in the same submenu where authentication is set.

5. Null-modem

Make sure that “Null-modem” option is switched off. In the console you can do that in the “`interfaces ppp`” menu or in the PPP interface edit box in the Java Console.

6. IP Address

IP addresses for PPP interfaces must be configured to establish a connection successfully. There must be two IP addresses configured:

Local address specifies the IP address to use for local interface. It will be told to the peer in the negotiation process.

Remote address specifies the address of the peer.

Depending on the configuration, *local* and *remote* addresses can be remotely set. This means that you can specify *local* and *remote* address on one host and configure no addresses on the other host. Or you can configure *local* addresses for both hosts and not configure the remote address at all - routers will exchange them automatically.

Local and *remote* addresses can be set in the “`ip ppp`” menu in the Console or in the “IP” menu under “PPP” submenu.

1.2-2 Troubleshooting

Check the connection using `monitor` command in the “`interfaces ppp`” menu. Check whether connection is established or not. If not, then check the following:

- Make sure that your PPP interface is set “up”.
- Check the hardware. Maybe the modem is not functioning, some hardware part is switched off or maybe the cable is damaged.
- User name, user's server name, and PPP interface server name do not coincide.
- User does not belong to the group PPP.
- Null modem for the interface is set on.

1.2-3 Example

For example you have two **MikroTik** routers that are connected together with a leased line:

```
Name: router1
Interface: ether1
Ethernet address: 10.1.0.5
Network mask: 255.255.255.240
Network address: 10.1.0.0
```

```
Name: router2
Interface: ether1
Ethernet address: 10.2.0.3
Network mask: 255.255.255.240
Network address: 10.2.0.0
```

You have to establish connection between those two routers. We assume that the hardware that provides the leased line is functioning properly. If you have problems turning it on, consult the modem's manual.

You have two PPP interfaces: `ppp1` and `ppp2`. The connection is made using `ppp1` interface.

We also assume that CHAP authentication is being used.

We will show how to configure both routers. One will be called `[router1]` and the other one `[router2]`.

Remember, that you can observe the current PPP interfaces status anytime:

```
[router1] interface ppp> print
# 0 ppp2          Status: down  MTU: 1500
                  MRU: 1500  Type: dial-in
                  Authentication: none
                  User Name:   Server Name:
                  Line speed: 9600 Idle time(seconds): 0
                  Phone number:   Tone dial: yes
                  Dial on demand: no  Null modem: no
                  Rings: 3
# 1 ppp1          Status: down  MTU: 115200
                  MRU: 1500  Type: direct
                  Authentication: none
                  User Name:   Server Name:
                  Line speed: 9600 Idle time(seconds): 0
                  Phone number:   Tone dial: yes
                  Dial on demand: no  Null modem: yes
                  Rings: 3
```

First of all we will make all necessary set up in the “`interfaces ppp`” menu. You should set PPP connection type, authentication type, server name, line speed and null-modem.

You have to execute the following commands:

```
[router1] interface ppp> set ppp1 up type direct auth \  
chap server test line 115200 nullmodem off
```

```
[router2] interface ppp> set ppp1 up type direct auth \  
chap server test line 115200 nullmodem off
```

Server name is `test` (note that for both routers it is the same), line speed is 115200 bps (maximal possible).

Both routers should have a user with the same parameters: user name (it should be `test`), password, and server name equal to the user name. So the following commands should be executed on both ends:

```
[router1] sys user> add name test password secret group \  
ppp server test
```

```
[router2] sys user> add name test password secret group \  
ppp server test
```

And finally set IP addresses. We will assign local and remote IP addresses on the `router1` and none on the `router2` (remember, that an alternative is to set only local addresses on both ends). We will assign addresses using 30 bits for the network mask, so the network contains only two addresses:

```
Network address: 10.0.0.0  
Network mask: 255.255.255.252  
Address for the router1: 10.0.0.1  
Address for the router2: 10.0.0.1
```

```
[router1] ip ppp> set ppp1 local 10.0.0.1 remote 10.0.0.2
```

```
[router2] ip ppp> set ppp1 local 0.0.0.0 remote 0.0.0.0
```

Now enable the `ppp1` interface and everything should work just fine:

```
[router1] interface ppp> set ppp1 up
```

```
[router2] interface ppp> set ppp1 up
```

You can check the connection using monitor command:

```
[router1] interface ppp> monitor ppp1
```

1.3 Null-modem Connection

Null-modem has almost the same settings as leased line connection, except that null-modem is set “on”.

1.3-1 Setting Parameters

1. PPP Interface Parameters

Make sure that the type of PPP connection is set to “Direct”. In the console you can do that in the “`interfaces ppp`” menu or in the PPP interface edit box in the Java Console.

2. Authentication

Set the authentication to a desired value. This can be done in the “`interfaces ppp`” menu in the Console. If you want to do it using the Java Console then click on the “Interfaces” menu and then edit the interface (of a type PPP). When you open an interface edit box, authentication is under “PPP” tag. Authentication can be set to *none*, *CHAP* or *PAP*. See *Used Terms Glossary* for explanations. If authentication is set to null then the router will not refer to user name and password at all.

3. Server Name

When leased line connection is being established, the router tries to initiate a connection. If it fails, it enters the *passive* mode -- waits for the other end to connect. If it does not fail (the router on the other end has failed and is the *passive* mode) then both ends try to establish a connection. If the authentication is set to *PAP* or *CHAP*, then user name and server name should be set correctly.

Server name that is defined for the PPP interface should coincide with the one that is defined for the user, which should coincide with the user name. That is because of the reasons described above – router will try to connect using the *server name* as a *user name*, if it fails it enters the passive (dial-in) mode, using *user name* as a *server name*.

So add a new user so that its name and server name are equal. Or set these parameters for an existing user. User parameters can be edited in the “`sys user`” menu in the Console. If you are using Java Console then click on the “System” menu and then on the “Users” menu. There you can add/edit users and their parameters. And also set user’s group to PPP.

Now you need to set the server name for the PPP interface equal to the user name and server name. This can be done in the “`interfaces ppp`” menu in the Console. If you are using Java Console simply open PPP-interface edit

box as described in the previous paragraph. Server name is set under the “PPP” tag.

4. Line Speed

Line speed should be set to the maximum your modem supports (most modern modems support 115200**bps** but be sure to check your modem's manual). Line speed is set in the same submenu where authentication is set.

5. Null-modem

Make sure that “Null-modem” option is switched on. In the console you can do that in the “`interfaces ppp`” menu or in the PPP interface edit box in the Java Console.

6. IP Address

IP addresses for PPP interfaces must be configured to establish a connection successfully. There must be two IP addresses configured:

Local address specifies the IP address to use for local interface. It will be told to the peer in the negotiation process.

Remote address specifies what the address of the peer will be.

Both these addresses are optional but they must be known in the time of IP address negotiation. This means that you can specify *local* and *remote* address on one host and configure no addresses on the other host. Or you can configure *local* addresses for both hosts and not configure remote address at all - routers will exchange them automatically.

Local and *remote* addresses can be set in the “`ip ppp`” menu in the Console or in the “IP” menu under “PPP” submenu.

1.3-2 Troubleshooting

Check the connection using `monitor` command in the “`interfaces ppp`” menu. If the connection hasn't been established then one of the following might have happened:

- Make sure that your PPP interface is set up.
- Check the hardware. May be your modem is not functioning, some hardware part is switched off or may be the cable is damaged.
- User name, user's server name and PPP interface server name do not coincide.
- User does not belong to the group PPP.
- Null modem for the interface is set off.

1.4 Dial-In

Dial-in mode is used to make router wait for incoming calls (e.g. dial-up server). When someone connects to the router it will try to authenticate the user - it will try to find the username among the ones whose *server name* is set to the same value as the one specified in PPP interfaces *server name* field. If this operation fails then connection won't established. If it succeeds then connection will be established.

1.4-1 Setting Parameters

Thus, follow these steps to set PPP interface to dial-in mode:

1. PPP Interface Parameters

Make sure that the type of PPP connection is set to "Dial-in". In the console you can do that in the "interfaces ppp" menu or in the PPP interface edit box in the Java Console.

2. Authentication

Set the authentication to a desired value. This can be done in the "interfaces ppp" menu in the Console. If you want to do it using the Java Console then click on the "Interfaces" menu and then edit the interface (of a type PPP). When you open an interface edit box, authentication is under "PPP" tag. Authentication can be set to *none*, *CHAP* or *PAP*. See *Used Terms Glossary* for explanations. If authentication is set to null then the router will not refer to user name and password at all.

3. User Parameters

Remember that only those users whose server name coincide to the one that is set in PPP interfaces will be able to dial-in. User parameters can be edited in the "sys user" menu in the Console. If you are using Java Console then click on the "System" menu and then on the "Users" menu. There you can add/edit users and their parameters. And also set user's group to PPP.

4. Server Name

Set *Server Name* to filter out which users will be able to dial into selected interface or "*" to allow all users. Server name is in the "interfaces ppp" menu in the Console. Open PPP-interface edit box. Server name is set under the "PPP" tag.

5. IP Address

IP addresses for PPP interfaces must be configured to establish a connection successfully. There must be two IP addresses configured:

Local address specifies the IP address to use for local interface. It will be told to the peer in the negotiation process.

Remote address specifies what the address of the peer will be.

Both these addresses are optional but they must be known in the time of IP address negotiation. This means that you can specify *local* and *remote* address on one host and configure no addresses on the other host. Or you can configure *local* addresses for both hosts and not configure remote address at all.

Local and *remote* addresses can be set in the “`ip ppp`” menu in the Console or in the “IP” menu under “PPP” submenu.

6. Line Speed

Line speed should be set to some value that your modem supports. Check your modem manual to be sure. Line speed is set in the same submenu where authentication is set.

7. Idle Time

Idle time specifies how long the link will be kept up without activity. If there is no traffic for this long period of time the link will be closed. Zero value means “don't close the link”. Idle time can be set where all PPP interface settings are done.

8. Null Modem

If null modem flag is set then the serial port will be treated as console port. So you can plug your serial console and work with it. You will be prompted to enter user name and password. If the entered information is correct then you will be logged in the router. To set null modem on go to the “`interfaces ppp`” menu in the Console or in the PPP interface edit box in the Java Console.

9. Rings

The special parameter that specifies the number of rings after which the router will “pick up the phone”. This parameter is set in the “`interfaces ppp`” menu in the Console. In the Java Console click on the “Interfaces” menu and then edit the desired PPP interface. “Rings” parameter is set under the “In” tag.

10. RADIUS Server

User names and passwords can be kept not only on your router, but also on the special RADIUS server. So when the user dials in and her name is not found on the router and RADIUS server is enabled then username and password will be authenticated by this server. To enable RADIUS via Console go to the “`ip ppp radius`” menu. To do it via Java Console enter the “IP”

menu and the “PPP” menu. Click on the RADIUS icon in the appeared list box. And set the required parameters. See User Manual for details. “Use RADIUS” enables the service. “Secret” is the router password for connecting to RADIUS server. “Enable Accounting” means that the router will send user logging information (login/logout time, bytes send/received).

1.4-2 Troubleshooting

If you have set all necessary parameters correctly then everything should work just fine. If it doesn't then something is wrong. Possible problems are the following:

- User name or password may not be typed correctly.
- Server name defined for the user does not coincide with the one set for the interface.
- PPP interface type is not “Dial-In”.
- You switched on “Null modem” parameter and you are trying to dial-in in the router. But now it accepts connection via serial port only.
- User name and password are correct and should be authenticated using RADIUS server, but RADIUS server is not enabled.

1.4-3 Example

You have a **MikroTik** router and a modem and you want to set it all up for dial-in:

Name: MikroTik
Interface: ether1
Ethernet address: 10.1.0.5
Network mask: 255.255.255.240
Network address: 10.1.0.0

No RADIUS server is being used. You want to add three new users that will have permissions to dial-in to your router. You want to assign IP addresses to them automatically (thus users does not need to specify their IP addresses when dialing). So when a user dials in the router she will be given an address of this serial interface she is dialing in. And you want to set idle time for 10 minutes.

Dial-in server is called `DialIn`.

Suppose you have two PPP interfaces on your router: `ppp1` and `ppp2`. You want to use `ppp1` for dial-in and you want to rename it to `dial-in`.

First of all you should set up your PPP interface correctly:

```
[MikroTik] interface ppp> set ppp1 name dial-in type \
dial-in auth chap server DialIn line 115200 rings 3 \
idle 600
```

Now the modem will answer the call after 3 rings. Idle time is set to 600 seconds i.e. 10 minutes.

Now you have to add three new users that will be able to use this dial-in server. Remember that to make this possible you have to set user's server name to DialIn:

```
[MikroTik] sys user> add name user1 password passwd1 \
desc "Dial-in user" group ppp ppp on server DialIn
```

```
[MikroTik] sys user> add name user2 password passwd2 \
desc "Dial-in user" group ppp ppp on server DialIn
```

```
[MikroTik] sys user> add name user3 password passwd3 \
desc "Dial-in user" group ppp ppp on server DialIn
```

Now these three users can use PPP dial-in interface.

Now all that is left to do is to set IP addresses correctly. As it was already said above you can either specify local IP addresses on both ends or specify both local and remote addresses only on one end. We will assign addresses using 30 bits for the network mask, so the network contains only two addresses:

```
Network address: 10.0.0.0
Network mask: 255.255.255.252
Address for the router1: 10.0.0.1
Address for the router2: 10.0.0.1
```

```
[MikroTik] ip ppp> set dial-in local 10.0.0.1 \
remote 10.0.0.2
```

Now you can set PPP interface up and everything should work just fine:

```
[MikroTik] interface ppp> set dial-in up
```

You can check the connection using `monitor` command:

```
[MikroTik] interface ppp> monitor dial-in
```

1.5 Dial-Out

This mode enables the router to connect to some dial-up server. You should follow these steps:

1.5-1 Setting Parameters

1. PPP Interface Parameters

Make sure that the type of PPP connection is set to “Dial-out”. In the console you can do that in the “`interfaces ppp`” menu or in the PPP interface edit box in the Java Console.

2. Authentication

Set the authentication to the same as used on the dial-up server. This can be done in the “`interfaces ppp`” menu in the Console. If you want to do it using the Java Console then click on the “Interfaces” menu and then edit the interface (of a type PPP). When you open an interface edit box, authentication is under “PPP” tag.

3. User Parameters

Set user name for the one that will be used for the authentication. Password will be checked using the router user database. User name is set in the “`interface ppp`” menu in the Console. If you are using Java Console then click on the “Interfaces” menu and then open PPP-interface edit box and enter the user name under “PPP” tag. If this parameter is disabled make sure that PPP connection type is set to “Dual-out”. And also set user’s group to PPP in the “`sys user`” menu.

4. IP Address

IP addresses for PPP interfaces must be configured to establish a connection successfully. There must be two IP addresses configured:

Local address specifies the IP address to use for local interface. It will be told to the peer in the negotiation process.

Remote address specifies what the address of the peer will be.

Both these addresses are optional but they must be known in the time of IP address negotiation. This means that you can specify *local* and *remote* address on one host and configure no addresses on the other host. Or you can configure *local* addresses for both hosts and not configure remote address at all - hosts will exchange them automatically.

Local and *remote* addresses can be set in the “`ip ppp`” menu in the Console or in the “IP” menu under “PPP” submenu.

5. Line Speed

Line speed should be set to some value that your modem supports. Check your modem manual to be sure. Line speed is set in the same submenu where authentication is set.

6. Idle Time

If you want to close your connection after some time of inactivity then set this parameter to a desired value. Zero value means "don't close the link". Idle time can be set where all PPP interface settings are done.

7. Phone Number

Set the phone number to the server's phone number. This can be done in the "interface ppp" menu in the Console and in the PPP-interface edit box under the tag "Out" in the Java Console.

8. Tone Dial

If your phone line supports tone dial then set this parameter to "ON". To do that go to the "interfaces ppp" menu in the Console or in the PPP interface edit box in the Java Console.

9. Dial On Demand

Dial on demand means that router won't try to connect immediately when selected PPP interface is enabled but only when some traffic will arrive to this interface. For this to work you'll have to configure routes to this interface because otherwise there will never be any traffic coming to this interface. You may also find it useful to set *idle time* to some reasonable value so that when there's no traffic the link is closed (it will be opened again if traffic comes in again). This parameter is set in the "interfaces ppp" menu in the Console. In the Java Console click on the "Interfaces" menu and then edit the desired PPP interface. "Rings" parameter is set under the "Out" tag.

10. Default Route

If you need to use dial-up server's address as a default route, you should set PPP interface parameter `defaultroute` to `on`. This is not required if you want to use only dial-up. But if you want to send your packets somewhere further, than you can use this parameter. But make sure that the router doesn't have any default route, as it can have only ONE default route. So delete an existing default route and then set this parameter to `on`.

1.5-2 Troubleshooting

If you have set all necessary parameters correctly then everything should work just fine. If it doesn't then something is wrong. Possible problems are the following:

- User name or password may not be typed correctly.
- The user name that is defined for the PPP interface couldn't be found in the router's user database.
- PPP interface type is not "Dial-Out".
- Phone number is not correct.
- You enabled tone dialing but your phone line doesn't support it.

1.5-3 Example

You have a **MikroTik** router and a modem and you want to set it all up for dial-out:

Name: MikroTik
 Interface: ether1
 Ethernet address: 10.1.0.5
 Network mask: 255.255.255.240
 Network address: 10.1.0.0

You have an account on a server that allows dial-in. This server uses PAP protocol for authentication. You do not have to assign local and remote IP addresses as the server will assign them automatically. Your phone line supports tone dial. And you want to set idle time for 10 minutes (i. e. 600 seconds). You do not want to use dial on demand mode. Also you want to use dial-up server's address as a default route.

Your account information is the following:

User name: john
 Password: doe.

Dial-up server phone number is 123456.

Suppose you have two PPP interfaces on your router: `ppp1` and `ppp2`. You want to use `ppp2` for dial-out and you want to rename it to `dial-out`.

First of all you should set up your PPP interface:

```
[MikroTik] interface ppp> set ppp2 name dial-out type \
dial-out auth pap user john line 115200 phone 123456 \
nullmodem off tone on demand off idle 600 defaultroute on
```

Now you have to add corresponding user to the router's user database:

```
[MikroTik] sys user> add name john password doe ppp on \  
group ppp desc "Dial-out user"
```

Since all required settings are made, all you have to do is to enable PPP interface:

```
[MikroTik] interface ppp> set dial-out up
```

You can check the connection using `monitor` command:

```
[MikroTik] interface ppp> monitor dial-out
```

2 Queues

When administering large networks, the traffic burst problem occurs rather often. One of the ways to avoid network traffic 'jams' is usage of traffic shaping. Thus, an administrator is able to allocate a definite portion of the total bandwidth and grant it to a particular network segment or interface. Also by using this mechanism the bandwidth of a particular nodes can be limited.

2.1 Understanding the Queuing

Queuing is a mechanism that gives an opportunity to control vital network properties, such as bandwidth allocation, delay variability, timely delivery and delivery reliability.

When a new queue is available, there is an option provided to change its queuing type (or algorithm).

PFIFO – standing for Packet First-In First-Out – is the simplest queuing algorithm. The packets are served in the same order as they are received. This is a default value for **MikroTik** router.

BFIFO – the same as above, except that this algorithm is byte-based but not packet-based.

RED – standing for Random Early Detection – an algorithm for congestion avoidance in packet-switched networks.

Split – this type allows the packets to be sorted by flow mark and specify the parameters for each such subnode separately. Each subnode can be of type *RED*, *PFIFO*, *BFIFO* or *Split*. If it is of type *Split* then can be split further. Otherwise it cannot.

For small limitations (64kBits, 128kBits) *RED* is more preferable. For larger speeds *PFIFO* will be as good as *RED*. *RED* consumes more memory and consumes more CPU then *PFIFO* & *BFIFO*.

Queuing mechanism uses flow marks to distinguish packets from the different networks. The packets can be marked in any chain (see the section about *Firewalls* for more details). If the packet went through the rules of several chains and was marked more than one time, then the last mark will be taken into account.

2.2 Example

For example your **MikroTik** router has an ether0 interface through which some client's traffic is running. The router is connected to the Internet via ether1 interface. This person's bandwidth should be reduced to 10000 bytes per second (for example he pays only for that amount of data).

Router configuration:

Name: MikroTik

ether0 address: 10.1.0.1 netmask: 255.255.255.0

ether1 address: 10.2.0.1 netmask: 255.255.255.0

This client's IP address: 10.1.0.2 netmask: 255.255.255.0

First of all you have to mark the packets going to and from this client's host. Let us mark the packets from the client in the *input* chain for any interface and the packets to the client in the *output* chain of any interface also. And the flow mark will be "abc". So you have to add two new rules:

```
[MikroTik] ip firewall rule> add input srcaddr 10.1.0.2 \
srcmask 255.255.255.255 flow abc
```

```
[MikroTik] ip firewall rule> add output dstaddr 10.1.0.2\
dstmask 255.255.255.255 flow abc
```

Now all the packets going to/from that client's host will be marked.

The next step is to set the queuing type. Let us choose the default one (PFIFO). Set the queue parameters in the Java Console.

- Set the *ether0* queue type to *split*;
- Add new queue by clicking on the "+" icon;
- Leave the queue type default;
- Under "Flow" tag make the appropriate settings:
 - set the "Flow Mark" to "abc";
 - set the "Limited At" to 10000 bytes per second;
 - select the "Bounded" checkbox;
 - leave other parameters as they were set by default;

Also make the same settings for the *ether1* interface.

That is all you have to do. Now the client's bandwidth is limited to 10000 bytes per second.

3 Rules & Chains

3.1 Network Firewall

A firewall is a system or group of systems that enforces an access control policy between two networks. The actual means by which this is accomplished varies widely, but in principle, the firewall can be thought of as a pair of mechanisms: one which exists to block traffic, and the other which exists to permit traffic. Some firewalls place a greater emphasis on blocking traffic, while others emphasize permitting traffic.

Probably the most important thing to recognize about a firewall is that it implements an access control policy. If you don't have a good idea what kind of access you want to permit or deny, or you simply permit someone or some product to configure a firewall based on what they or it think it should do, then they are making policy for your organization as a whole.

3.2 Filtering Rules

Generally, the filtering rules are expressed as a table of conditions and actions that are applied in a certain order until a decision to route or drop the packet is reached. When a particular packet meets all the conditions specified in a given row of the table, the action is carried out specified in that row (whether to route or drop the packet) is carried out. In some filtering implementations, the action can also indicate whether or not to notify the sender that the packet has been dropped (through an ICMP message), and whether or not to log the packet and the action taken on it.

Rules are organized in chains. There are three default chains, which cannot be deleted. More chains can be added for grouping together filtering rules. When processing a chain, rules are taken from the chain in the order they are listed from top to bottom.

3.2-1 Packet Description

Packets can be described using the following criteria:

- Source network address and mask;
- Destination network address and mask;
- Interfaces
 - ✓ for input packets: interface through which packet entered the router;
 - ✓ for forwarding packets: interface through which the packet should exit the router;
 - ✓ for output packets: interface through which the packet should exit the router;
- TCP protocol flags. Works only for TCP protocol.

Criteria is:

- ✓ *any* flag – includes all TCP option packets;
- ✓ *syn* flag – includes only connection establishing packets;
- ✓ *nosync* flag – includes all TCP option packets except connection establishing packets.

It is also possible to use an inversion of some of the mentioned parameters. Actually, any configuration is possible without inversion, but it can reduce number of rules thus increase the speed of filtering process.

3.2-2 Actions on Rules

Actions that can be applied to the packet

Action	Description
Accept	Accept packet (and omit remaining rules)
Reject	Reject packet and send message: ICMP Destination Unreachable
Deny	Drop packet (without sending any messages)
Masq	Use masquerading (works only for <i>forward</i> chain)
Return	Return to the chain from which this rule has been invoked
Jump	Jump to another chain

Flow mark

Packet can also be marked as one that belongs to certain traffic. This parameter is used for Class-Based Queuing. Flow mark is used only inside the router, and it is lost when the packet exits the router. If there are several rules for marking the packet, then after each rule is applied, the flow mark is overwritten with the one specified in that rule. So the packet is marked with the flow mark from the last applied rule.

Chains

- ◇ Packet that entered the router first of all “moves” through the **input** chain;
- ◇ Packet that should be forwarded - through the **forward** chain;
- ◇ The one that exits the router - through the **output** chain.

So these three chains are default ones. They cannot be deleted. It is possible to create new chains.

You can create a rule with action *JUMP* that “moves” packet to another chain and applies its rules to the packet. Action *RETURN* allows to return to previous chain (the one that invoked the last *JUMP*) and continue applying the rest of its rules to the packet.

Question: *Why this is useful?*

Answer:

1. Sometimes *INPUT* and *OUTPUT* chains should contain identical rules. For that we can make a new chain with those rules and add *JUMP* rule (jump to the new chain) in *INPUT* and *OUTPUT* chains.
2. Sometimes a rule set should be applied to several interfaces. For that we can group the rules in a chain, without specifying the interface for these rules. Make a rule that detects packets coming through the specified interface and uses *JUMP* action to the new chain.
3. Sometimes a rule set should be applied for several IP networks or hosts. For that we can group the rules in a chain, without specifying the source and destination addresses. Make a rule that detects packets for the desired network and uses *JUMP* action to the new chain.
4. Too many rules. The more rules a chain contains the slower filtration works.

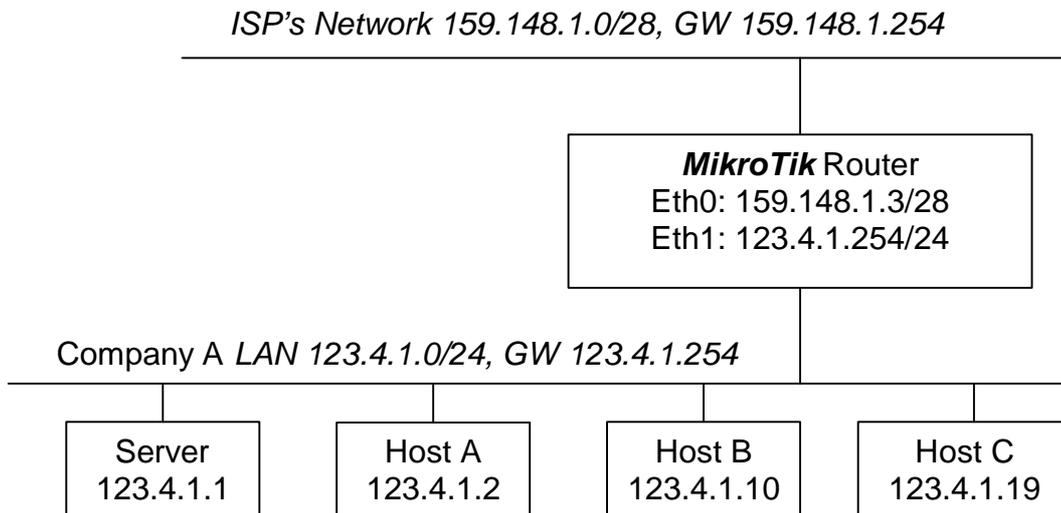
Filtering has some impact on the router's performance. Constructing your filter lists so that, on average, the fewest possible filter rules are checked for each packet can minimize this impact. For example, placing a rule that matches all TCP packets that are not connection open attempts (i.e. filter protocol "tcp-nosyn") at the start of the list should improve things a lot.

The first thing we can do is to move the most often used rules to the beginning of the chain. But this is not always possible. Then we can create a new chain that will contain majority of the rules, so it will reduce this "initial" chain.

3.3 Examples of Using Firewall

3.3-1 Problem # 1

Consider the following network structure:



Company A wants its computers have free access to the Internet, but have firewall to prevent unauthorized access to the LAN and the router from the outside world.

At the same time, Company A wants to run a web, ftp, and mail server, that would be accessible from the outside. Also, Company's boss wants unlimited access to his computer 123.4.1.10 from his remote office network (123.4.10.0/24).

Ping and UDP packets are accepted, if the destination is Company's network or and the router.

In such a situation, following set of rules can be implemented for the input chain:

Allow established connections.

This is placed at the top of the chain to improve the router's performance

```
/add input action accept srcaddr !123.4.1.01 \
srcmask !255.255.255.0 dstaddr 123.4.1.0 \
dstmask 255.255.255.0 interface Eth0 tcp nosyn
```

```
/add input action accept srcaddr !159.148.1.3 \
srcmask !255.255.255.255 dstaddr 159.148.1.3 \
```

¹ Symbol "!" means "NOT".

```
dstmask 255.255.255.255 interface Eth0 tcp nosyn
```

Drop and log attempts to spoof internal addresses, or the external address

```
/add input action deny srcaddr 123.4.1.0 \  
srcmask 255.255.255.0 interface Eth0 proto ALL log yes
```

```
/add input action deny srcaddr 159.148.148.3 \  
srcmask 255.255.255.255 interface Eth0 proto ALL log yes
```

Allow UDP responses

```
/add input action accept dstaddr 123.4.1.0 \  
dstmask 255.255.255.0 interface Eth0 proto UDP
```

Allow incoming authentication and pop3, http, mail, and ftp connections

```
/add input action accept dstaddr 124.4.1.0 \  
dstmask 255.255.255.0 dstports 113-113 interface Eth0 \  
tcp syn
```

```
/add input action accept dstaddr 124.4.1.1 \  
dstmask 255.255.255.255 dstports 110-110 interface Eth0 \  
tcp syn
```

```
/add input action accept dstaddr 124.4.1.1 \  
dstmask 255.255.255.255 dstports 80-80 interface Eth0 \  
tcp syn
```

```
/add input action accept dstaddr 124.4.1.1 \  
dstmask 255.255.255.255 dstports 25-25 interface Eth0 \  
tcp syn
```

```
/add input action accept dstaddr 124.4.1.1 \  
dstmask 255.255.255.255 dstports 21-21 interface Eth0 \  
tcp syn
```

```
/add input action accept srcports 20-20 \  
dstaddr 124.4.1.0 dstmask 255.255.255.0 interface Eth0 \  
tcp syn
```

#Allow ping packets

```
/add input action accept dstaddr 123.4.1.0 \  
dstmask 255.255.255.0 interface Eth0 proto ICMP
```

```
/add input action accept dstaddr 159.148.1.3 \  
dstmask 255.255.255.255 interface Eth0 proto ICMP
```

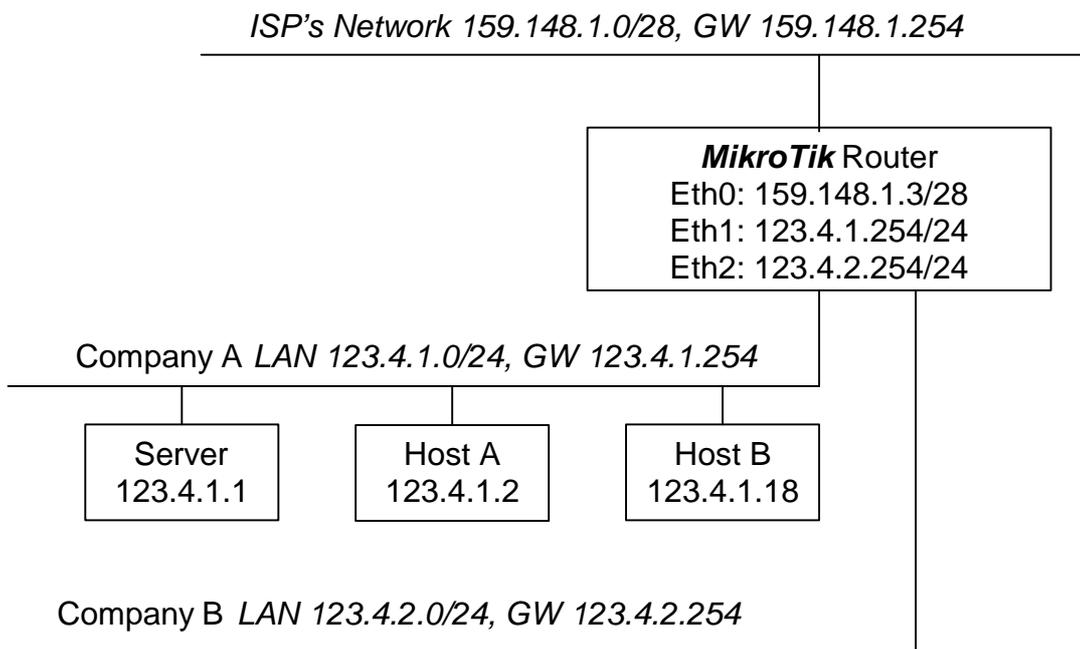
#Drop and log everything else

```
/add input action deny proto ALL interface Eth0 log yes
```

Good idea is to monitor the logs for a while after setting up the filtering. You may have been forgotten some service you were providing and add the rule for it above the last one. Your final goal is to track the bad guys only.

3.3-2 Problem # 2

Network structure is like in the previous Example, except that you have two internal networks. There is a Company B network 123.4.2.0/25, that also needs to be protected.



One option would be to 'duplicate' the input chain rules to reflect the presence of another network. Instead one may wish to make a new chain, which is processed only if the rules of the input chain match the packets for one or another network. Otherwise, the last rule of the input chain drops and logs the unsolicited packet:

Allow packets from other networks than ours, everything else is denied

```
#Add a new chain in the "ip firewall chain" menu
/add name ab
```

```
#No add the required rules
/add input action jump target ab proto All \
srcaddr !123.4.1.0 srcmask !255.255.255.0 \
dstaddr 123.4.1.0 dstmask 255.255.255.0 interface Eth0
```

```
/add input action jump target ab proto All \
srcaddr !123.4.2.0 srcmask !255.255.255.0 \
dstaddr 123.4.2.0 dstmask 255.255.255.0 interface Eth0
```

```
/add input action jump target ab proto All \
srcaddr !159.148.1.3 srcmask !255.255.255.255 \
dstaddr 159.148.1.3 dstmask 255.255.255.255 \
interface Eth0
```

```
/add input action deny proto All log yes interface Eth0
```

The new chain would be as follows:

Allow established connections

```
/add ab action accept tcp nosyn interface Eth0
```

Drop and log attempts to spoof internal addresses, or the external address

```
/add ab action deny proto ALL log yes srcaddr 123.4.1.0 \
srcmask 255.255.255.0 interface Eth0
```

```
/add ab action deny proto ALL srcaddr 159.148.1.3 \
srcmask 255.255.255.255 interface Eth0 log yes
```

Allow UDP responses

```
/add ab dstaddr 123.4.1.0 dstmask 255.255.255.0 \
proto UDP action accept interface Eth0
```

Allow incoming authentication and pop3, http, mail, and ftp connections

```
/add ab action accept dstaddr 124.4.1.0 \
dstmask 255.255.255.0 dstports 113-113 interface Eth0 \
tcp syn
```

```
/add ab action accept dstaddr 124.4.1.1 \
dstmask 255.255.255.255 dstports 110-110 interface Eth0 \
tcp syn
```

```
/add ab action accept dstaddr 124.4.1.1 \
dstmask 255.255.255.255 dstports 80-80 interface Eth0 \
tcp syn
```

```
/add ab action accept dstaddr 124.4.1.1 \
dstmask 255.255.255.255 dstports 25-25 interface Eth0 \
tcp syn
```

```
/add input action accept dstaddr 124.4.1.1 \
dstmask 255.255.255.255 dstports 21-21 interface Eth0 \
tcp syn
```

```
/add ab action accept srcports 20-20 \  
dstaddr 124.4.1.0 dstmask 255.255.255.0 interface Eth0 \  
tcp syn
```

#Allow ping packets

```
/add ab action accept dstaddr 123.4.1.0 \  
dstmask 255.255.255.0 interface Eth0 proto ICMP
```

```
/add ab action accept dstaddr 159.148.1.3 \  
dstmask 255.255.255.255 interface Eth0 proto ICMP
```

#Drop and log everything else

```
/add ab action deny proto ALL interface Eth0 log yes
```

3.4 Masquerading

Masquerading is used for enabling hosts with local addresses to communicate with other networks using the interface address of the gateway router.

Masquerading Principles

Masquerading enables establishing connections between local network hosts and hosts outside the local network, if the local host initializes the connection. Outside network hosts can connect to local hosts with non-global addresses only if

- there is appropriate routing to the local network, or
- Network Address Translation (NAT) is used, or
- there is secondary address space and routing used for these hosts.

Timeouts

Each connection has its timeout – if no packets are passing the router for this certain time it frees all related to this connection resources.

Here are the timeout values:

1. For TCP 15 minutes.
2. For TCP fin state 2 minutes.
3. For UDP 5 minutes.
4. For ICMP 125 seconds.

Each specific masquerading (FTP, Quake, etc.) is limited to 12 connections.

3.4-1 Example # 1

For example, a router has two interfaces and following addresses assigned to them:

interface1: local

address/netmask: 192.168.0.1/255.255.255.0

interface2: global

address/netmask: 159.148.60.2/255.255.255.224

In this case, local hosts from the network 192.168.0.0/255.255.255.0 can use their default gateway 192.168.0.1 (router) to access hosts on other networks, if the following firewall rule is added to the forward chain:

Action: masquerade

Source Addr/Mask: 192.168.0.0/255.255.255.0

Destination Addr/Mask: 0.0.0.0/0.0.0.0

Protocols/Interfaces: all

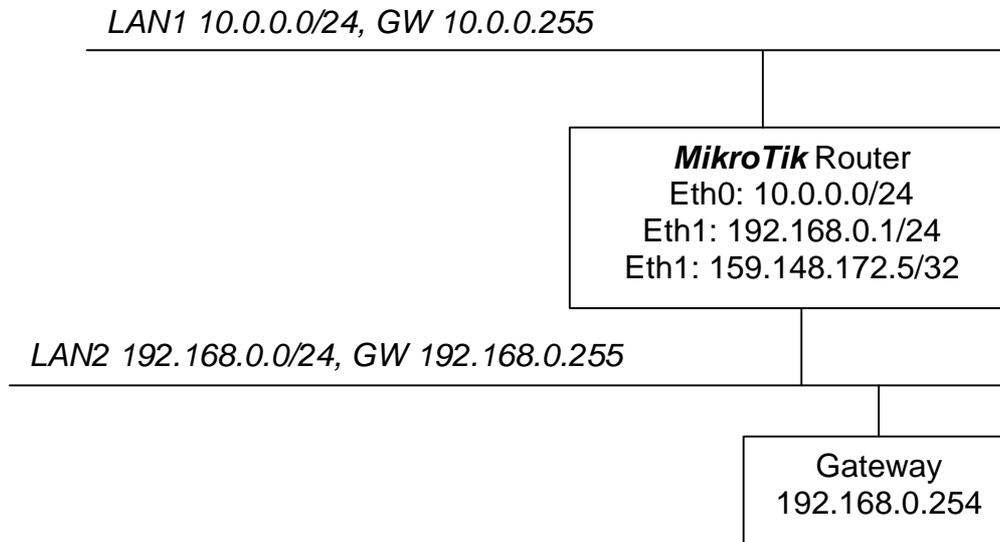
According to this rule, packets originating from network 192.168.0.0/255.255.255.0 will be masqueraded, i.e., their source address will be substituted by the external interface address 159.148.60.2 of the router.

Implementation

Add the above-described rule to the forward chain:

```
[MikroTik] ip firewall rule> add forward action masq \  
srcaddr 192.168.0.0 srcmask 255.255.255.0
```

Interfaces and ports are set to `all` by default.

3.4-2 Example # 2

As this diagram shows MikroTik router connects two Local Area Networks:

LAN1: 10.0.0.0/24

LAN2: 192.168.0.0/24

Router has two interfaces and three IP addresses. One of these addresses is a global IP address. Interface that is connected to LAN2 has two addresses:

Eth1: 192.168.0.1/24

Eth1: 159.148.172.5/32

Consider the following situation: you want the packets originated from LAN1 to be routed to LAN2 gateway and to masquerade their source addresses, i.e. to replace them by the router's global IP address.

```
# Masquerading LAN1 IP addresses
[MikroTik]> ip firewall rule add forward action masq \
srcaddr 10.0.0.0 srcmask 255.255.255.0 interface Eth1
```

```
# Routing the packets to LAN2 gateway
[MikroTik]> ip route add gw 192.168.0.254 presrc \
159.148.172.5 interface Eth1
```

4 Network Address Translation

NAT (Network Address Translation) is the translation of an IP address used within one network to a different IP address known within another network. One network is designated the *inside* network and the other is the *outside*. Typically, an administrator maps the local inside network addresses to one or more global outside IP addresses and unmaps the global IP addresses on incoming packets back into local IP addresses. This helps ensure security since each outgoing or incoming request must go through a translation process that also offers the opportunity to qualify or authenticate the request or match it to a previous request. NAT is not used to conserve on the number of global IP addresses and to let the whole network use a single IP address in its communication with the world. The masquerading function of firewalling should be used instead.

For more information about NAT, see RFC 1631. For example, you can visit this site: <http://www.faqs.org/rfcs/rfc1631.html>

4.1 When to Use NAT

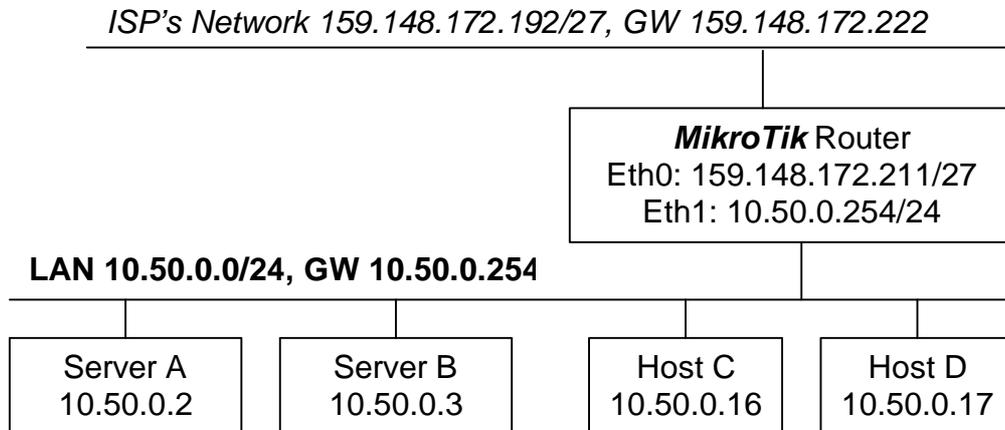
NAT should be used when:

- Local addresses should be matched to external addresses for accessing hosts with local addresses from outside, and/or
- Local ports have to be matched to different external ports.

In all other cases, when local addresses are used, and local hosts need access to external networks, the masquerading function of firewalling would do the address translation from local addresses to one external address – that of the router's external interface.

4.2 Example of Using NAT

Let us consider a network topology shown in the following diagram:



4.2-1 Problem - Host's Address Matching

Server A should be accessible using global address 159.148.172.130. The following NAT rules should be added to translate between global address 159.148.172.130 and local address 10.50.0.2

```
# Incoming packets
/ip nat add interface Eth0 direction in translate on \
dst 159.148.172.130 dmask 255.255.255.255 \
nat-dst 10.50.0.2 nat-dmask 255.255.255.255
#
# Outgoing packets
/ip nat add interface Eth0 direction out translate on \
src 10.50.0.2 smask 255.255.255.255 \
nat-src 159.148.172.130 nat-smask 255.255.255.255
```

The IP address 159.148.172.130/32 must be routed by the ISP's gateway router 159.148.172.222 to your router 159.148.172.211.

4.2-2 Problem - Host's Address and Port Matching

Port 80 of Server B should be accessible using global address 159.148.172.129 and port 8079. The following NAT rules should be added to translate between global address:port 159.148.172.129:8079 and local address:port 10.50.0.1:80

```
# Incoming packets
/ip nat add interface Eth0 direction in translate on \
dst 159.148.172.129 dmask 255.255.255.255 \
dports 8079-8079 \
nat-dst 10.50.0.1 nat-dmask 255.255.255.255 \
nat-dport 80
#
# Outgoing packets
/ip nat add interface Eth0 direction out translate on \
src 10.50.0.1 smask 255.255.255.255 \
sports 80-80 \
nat-src 159.148.172.129 nat-smask 255.255.255.255 \
nat-sport 8079
```

The IP address 159.148.172.129/32 must be routed by the ISP's gateway router 159.148.172.222 to your router 159.148.172.211.

4.2-3 Problem - Masquerading and Host's Address Matching

If the Host C and Host D need access to other networks, a masquerading rule should be added to the forward chain if of IP firewalling configuration:

```
# Leave the packets from Server A unchanged.
# (NAT will do the address translation)
/ip firewall rule add forward \
action none interface Eth0 \
srcaddr 10.50.0.2 srcmask 255.255.255.255
#
# Use the address of router's interface Eth0
# 159.148.172.211 for all other connections
/ip firewall rule add forward \
action masq interface Eth0 \
srcaddr 10.50.0.0 srcmask 255.255.255.0
```

If you do not want to use the router's address 159.148.172.211, you can add a pair of NAT rules that do the address translation to 159.148.172.128

```
# Incoming packets
/ip nat add interface Eth0 direction in translate on \
dst 159.148.172.128 dmask 255.255.255.255 \
nat-dst 159.148.172.211 nat-dmask 255.255.255.255
#
# Outgoing packets
/ip nat add interface Eth0 direction out translate on \
src 159.148.172.211 smask 255.255.255.255 \
nat-src 159.148.172.128 nat-smask 255.255.255.255
```

NAT is processed after the firewall, if a packet is leaving the router through some interface. Therefore the source address:port of packets originated from 10.50.0.0/24 will be first replaced by the routers address 159.148.172.211 and a new port of the router will be allocated for the connection. Then, NAT will replace the new source address 159.148.172.211 of the packet by address 159.148.172.128

The IP address 159.148.172.128/32 must be routed by the ISP's gateway router 159.148.172.222 to your router 159.148.172.211.

4.2-4 Problem - Network Address Matching

You want to match a global network address to the local network address and leave port numbers unchanged, i. e. destination address of the packets destined to the network 159.148.172.128/29 will be replaced with the local network address 10.50.0.0/29.

You have to execute the following command:

```
# Incoming packets
/ip nat add interface Eth0 direction in translate on \
dst 159.148.172.128 dmask 255.255.255.248 \
nat-dst 10.50.0.0 nat-dmask 255.255.255.248
#
# Outgoing packets
/ip nat add interface Eth0 direction out translate on \
src 10.50.0.0 smask 255.255.255.248 \
nat-src 159.148.172.128 nat-smask 255.255.255.248
```

Packets will be translated according to the following table:

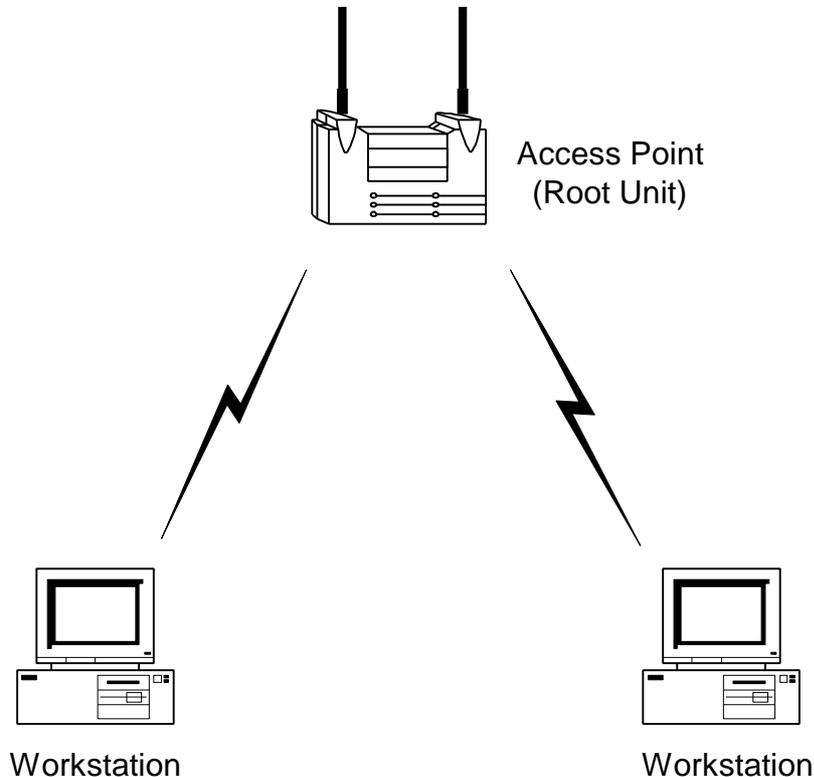
```
159.148.172.129 <-> 10.50.0.1
159.148.172.130 <-> 10.50.0.2
159.148.172.131 <-> 10.50.0.3
159.148.172.132 <-> 10.50.0.4
159.148.172.133 <-> 10.50.0.5
159.148.172.134 <-> 10.50.0.6
```

For correct matching of IP addresses between two subnets, it is required, that both subnets are of equal size.

The whole subnet 159.148.172.128/29 must be routed to 159.148.172.211 by the ISP's router 159.148.172.222.

5 Aironet Wireless Infrastructure

The wireless infrastructure is the communications system that combines Access Points, mobile nodes and fixed nodes.



If you have chosen this network type, you have to set the required parameters correctly. Minimal settings for this network type are:

1. Network type
This parameter is set to `infrastructure` network type by default. So you do not have to set this parameter.
2. Receive/Transmit Diversity
This parameter should be changed if you have not one, but two antennas. Default Aironet adapter settings are: one right antenna. By default receive and transmit diversities are set to `right`. So you should not have to change this parameter if you didn't change the adapter default configuration.
3. Data Rate
This parameter default value is `1 Mbps`. Possible values are (in Mbps): `1`, `2`, `5.5`, `11` and `auto`. The value must be set to correspond to the associated Access Point setting.

4. Service Set Identifier (SSID)

This value **MUST** match the SSID of any/all Access Points that you want to communicate with. You can specify 3 (three) SSID's for one interface. If connection with the first one fails, the software will try the second one and so on.

5. Access Point MAC Address

This parameter is not necessarily required. But if you want you can specify up to 4 (four) MAC addresses of the Access Points.

So the minimal settings for your Aironet adapter to work can be done with the following command:

```
[MikroTik] interface pc> set pc1 up rate auto \  
ssid1 ident_one
```

Use `monitor` command to see whether the node is associated with the Access Point.

```
[MikroTik] interface pc> monitor pc1
```

To see the current settings of an interface use `print` command

5 Network Prefix

Network Prefix is a parameter used in IP address configuration (see section 6.1 of the Software Manual). Network Prefix shows, what network can be reached through the specified interface with the specified IP address. If not defined, it will be calculated from Local Address and Network Mask.

For example, you can add an address:

```
[MikroTik] ip address> add local 10.1.1.1 \
mask 255.255.255.0 prefix 10.1.1.0 interface ether1
```

This means that the network 10.1.1.0 with 24 bit network mask can be reached through the interface `ether1` with the local address 10.1.1.1.

But you can specify network prefix so that the interface IP address is not from that network. For example:

```
[MikroTik] ip address> add local 20.0.0.1 \
mask 255.255.255.0 prefix 10.1.1.0 interface ether1
```

As in the previous example, this means that the network 10.1.1.0 with 24 bit network mask can be reached through the interface `ether1` with the local address 20.0.0.1.

This can be useful in case of the point-to-point interfaces to save IP addresses (in Cisco terminology this is called "unnumbered interfaces"). You cannot reach the network through them, so there is no need to make IP network for that connection, still you have to specify what host is on other side of connection. This can be achieved by using mask 255.255.255.255. This will cause new "host" route to appear (instead of usual "network" route) in the routing table.

Consider this example of synchronous interface:

```
[MikroTik] ip address> add local 10.0.0.1 \
mask 255.255.255.255 prefix 10.0.0.2 interface sync0
```

So if you have only one IP address and several synchronous interfaces, you can simply change the prefix in each command:

```
[MikroTik] ip address> add local 10.0.0.1 \
mask 255.255.255.255 prefix 10.0.0.2 interface sync0
```

```
[MikroTik] ip address> add local 10.0.0.1 \
mask 255.255.255.255 prefix 10.0.0.3 interface sync1
```

```
[MikroTik] ip address> add local 10.0.0.1 \
mask 255.255.255.255 prefix 10.0.0.4 interface sync2
```

```
[MikroTik] ip address> add local 10.0.0.1 \  
mask 255.255.255.255 prefix 10.0.0.5 interface sync3
```

The hosts on the other end of each connection have to be configured with prefix 10.0.0.1 mask 255.255.255.255.